Segmentation of Moving Objects in Video Sequences with a Dynamic Background

Chu TANG

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

September 2012

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Chu Tang

Entitled: "Segmentation of Moving Objects in Video Sequences with a Dynamic Background"

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____	Chair
Dr. R. Raut	
_____	Examiner, External
Dr. T. Fancott (CSE)	To the Program
_____	Examiner
Dr. M. N. S. Swamy	
_____	Supervisor
Dr. C. Wang	
_____	Supervisor
Dr. M. O. Ahmad	

Approved by: _____

Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

_____20_____ _____

Dr. Robin A. L. Drew
Dean, Faculty of Engineering and
Computer Science

ABSTRACT

Segmentation of Moving Objects in Video Sequences with a Dynamic Background

Chu Tang, MASc.

Concordia University, 2012

Segmentation of objects from a video sequence is one of the basic operations commonly employed in vision-based systems. The quality of the segmented object has a profound effect on the performance of such systems. Segmentation of an object becomes a challenging problem in situations in which the background scenes of a video sequence are not static or contain the cast shadow of the object. This thesis is concerned with developing cost-effective methods for object segmentation from video sequences having dynamic background and cast shadows.

A novel technique for the segmentation of foreground from video sequences with a dynamic background is developed. The segmentation problem is treated as a problem of classifying the foreground and background pixels of the frames of a sequence using the pixel color components as multiple features of the images. The individual features representing the pixel gray levels, hue and saturation levels are first extracted and then linearly recombined with suitable weights to form a scalar-valued feature image. Multiple features incorporated into this

scalar-valued feature image allows to devise a simple classification scheme in the framework of a support vector machine classifier. Unlike some other data classification approaches for foreground segmentation, in which a priori knowledge of the shape and size of the moving foreground is essential, in the proposed method, training samples are obtained in an automated manner. The proposed technique is shown not to be limited by the number, patterns or dimensions of the objects.

The foreground of a video frame is the region of the frame that contains the object as well as its cast shadow. A process of object segmentation generally results in segmenting the entire foreground. Thus, shadow removal from the segmented foreground is essential for object segmentation. A novel computationally efficient shadow removal technique based on multiple features is proposed. Multiple object masks, each based on a single feature, are constructed and merged together to form a single object mask. The main idea of the proposed technique is that an object pixel is less likely to be indistinguishable from the shadow pixels simultaneously with respect to all the features used.

Extensive simulations are performed by applying the proposed and some existing techniques to challenging video sequences for object segmentation and shadow removal. The subjective and objective results demonstrate the effectiveness and superiority of the schemes developed in this thesis.

AKNOWLEDGEMENTS

I would like to take this opportunity to express my deep gratitude to my supervisors, Professor M. Omair Ahmad and Professor Chunyan Wang, for their constant support, patience and invaluable guidance during this research. I am grateful to them for spending many hours with me discussing about my research. The suggestions from them have been very useful.

I would also express my sincere thanks to my dear parents for their support and encouragement during my study and research work. Special thanks to my girlfriend, Qianwen Xu, whose patience, love and support have made the completion of this thesis a reality.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

K	Number of Gaussian distributions
α	Learning rate of GMM
$w_{k,t}$	Weight of the kth Gaussian distribution for the tth frame
μ_t	Mean of a Gaussian distribution for the tth frame
σ_t^2	Variance of a Gaussian distribution for the tth frame
X_t	Value of a pixel of the tth frame
ρ	Updating parameter for a Gaussian distribution
$\eta(\cdot)$	Gaussian probability density function
x_i	Feature vector of the ith training sample
N	Number of the training samples
s_i	Class label of the ith training sample
R^m	m-dimensional feature space
w^T	Normal vector of a hyperplane
b	Bias of a hyperplane
a_i	Lagrange multipliers
γ_i	Slack variables

χ_i	Slack variables
C	Parameter for balancing the training error and margin in SVM
$K(\cdot)$	Kernel function
$F(\cdot)$	Classification decision function
y_p	Feature vector of the pth pixel to be classified
F_t	Foreground mask for the tth frame generated by GMM
F_m	Foreground mask generated by merging F_{t-2}, F_{t-1} and F_t
$\cdot$	Pixel-wise AND operation
F_{om}	Foreground mask generated by morphological opening on F_m
K_F	Number of training samples of foreground class
K_B	Number of training samples of background class
R	Region that convers the foreground in F_{om}
I_G	Gray level image of the current frame
B_G	Gray level background image generated by GMM
I_{DG}	Absolute difference image between I_G and B_G
I_{DH}	Absolute hue difference image
I_{DS}	Absolute saturation difference image

I_{D0}	A difference image combining I_{DG}, I_{DH} and I_{DS}
I_D	Feature image
IF_t	Gray level foreground image produced by GMM
I_{DT}	Temporal difference image between IF_t and IF_{t-2}
ε_{erode}	Coefficient for erosion operation
ε_{dilate}	Coefficient for dilation operation
O	Object mask
S	Shadow mask

LIST OF ABBREVIATIONS

GMM	Gaussian mixture model
LBP	Local binary patterns
TPF	Texture pattern flow
SVM	Support vector machine
2-D	Two dimensional
RGB	Red Green Blue
HSV	Hue Saturation Value
FP	False positive
TP	True positive
FN	False negative
FAR	False alarm rate
TRDR	Tracker detection rate

Chapter 1

Introduction

1.1. Background

The tremendous increase in the recent years of vision-based applications, such as video surveillance, contents based video analysis, intelligent traffic monitoring, analysis of sports videos, etc., has necessitated resolutions of many challenging problems. Segmentation of objects of interest from a video sequence is one of the most basic and essential operations of vision-based systems. A good segmentation of moving objects is crucial for their analysis and for the understanding of their actions. A typical frame of a video sequence usually consists of a foreground comprising the objects and a background, which is the scene of the frame from which the objects have been removed. Dynamic background such as swaying tree branches, waving water and cast shadows in a complex scene makes the task of object segmentation a very challenging problem.

1.2. Literature Review and Motivation

In this section, a brief review of the work in the literature for segmentation of moving objects is presented and the motivation for the work undertaken in this thesis is given. There are a number of different methods that have been developed in the

literature for the segmentation of moving objects in video sequences. These methods can be divided into two categories: image difference based methods and statistical model based methods.

Since the pixel values of objects, in general, differ from those of the background, the pixel values between successive frames change due to the motion of the objects. This fact can be used to segment an object from the frame of a sequence. Methods in [1-5] are examples of object segmentation based on this principle. These methods have not proven to be very effective if the frame rate of the sequence is not sufficiently high or the background is not static. There are other factors, such as change in illumination between frames that also affect the performance of these methods. In order to overcome some of these problems, background modeling based methods, in which the statistical information of the previous frames is used to build a background model for the current frame [6-32]. In [6-8], the problem arising from a low frame rate has been overcome by obtaining a model of the background as the average of some past frames. Stauffer and Grimson [9] have proposed a segmentation algorithm in which the background pixels have been assumed to have a Gaussian mixture model (GMM) distribution, whose parameters are made to change automatically with each new frame. This method works well with a slowly changing background. Even though the GMM method of [9] cannot deal effectively for segmenting objects from sequences with a rapidly changing background, it has provided good motivation for further work of object segmentation based on statistical background modeling.

Some other researchers have modified the GMM method by taking into account the information in the neighboring pixels instead of that of the current pixel alone [10-14]. These methods do perform better than the GMM does for video sequences in which the motions of the background pixels are not large. However, these methods still cannot effectively segment the objects from video sequences with a rapidly changing background because of the same reason as that of the original GMM method, namely, the inadequacy of the Gaussian mixture model in a rapidly changing background.

In order to effectively segment the objects from video sequences with rapidly changing backgrounds, texture feature is used to model the background in [15-18]. In these methods, a vector consisting of the texture features of pixels in the neighborhood of a pixel is used as a feature vector. Since these feature vectors for object pixels follow patterns different from that of the background pixels, these patterns are then used to distinguish the two types of pixels. Specifically, Heikkilä and Pietikäinen [18] have used a modified local binary pattern (LBP) to encode the texture feature of each pixel, and the histogram of the texture feature of the pixels in a neighborhood of the pixel under consideration is computed and used as a feature vector of the pixel. They model each background pixel as a set of feature vectors and for every new observation, the feature vector of an observation is matched individually with each of those of the corresponding background model. Based on the outcome of this comparison, the pixel in question is classified and the background feature vectors modified. These texture based background modeling methods in [15-18] can, to some extent, deal with the

sequences with a rapidly changing background; however, the use of only the texture feature in these methods, is generally not adequate to build a robust background model.

In [19-32], multiple features are used in order to achieve a better object segmentation performance. In particular, in [32], a scheme of texture pattern flow (TPF) has been devised to encode both the texture feature of a pixel and the temporal relationship between the pixels at same positions in different frames. The TPF is used to model each pixel to build the background model. This method, due to its efficient use of multiple features of the scenes in the background modeling, gives a performance superior to that of [18], in terms of its robustness to different kinds of backgrounds and its segmentation accuracy.

Methods in [6-32] mentioned above have similar schemes in that they all use the statistical information of some past frames to build a background model for the current frame, and by comparing the current frame and the background model, the segmentation of the objects is carried out. Another approach of segmenting the moving objects of video sequences is to classify the two types of pixels, namely the object pixels and those belonging to the background, by employing a trained classifier without using a background model. Methods in [34-38] are examples using such approach. In these methods, different features are first extracted and incorporated into a feature vector for a pixel in question. Some training samples of both the moving object and the background are manually selected with labels assigned to them to indicate their classes.

The feature vectors of these training samples are used to train the classifier, and finally, the classification decision for a pixel is made by the trained classifier based on the feature vector of this pixel. Although these methods yield good segmentation performance, they are rather restrictive in that a knowledge of the type of the objects and the backgrounds to be classified needs to be known *a priori* and the training samples have to be manually selected from other scenes having similar objects or backgrounds.

A method of segmenting an object usually results in segmenting the entire foreground that not only contains the object of interest but also its cast shadow. The reason for the shadow part of the foreground also getting segmented is that both the object and shadow carry the same motion information: The methods presented in [1-38] either do not focus on the shadow pixels or their schemes have embedded in them a stage for the removal of the shadow pixels from the segmented foreground. There are a few methods [39-42] in the literature for removing the shadow pixels from the segmented foreground. The methods in [39, 40], in particular, use only one feature, which is inadequate to distinguish an object pixel from a shadow pixel for their classification. On the other hand, the methods in [41, 42] use multiple features, thus improving the shadow removal performance over those using only one feature. However, in [41], the use of even two features is not able to yield a satisfactory segmentation accuracy, and in [42], the high-accuracy segmentation by using a large number of features is achieved at the expense of a large computational complexity.

From the foregoing discussion, it is clear that the methods described above are not able to accurately segment an object from the video sequences containing cast shadows and a rapidly changing background, or they have limitations arising from the requirement of manually selecting the training samples or *a priori* knowledge of the shape and size of the object. Therefore, it is imperative to develop cost-effective automated techniques for segmenting objects from the video sequences with cast shadows and a dynamic background.

1.3. Objective and Scope of the Thesis

This thesis is concerned with the development of computationally simple techniques for an accurate segmentation of objects with cast shadows from video sequences having a dynamic background. This objective is achieved by investigating the problem in two phases.

Since the cast shadow, if present in a video frame, is an integral part of the object, the first half of the study undertaken in this thesis focusses on developing an efficient technique for segmenting the entire foreground that consists of an object and the associated cast shadow. The proposed method takes advantage of the GMM [9] segmentation techniques and produces a scalar-valued feature image derived as a linear combination of the color components of the pixels used as multiple features. The

feature image is then used in an automated framework of support vector machine (SVM) [33] to segment the entire foreground.

In the second half of the thesis, a technique is developed to segment the object of interest from the segmented foreground. The method is developed based on the premise that the shadow removal capacity of a technique can be enhanced by using an appropriate number of suitable features.

1.4. Organization of the Thesis

This thesis is organized as follows.

In Chapter 2, a brief review of the background material relevant to the work undertaken in this thesis is carried out. Two commonly used signal models to represent color images are described. One of the earliest works on the modeling of dynamic background of video sequences, the Gaussian mixture model (GMM) [9], is presented. A very efficient data classification scheme, the scheme of support vector machine [33], is also described briefly, in this chapter.

In Chapter 3, the proposed foreground segmentation method is developed. The method of constructing a scalar-valued feature image and that of an automatic selection of training samples for applying it to a SVM-based classification scheme are described in detail. Experimental results obtained by applying the proposed and some of the

existing methods to a number of video sequences are presented to demonstrate the effectiveness and superiority of the method proposed in this chapter.

In Chapter 4, a simple and effective shadow removal method using multiple features is presented. The methods of producing three shadow masks, each based on a single feature, the corresponding incomplete object masks, and finally a complete object mask are described. Experimental results obtained by applying the proposed shadow removal method as well as some existing methods to a number of video sequences are presented in order to demonstrate the effectiveness of the method developed in this chapter.

Finally, Chapter 5 concludes the thesis by summarizing the work carried out in this study and by highlighting its contributions.

Chapter 2

Background Material

2.1. Introduction

In this chapter, a brief account of the background material necessary for the development of the work undertaken in this thesis is given. Discrete images are described or represented in terms of the gray levels or the values of color components as a function of pixel positions. This chapter starts with a brief description of representing discrete images. As features of signals are vital in any signal detection problem, a few commonly used features in image processing are next discussed. The Gaussian mixture model (GMM) is one of the simplest techniques to model dynamic backgrounds, and it is used in this thesis for a preliminary foreground segmentation, a brief description of this technique is also given. Finally, since the problem of foreground segmentation is treated as a data classification problem in this thesis, and the support vector machine is one of the most efficient techniques for data classification, we conclude this chapter by providing a brief description of this technique.

2.2. Image Signal Representations

A digital image is represented by its color components or gray levels as a function of the pixel position in a 2-D space. In the case of color image, the color signal at each pixel is generally decomposed and represented by three components. There are two ways of decomposing and representing, namely, RGB and HSV color spaces, widely used in image processing and display.

In the RGB color space, the intensity of a color pixel in an image is decomposed into three components-red, green and blue. In digital image processing, each of these components is usually quantized into 256 levels.

The decomposition in the HSV color space is very different from that of RGB. A color pixel is decomposed into hue, saturation and pixel intensity value components. The hue component represents the color element, the saturation component is about the purity of the color element, and the pixel intensity value represents the magnitude. By decomposing a color pixel in either color space, the chromaticity gets separated from the intensity, and the chromaticity information of this pixel is mainly concentrated in the hue and saturation components.

Sometimes, for simplicity, only the intensity information of a pixel is concerned. A color pixel is, therefore, converted into gray level. In this conversion, the color information of the pixel is discarded, and only the intensity information remained.

2.3. Image Feature Representations

In image processing, a single pixel value in the gray level or in a color component could be used as a feature of this pixel. However, the pixel value alone may not be able to provide sufficient information, thus, the information of neighborhood pixels is usually needed to represent the feature of the pixel.

Histogram could provide a statistical distribution of pixel signal strength in any region of an image. The histogram of pixel values in region of the neighborhood of a pixel is an effective way of representing the feature of the pixel. Unless the neighborhood regions of two pixels are similar in pixel signal strengths, the histograms of the two regions corresponding to the two pixels are generally significantly different. As shown in Figure 2.1, two pixels may have similar values, but the histograms of their neighborhood regions could be significantly different. This could be taken advantage of to distinguisht the two pixels.

Another way of representing the neighborhood information of a pixel is to use the gradients. By computing the gradients of the pixel values in a region centered at a pixel, the relations between this pixel and the neighborhood pixels can be obtained. Similar to histograms, the pixel gradients could also be used to distinguish two pixels.

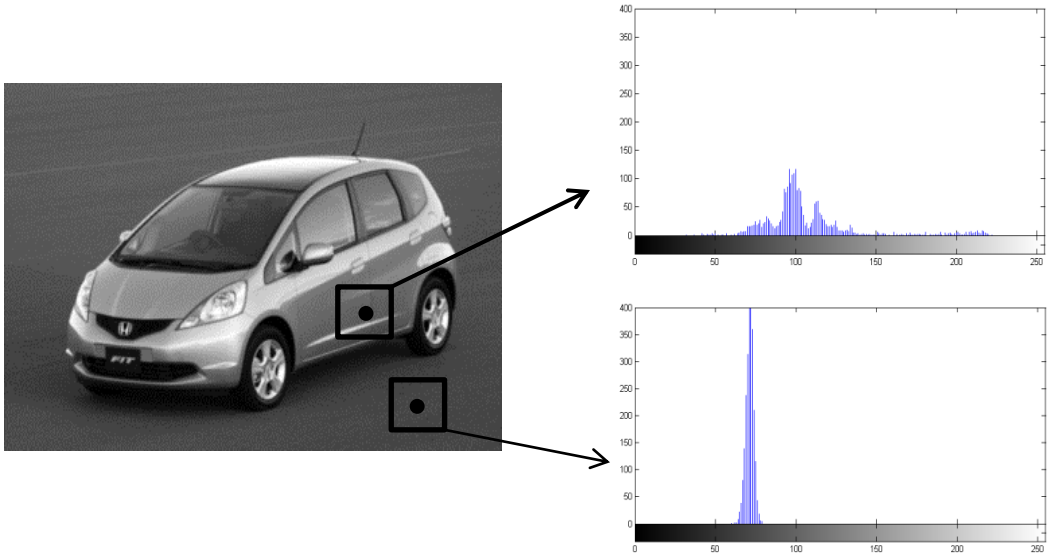


Figure 2.1 An illustration of feature representation of a pixel using histogram of the neighborhood regions.

2.4. Gaussian Mixture Model

There are a number of moving objects segmentation methods that have been proposed during the past years [1-32] and [34-38]. As described in the previous chapter, an approach to object segmentation is employing a background model which is obtained by using the statistical information of the past frames. The Gaussian mixture model (GMM) [9] proposed by Stauffer and Grimson is a background model based on the assumption that a background pixel has a Gaussian mixture model (GMM) distribution.

In the GMM method, each background pixel is assumed to have a set of K Gaussian distributions, where K is a small positive integer. The value of a pixel under

consideration in the current frame is matched individually with each of the K distributions. The weights of each distribution are adjusted as

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}) \tag{2.1}$$

where α is a pre-specified parameter called learning rate, and $M_{k,t}$ is 1 if the pixel under consideration first matches the *k*th distribution and 0 for the remaining distributions.

If the pixel under consideration is found to match a distribution, then the mean and variance of this distribution are updated, using the mean and variance of the corresponding pixel in the previous frame and the current pixel value, as

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{2.2}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^2 \tag{2.3}$$

where μ_t is the mean for this distribution, X_t is the value of the pixel under consideration, σ_t^2 is the variance for this distribution, and ρ is given by

$$\rho = a\eta(X_t|\mu_k, \sigma_k) \tag{2.4}$$

η being a Gaussian probability density function. The means and variances of the other unmatched distributions remain the same as that of the corresponding pixel in the previous frame.

The pixel under consideration is regarded as a background pixel if at least one match is found for it. On the other hand, if the value of the current pixel does not match with any of the *K* distributions, the pixel under consideration is regarded as a

foreground pixel. In this case, the least probable distribution, i.e., the one having the lowest ratio of its weight to variance among all the K distributions, is replaced by a new one with its mean equal to the current pixel value and a large variance, as well as a small weight in order keep the ratio of its weight to variance smallest compared to that of the other (K-1) distributions.

The weights of all the distributions of the pixel are normalized, and the K distributions are arranged in decreasing order of the ration of weight to variance. Finally, the value of a pixel of the background of a given frame is obtained as the weighted sum of the first J ($1 \leq J \leq K$) mean values for which the sum of the associated weights is equal or greater than a pre-specified threshold.

The GMM method has a good performance in segmenting foreground from sequences with a slowly changing background. Although it becomes less effective in segmenting objects from video sequences with a rapidly changing background, it nevertheless removes a relatively large number of background pixels from the frame.

2.5. Support Vector Machine

A support vector machine (SVM) [33], which is designed based on the principle of induction of *structural risk minimization* [43], is generally used for data classification. Although the use of SVM has been extended to multi-class data classification problems, and the extension is still an ongoing topic [46], the original use of SVM is to solve

classification problems of data belong to two classes. The segmentation of objects could be regarded as a two-class data classification problem, since the pixels of a frame belong either to the foreground or to the background. Thus, a two-class SVM is a suitable classifier that can be used for object segmentation.

Figure 2.2 shows the use of SVM in a data classification problem. There is a bunch of data from two classes mixed together to be classified. A set of training samples of the two classes are manually selected with their class labels specifying their classes are used to train the SVM classifier. The classification decisions for the data to be classified are made by the trained classifier.

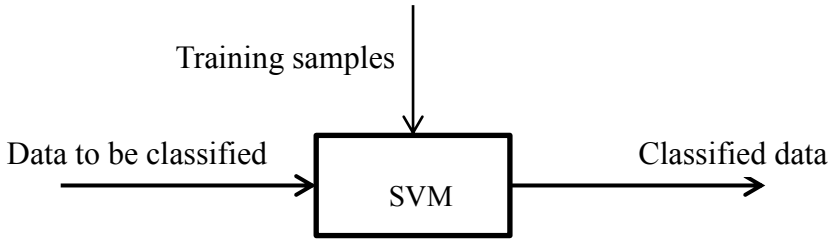


Figure 2.2 An SVM classifier.

Let $\{x_i, s_i\}$, $i = 1, 2, ..., N$. denote a set of N training samples, where sample is represented by an m-dimensional feature vector $x_i \in R^m$, and the class label of x_i $s_i \in \{+1, -1\}$. A hyperplane in the feature space can be described as

$$w^T x + b = 0 \tag{2.5}$$

where $w \in R^m$, and b is a bias. Assuming that the training samples are linearly separable, the aim is to define a hyperplane that divides the set of samples such that all

the points with the same label are on the same side of this hyperplane [47]. However, there may be more than one such hyperplane; thus, an optimal hyperplane that maximizes the minimum distance of the training samples from this hyperplane needs to be found, and this optimal hyperplane is used to classify the data.

This is an optimization problem for the parameters *w* and *b* corresponding to the optimal hyperplane. The solution of this problem for the linearly separable training samples could be obtained by formulating the proble as

$$\text{Minimize:} \quad L(w) = \frac{1}{2}||w||^2 \tag{2.6}$$

subject to:

$$s_i(w^T x_i + b) \geq 1,\ i = 1, 2,\ldots,N \tag{2.7}$$

To solve this problem, the non-negative Lagrange multipliers a_i are introduced, and the problem is transformed as

$$max_a\{\frac{1}{2}\left|\left|w\right|\right|^2 - \sum_{i=1}^{N} a_i(s_i(w^T x_i + b) - 1)\} \tag{2.8}$$

where $a_i \geq 0$.

The above formulation to this optimization problem is based on the assumption of zero tolerance to noise. Thus, a solution to problem may not exist in the case of noisy training samples. Taking into account a soft margin for noise, Cortes and Vapnik [33] have modified the formulation of the optimization problem as

$$\text{Minimize:} \quad L(w, \gamma_i) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\gamma_i \tag{2.9}$$

subject to:

$$s_i(w^T x_i + b) \geq 1 - \gamma_i, \quad i = 1, 2,\ldots,N \tag{2.10}$$

where γi are called slack variables which relate to the soft margin, and C is a parameter used to balance the margin and the training error. Similarly, by introducing the non-negative Lagrange multipliers ai, χi, the problem is transformed as

$$max_{a,\chi}\{\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\gamma_i - \sum_{i=1}^{N} a_i(s_i(w^T x_i + b) - 1 + \gamma_i) - \sum_{i=1}^{N}\chi_i\gamma_i\} \tag{2.11}$$

where $a_i, \chi_i \geq 0$.

However, if the training samples are not linearly separable, a hyperplane that could classify all the training samples with no error does not exist. In such a case, a non-linear SVM is used. A kernel is introduced in order to map the input vectors ($x_i \in R^m$) into a higher dimensional space thus making the data linearly separable, as shown in Figure 2.3. Therefore, the optimal hyperplane could be found in this higher dimensional space.

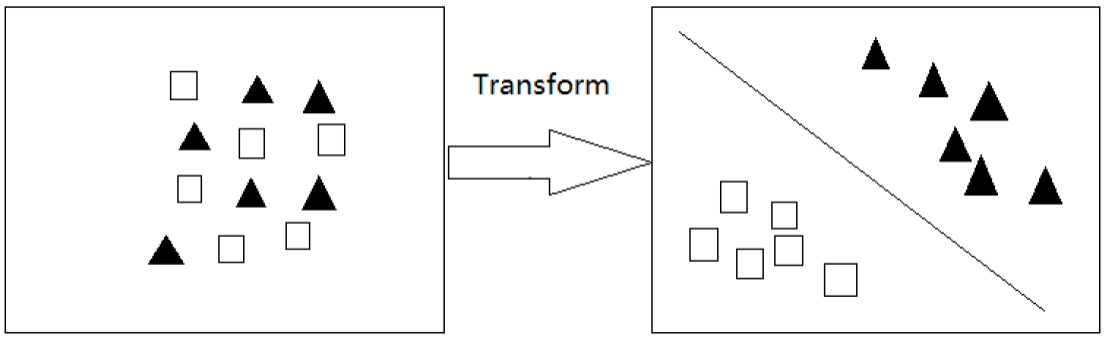


Figure 2.3 Transformation of non-linearly separable space to a separable space.

A function is a kernel when it satisfies the Mercer's condition [48]. Several typical kernel functions are as follows:

(i.) Gaussian Radial basis function: $K(x,y) = exp(-\frac{||x_i - x_j||^2}{\sigma})$.

(ii.) Polynomial function: $K(x,y) = (x^T y + \Theta)^d$.

(iii.) Sigmoid function: $K(x,y) = tanh(x^T y + \Theta)$.

(iv.) Inverse multi-quadric function: $\mathrm{K(x,y)} = \frac{1}{\sqrt{||x-y||^2 + c^2}}$.

In the above expressions the constants σ,Θ, d and c are the parameters of the kernels.

According to the optimization problem described above, the class label of the pth data is obtained by the decision function given by

$$F(p) = sign\left(\sum_{i=1}^{N} a_i s_i K(x_i, y_p) + b\right) \tag{2.12}$$

where $K(x,y)$ is a kernel function, y_p is the feature vector for the data to be classified, and the set of parameters $a = \{a_i, i = 1...N\}$ is obtained by maximizing the function

$$W(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i,j=1}^{N} a_i a_j s_i s_j K(x_i, x_j) \tag{2.13}$$

subject to the constraints:

$$0 \le a_i \le C \tag{2.14}$$

$$\sum_{i=1}^{N} a_i s_i = 0 \tag{2.15}$$

The SVM technique for classification is known to provide good results in situations where the number of training samples is limited [52].

2.6. Summary

In this chapter, the background material necessary for the development of an object segmentation technique has been presented. Two different representations of a color signal, namely RGB and HSV, have been described. In the former, a color signal is decomposed into red, green and blue components, whereas in the latter, the element of chromaticity of a color signal is separated from that of intensity. Both these representations will be used in the work of this thesis. Since the use of only pixel values as a feature is insufficient to classify pixels, the use of a feature that is not only based on the pixel value of the pixel in question but also on the values of the neighboring pixels, could be more appropriate. Since the histogram and gradients in the neighborhood of a pixel provides such a feature to the pixel, in this chapter, we have also described these two features. A basic method of segmentation and a method of data classification, namely, GMM and SVM, which are used in the proposed object segmentation scheme, have also been described.

Chapter 3

Foreground Segmentation in Video Sequences with a Dynamic Background

3.1. Introduction

A frame of a video sequence consists of a foreground comprising the moving objects and the associated cast shadows, and a background which is the scene of the frame from which the foreground has been removed. Segmenting moving objects from a video sequence is essential in many vision-based applications. A first step to segment the object of a frame is to segment the entire foreground of the frame. In cases where there are some moving elements other than the objects, the problem of foreground segmentation becomes very difficult, since very often the signal features in the foreground regions may have significant similarities with those in the background.

In this chapter, a novel two-stage algorithm for foreground segmentation is presented [53]. In the first step, the *Gaussian mixture model* (GMM) [9] is used to carry out a coarse segmentation, that is, to classify the pixels into two groups such that the first one consists of moving pixels belonging to the foreground or to the moving parts of the background, and the second one consists of the remaining image pixels. The pixels resulting from the first stage are then classified in the second stage to identify those belonging to the foreground. The pixel classification in this stage is carried out in

the framework of a support vector machine (SVM) [33]. However, the process of SVM is not applied to the signal values of the pixels to be classified, but to those of a feature image. A new method is proposed to extract multiple features from the original image frames to create a feature image in order to conduct the SVM classification.

The chapter is organized as follows. In Section 3.2, the architecture of the proposed scheme for the foreground segmentation is first presented as an interconnection of various modules. The architecture with its modules and their interconnection lays a foundation for the proposed schemes and forms the basis for the design of its two important modules, namely a training pixel selection module and a feature extraction module. A detailed and systematic development of these two modules as well as that of the SVM module is carried out. In Section 3.3, subjective and objective results of foreground segmentation are provided by applying the proposed technique to a number of benchmark sequences and compared with those obtained by using some existing schemes. Finally, in Section 3.4, the work presented in this chapter is summarized and some of the important attributes of the proposed method highlighted.

3.2. Proposed Method

In the proposed method, the segmentation of foreground from a sequence is obtained in two stages: GMM is used in the first stage to remove most of the

background pixels from the scene; in the second stage, an SVM-based segmentation scheme is applied to the foreground obtained from the first stage using the information from the outputs of the first stage as well as from the original sequence in order to get a final foreground. A schematic of the proposed method is shown in Figure 3.1.

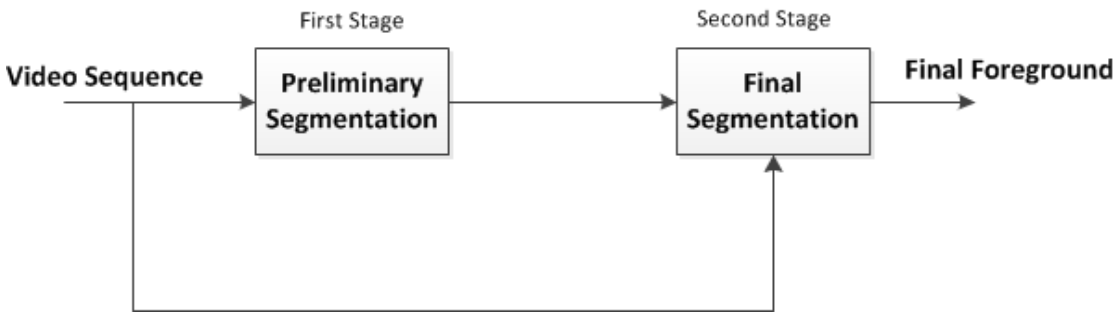


Figure 3.1 Two stages of the proposed segmentation scheme.

3.2.1. GMM-Based Foreground Segmentation

The objective in the first stage is to provide a segmented foreground output that contains as few background pixels as possible. Since the GMM method is capable of facilitating this goal relatively inexpensively, this method is adopted in the first stage of the proposed scheme. In the GMM method each background pixel is assumed to have a set of K Gaussian distributions, where K is a small positive integer. The value of a pixel under consideration in the current frame is matched individually with the distributions in a sequence from the first to the Kth. If the k^*th distribution ($1 \le k^* \le K$) is a matched distribution, then the mean and variance of the k^*th distribution is updated using the mean and variance of the corresponding pixel in the previous frame and the current pixel value. The other unmatched distributions of the pixel are kept the same as that of

the corresponding pixel in the previous frame. The pixel under consideration is regarded as a background pixel if at least one match is found. The weights of the matching distributions are updated, while those of the other distributions are reduced by a constant factor. On the other hand, if the value of the current pixel does not match with any of the K distributions, the pixel under consideration is regarded as a foreground pixel. In this case, the least probable distribution, i.e., the one having the lowest ratio of its weight to variance among all the K distributions, is replaced by a new one with its mean equal to the current pixel value and a large variance as well as a small weight in order keep the ratio of its weight to variance smallest compared to that of the other (K-1) distributions. The weights of all the distributions of the pixel in either case, that is, matched or unmatched, are normalized, and the K distributions are re-ordered according to the value of their weight to variance. Finally, the value of a pixel of the background of a given frame is obtained as the weighted sum of the first J ($1 \leq J \leq K$) means for which the sum of the associated weights is equal or greater than a pre-specified threshold.

Since the GMM method integrates in it the information on the pixel history, it can effectively deal with slowly-changing backgrounds. On the other hand, for scenes with a dynamic background, a significant amount of background pixels still remain as part of the segmented foreground image obtained from using the GMM method. Figure 3.2 is an example of the results of segmenting a foreground from a sequence with a dynamic background by using the GMM method. Figure 3.2(b) shows the background image

and Figure 3.2(c) the binary foreground mask corresponding to the original frame shown in Figure 3.2(a). It is clear from Figure 3.2(c) that there are a large number of pixels belonging to the dynamic background that are falsely detected as foreground pixels, thus making a further segmentation a necessity to remove these moving background pixels.

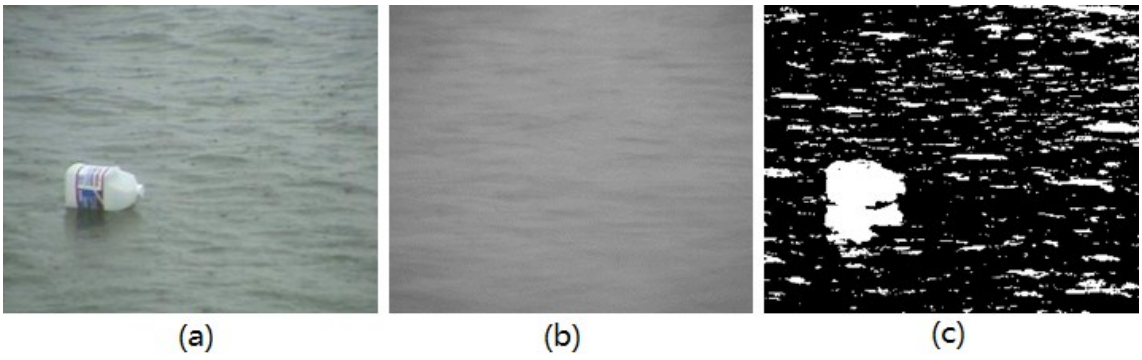


Figure 3.2 (a) Original frame. (b) Gray level background image generated by GMM. (c) Binary foreground mask obtained from GMM.

3.2.2. Data-Classification-Based Foreground Segmentation

In the second stage, the objective is to remove the moving background pixels from the segmented foreground obtained from the first stage. In the segmented foreground, there are two types of pixels mixed together: pixels belonging to the foreground and those from the background that are moving. Thus, the overall segmentation objective could be achieved by classifying these two types of pixels. The main idea used in the second stage is to treat the segmentation problem of this stage as a data classification problem.

Classification of data is usually carried out by making decisions based on some features of the data. The features used for classification are, therefore, vital for the classification performance. The difference between the foreground and the moving background pixels in regard to a feature may not be significant enough to distinguish them accurately. In the proposed method, a spatiotemporal feature image using multiple features of the image is constructed in order to make the difference between these two entities more pronounced for classification. For the purpose of classification, a small but equal number of image pixels called the training samples (samples with known classes) are selected from each of the two classes. The image pixels with their features specified by the feature image are then classified using classification knowledge of the training samples in a support vector machine (SVM) classifier. In the proposed scheme, a method is also developed for an automatic generation of the training samples. A schematic containing different modules to perform the various tasks of the proposed scheme for foreground segmentation is shown in Figure 3.3.

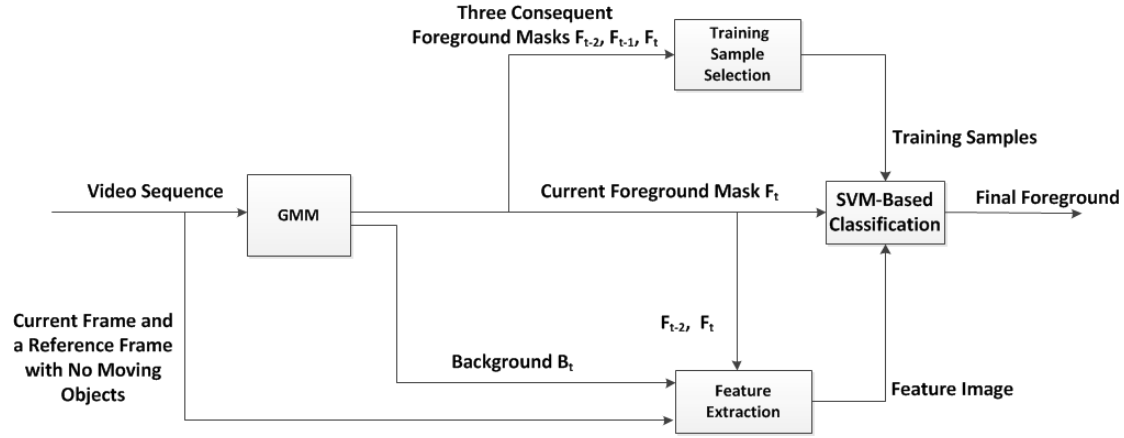


Figure 3.3 Scheme of proposed method.

A. Training Sample Selection

In order to train a classifier, certain number of samples, each labeled with its class, is needed. In [34-38], a priori knowledge of the type of the objects and the backgrounds to be classified needs to be known and then the training samples are manually selected from other scenes having similar objects or backgrounds. This type of the selection scheme for the training samples makes the application of a segmentation method rather restrictive. In the proposed scheme, a selection of the training samples is carried out automatically by the Training Pixel Selection module. For the purpose of this selection, we use the segmented foreground masks obtained from GMM.

As seen from Figure 3.2(c), the size of the cluster of pixels representing the real foreground is larger than those corresponding to the moving background pixels. This observation is used to obtain a mask that consists of pixels that predominantly belong to the foreground. Such a mask can be obtained as

$$F_m = F_{t-2} \cdot F_{t-1} \cdot F_t \tag{3.1}$$

where F_{t-2}, F_{t-1} and F_t denote the foreground masks obtained from GMM corresponding to the current and previous two frames, and $\cdot$ represents pixel-wise AND operation. Figure 3.4 shows an example of forming such a mask F_m. It is seen from this figure that the moving foreground has a large region in this mask, and only a much smaller number and size, of the regions corresponding to the moving background pixels in comparison to that in the mask F_t.

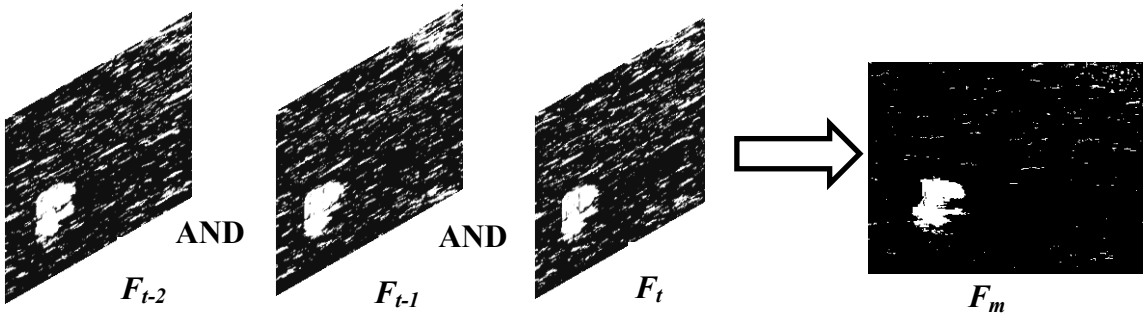


Figure 3.4 Generation of F_m using three successive foreground masks.

A morphological opening operation [61] can now be applied to F_m in order to further remove from it the background pixels and to obtain the final mask F_{om} that, as shown in Figure 3.5, consists overwhelmingly of the pixels representing the foreground. The locations with a value of logic one in this mask represent the locations of the foreground pixels. Note that in process of removing the background pixels falsely retained in F_m, some of the foreground pixels are also removed due to the erosion step of the opening operation. If K_F is the number of foreground training samples to be chosen from F_{om}, and N_F is the total number of foreground pixels in F_{om}, then K_F training samples are selected by uniformly sampling the foreground region in F_{om} at a spatial sampling rate N_F/K_F, thus ensuring the foreground training samples to be evenly distributed.

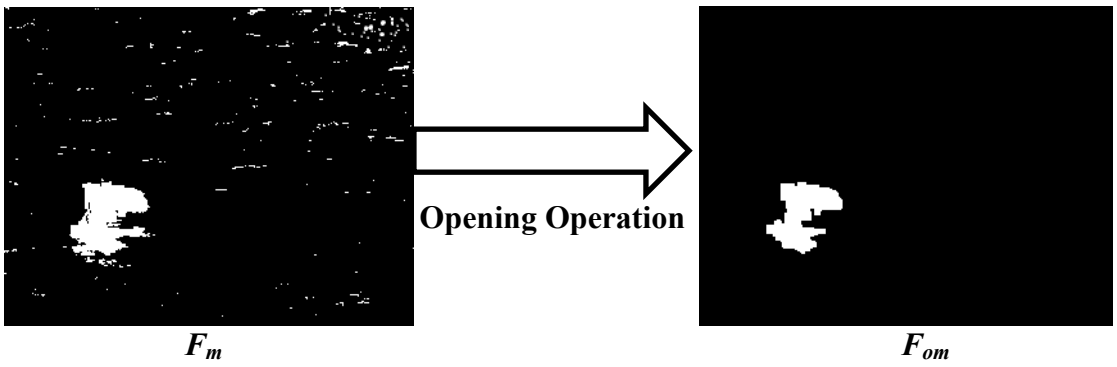


Figure 3.5 Morphological opening.

Next, a technique is developed for selecting background training samples from F_t. In order to ensure that all the samples in this selection are only the moving background pixels, a sufficiently large region R of F_t that possibly contains all the foreground pixels needs to be excluded from F_t before this selection. To identify such a region R, we make use of the mask F_{om}. This mask, as shown in Figure 3.6(a), is divided into blocks of appropriate size. From F_{om}, the region R is obtained as a polygon consisting of a contiguous set of blocks such that: (i) all the peripheral blocks in R do not have logic "1" pixel, from F_{om}, and (ii) each of the blocks interior to the peripheral blocks has at least one logic "1" pixel from F_{om}. Figure 3.6(b) shows such a region R corresponding to the foreground mask of Figure 3.6(a). With an appropriate choice of block size, this method should ensure that the region R has all the foreground pixels even if some of these pixels are not identified in F_{om}.

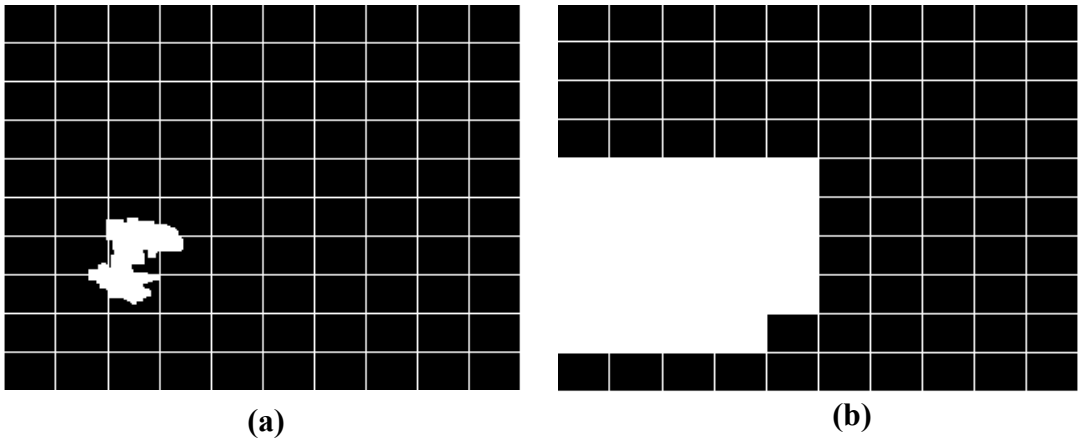


Figure 3.6 (a) F_{om} divided into small blocks. (b) Identified region R.

Once the region R has been identified in F_{om}, all pixels of F_t in its region corresponding to R are made to have a logic "0" value, giving a mask F_b, as shown in Figure 3.7. From this mask, K_B pixels are selected as background training samples by uniformly sampling the pixels with logic "1" in F_b.

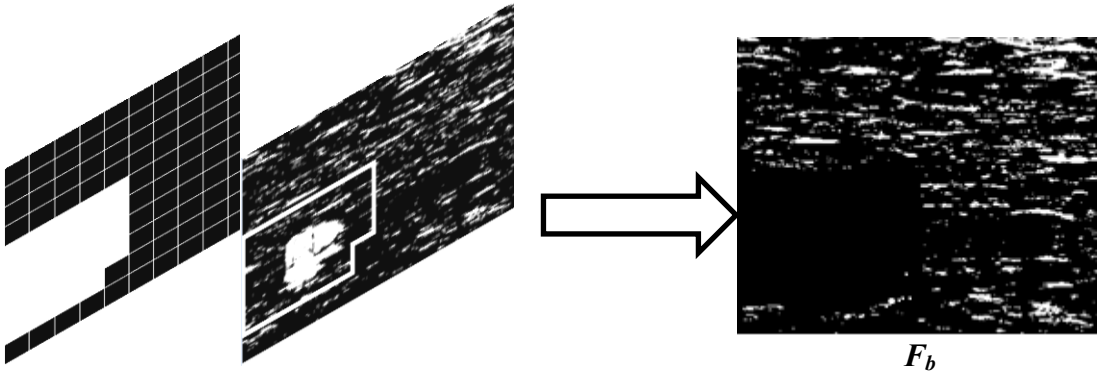


Figure 3.7 Generation of the mask F_b.

B. Feature Extraction

Data is usually classified based on the differences in a feature or features of the data belonging to the different classes. As mentioned earlier, for the present problem,

there are only two classes: moving foreground pixels and those belonging to moving background. In classifying these two types of pixels, we must choose a feature or a set of features that would make the two types of pixels distinctly different. With regard to only one feature, say the pixel gray level, pixels belonging to these classes may not be sufficiently different for classification. The objective of the feature extraction module of the proposed scheme is to construct a feature image based on multiple features such that with reference to such a feature image, the moving background pixels are significantly different from those belonging to the moving foreground.

Let I_G be a gray level frame and B_G the gray level background image corresponding to I_G. The value in $|I_G - B_G|$ corresponding to a pixel position in the moving foreground will be, in general, larger than that corresponding to a pixel position in the background. Since it is not possible to have B_G corresponding to I_G, we should consider the use of an approximate alternative. Figure 3.8(b) shows a gray level frame of the sequence that does not happen to have the object appearing in its scene. Thus, in reality it is not a background image corresponding to the gray level frame in question shown in Figure 3.8 (a). Figure 3.8(c) is the absolute difference frame. It is seen from this figure, that intensity of many background pixel positions is not as low as one would like to have. This is because of the fact that background frame is not corresponding to I_G. An alternative is to use the background frame corresponding to I_G generated by the GMM module. The background frame, shown in Figure 3.8(d), has a better correspondence with I_G in terms of the moving background pixels, since it is created

using the latter. Figure 3.8(e) shows the difference image I_{DG} when the GMM generated background image is used. A comparison between Figures 3.8(c) and (e) indicates that the use of the GMM-generated background image provides a better distinction between the moving foreground and background pixels.

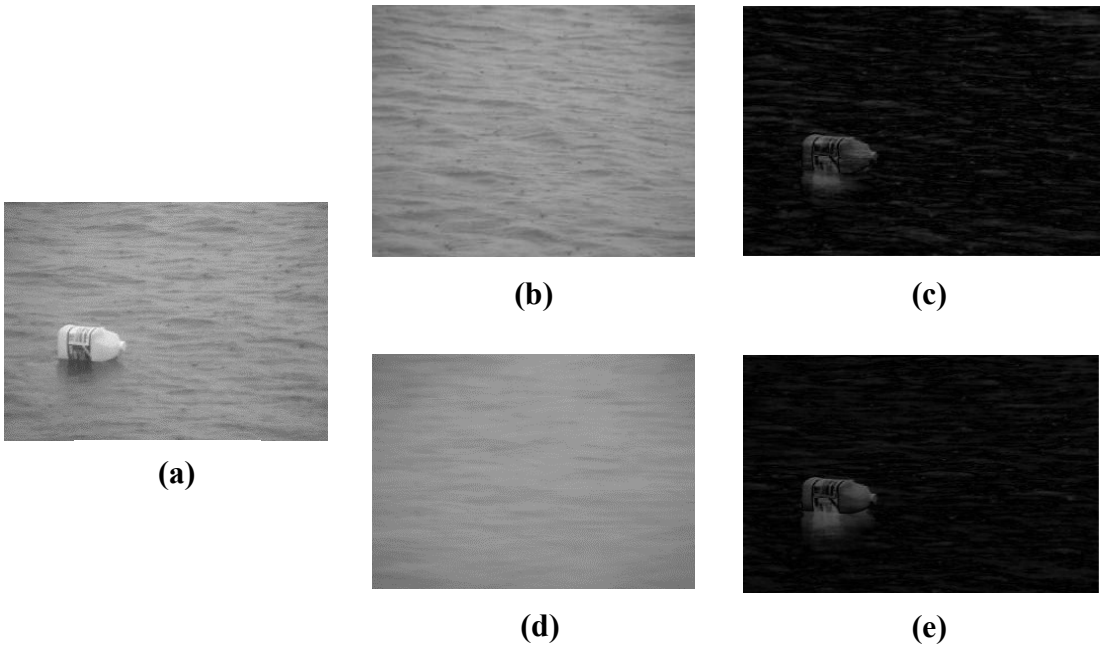


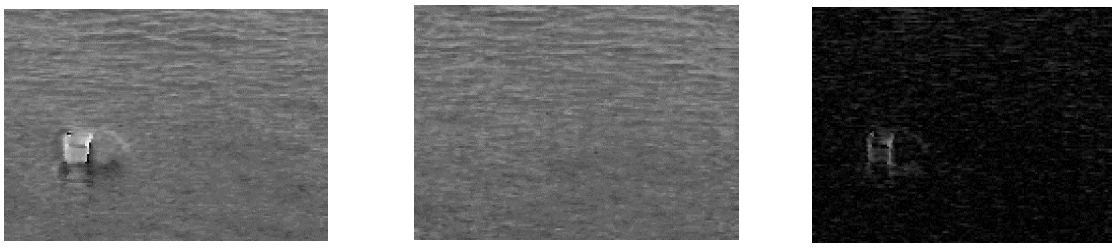
Figure 3.8 (a) Original gray level frame. (b) A gray level frame containing no object. (c) Difference between (a) and (b). (d) Gray level background image produced by GMM. (e) Difference between (a) and (d).

Next, we consider the color feature to construct difference images. In view of the fact that it would be computationally very expensive to construct a color background image corresponding to a frame under consideration, we choose a color frame that does not contain the moving objects in its entire scene as the color background image. Figure 3.9(b) shows such a frame that can be used as background image for the foreground segmentation of the image shown in Figure 3.9(a).

(a) **(b)**

Figure 3.9 (a) Current color frame. (b) Background image.

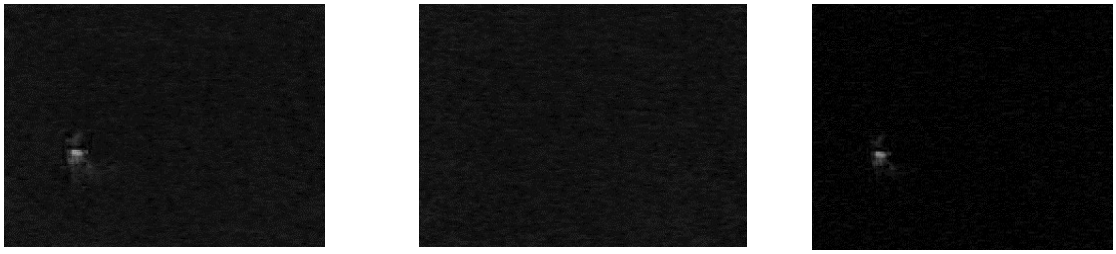
In the proposed scheme, the color feature is considered based on the HSV system, where the hue component H determines the color (hue) scale, S the saturation of the color and V the intensity value. Since the intensity has already been used in considering the gray level feature, we use the hue and saturation components of the color for constructing the difference images. Figures 3.10 (a) and (b) show the hue components corresponding to the color images shown in Figures 9(a) and (b), respectively. The absolute difference hue image I_{DH} is shown in Figure 3.10(c).



(a) **(b)** **(c)**

Figure 3.10 (a) Hue component of the current frame. (b) Hue component of the background. (c) Difference between (a) and (b).

Similarly, the saturation components of the foreground, background and absolute difference (I_{DS}) images corresponding to the images of Figure 3.9 are given by the images shown in Figures 3.11(a), (b) and (c).



(a) **(b)** **(c)**

Figure 3.11 (a) Saturation component of the current frame. (b) Saturation component of the background. (c) Difference between (a) and (b).

For the sake of uniformity, the pixels in the images of Figures 3.10 and 3.11 are re-quantized in order to have the same number of levels as the number of gray levels of the images in Figure 3.8, i.e., 256.

As in the case of the gray level difference image I_{DG}, in the hue and saturation difference images I_{DH} and I_{DS}, the pixel values corresponding to the background region are, in general, smaller than that corresponding to the foreground region. This difference in the pixel values of the two regions can be further enlarged by obtaining a weighted sum of the three difference images, given by:

$$I_{D0} = w_1 \cdot I_{DG} + w_2 \cdot I_{DH} + w_3 \cdot I_{DS} \tag{3.2}$$

where w_1, w_2 and w_3 are the weights of the individual difference images used to obtain the overall difference image. The idea behind this weighted sum instead of having simply a sum of the three difference images is to emphasize or de-emphasize a

difference image in obtaining I_{D0} depending on its ability to distinct the two types of pixels. In order to determine the values of the three weights, we proceed as follows. Using the positions of the foreground and background samples as determined in the previous section, we determine the median pixel values for the foreground and background sample pixels in the three difference images and obtain the following three ratios:

$$R_G = \frac{\hat{I}_{GF}}{\hat{I}_{GB}} \tag{3.3}$$

$$R_H = \frac{\hat{I}_{HF}}{\hat{I}_{HB}} \tag{3.4}$$

$$R_S = \frac{\hat{I}_{SF}}{\hat{I}_{SB}} \tag{3.5}$$

where $\hat{I}_{GB}$, $\hat{I}_{HB}$ and $\hat{I}_{SB}$ are the median gray, hue and saturation levels of the background samples, and $\hat{I}_{GF}$, $\hat{I}_{HF}$ and $\hat{I}_{SF}$ are the respective median values of the foreground samples. We have performed an experiment involving different kinds of frames and have observed that the values of R_G, R_F and R_S are approximately in the ratio 1:0.6:0.4. Accordingly, we choose the values of weights as $w_1 = 0.5$, $w_2 = 0.3$ and $w_3 = 0.2$. Implementing (2) using the difference images shown in Figures 3.8(e), 3.10(c) and 3.11(c) with these weights, we obtain the overall difference image I_{D0} shown in Figure 3.12. It is seen from this overall difference image that the contrast between the foreground and background pixels is, in general, more than in any of its three constituent difference images.

Figure 3.12 Constructed difference image I_{D0}.

Often the values of I_{D0} at the boundary of the foreground are lower than that in its interior. This may result in some of the pixels at the boundary of the foreground to be misclassified as background pixels. Thus, the expression for I_{D0} as given by (3.2) needs to be modified in order to avoid such a possibility.

The temporal difference between frames is used to enhance the values of the boundary pixels of the foreground. Let IF_{t-2}, IF_t denote two gray level foreground images produced by the GMM module corresponding to the current and previous to the previous frames. The difference image of these two frames is obtained as

$$I_{DT} = |\, IF_{t-2} - IF_t| \tag{3.6}$$

Since in I_{DT}, the values of the pixels at the boundary of the foreground region are, in general, larger than those of the non-boundary region, I_{D0} given by (3.2) is modified by adding to it the I_{DT} with a small weight:

$$I_D = w_1 \cdot I_{DG} + w_2 \cdot I_{DH} + w_3 \cdot I_{DS} + w_4 \cdot I_{DT} \tag{3.7}$$

The value of the weight w_4 is chosen to be smaller relative to the other weights so as not to increase the pixel values at the boundaries of the various regions representing the moving background pixels in I_{D0}. In our experiments, we have chosen w_4 to have value of 0.1. The values of the other weights are, therefore, modified as $w_1 = 0.45$, $w_2 = 0.27$, $w_3 = 0.18$. Figure 3.13 shows the overall difference image corresponding to the image of Figure 3.12 obtained by implementing (3.7). It is seen from this figure that the contrast between the values of the foreground and background pixels in Figure 3.13 is the same as that in Figure 3.12 except for the former's foreground boundary pixels, whose values have been enhanced. Since the pixel values of I_D will be used as the feature of a given frame to distinguish between its foreground and background pixels, we will call I_D a feature image.

Figure 3.13 Constructed feature image I_D.

Similar to other feature extraction techniques, in the proposed method for the construction of the feature image, multiple features have been used. However, the main advantage of this method lies in incorporating these multiple features into a single feature characterized by the pixel values of the feature image I_D. Regardless of the

nature of the foreground or moving background pixels and the number of features used, the proposed feature extraction method results in a feature image with scalar-valued pixels. This very characteristic of the feature image I_D, as we will see in the next subsection, can be used to simplify the classification of foreground and moving background pixels.

C. SVM-Based Classification

In this section, the pixels corresponding to the foreground mask produced by the GMM module are classified using the classification technique of SVM [33]. The SVM technique for classification is known to provide good results in situations such as ours, where the number of training samples is limited [52]. Since in our feature extraction scheme, multiple features have been incorporated into a single feature characterized by the pixel values of I_D, the complexity of the SVM based classification using I_D can be expected to be lower than those of the SVM classifiers that directly use multiple features. Even though the pixel values of the feature image I_D, in general, discriminate well between the foreground and moving background pixels, there are still a number of pixels both in the actual foreground and moving background with values such that the corresponding pixels could be misclassified. In order to reduce the risk for such a misclassification, in the SVM classifier, instead of using a pixel value of I_D, we use the histogram of the pixels in a window centered at the pixel in question. The use of a

suitable size window, in obtaining local histograms can be expected to reduce the risk of misclassification.

By using the feature image I_D, the SVM module first constructs local histogram for all the pixels classified as foreground pixels in the foreground mask produced by the GMM module. Let x_i ($i = 1, ..., N = K_F + K_B$) denote histograms of all the training samples and y_p the histogram of the pth foreground pixel to be classified in F_t. The ith training sample is assigned a label s_i, where s_i = "+1" or "-1", depending on whether this training sample belongs to the foreground or the moving background. The SVM technique makes an optimal classification of the pth pixel as

$$F(p) = sign\left(\sum_{i=1}^{N} a_i s_i K(x_i, y_p) + b\right) \tag{3.8}$$

where $K(x,y)$ is a kernel function, the set of parameters $a = \{a_i, i = 1...N\}$ is obtained by maximizing the function:

$$W(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i,j=1}^{N} a_i a_j s_i s_j K(x_i, x_j) \tag{3.9}$$

subject to the constraints:

$$0 \leq a_i \leq C \tag{3.10}$$

$$\sum_{i=1}^{N} a_i s_i = 0 \tag{3.11}$$

C being a pre-specified parameter, and b is a bias whose value is estimated using a_i's, x_i's and s_i's. The kernel function is chosen, as in [68], to be a measure of similarity between its two histogram arguments:

$$K(H_1, H_2) = \sum_{l=0}^{M} min\{h_l^{(1)}, h_l^{(2)}\} \quad (3.12)$$

where $h_l^{(1)}$ and $h_l^{(2)}$ are the lth bin of the histograms H_1 and H_2, respectively. This function is a kernel function under the premise that $\sum_{l=0}^{M} h_l^{(1)} = \sum_{l=0}^{M} h_l^{(2)}$, which is satisfied in our case.

According to (3.8), the weighted and signed similarities of the histogram of the pixel to be classified with that of each of the training samples are accumulated and then used to make the final classification decision for the pixel. The pixel p is classified as foreground or moving background pixel depending on whether $F(p)$ = +1 or -1.

3.3. Simulation Results and Performance Evaluation

In order to assess the proposed method, in this section, we apply it to segment the foregrounds of a number of video sequences with a dynamic background. The visual and quantitative results are compared with those obtained by using GMM [9] and two other methods presented in [18] and [32].

For our experiments, we set the number of training samples from the moving foreground and that from the moving background as $K_F = K_B = 400$. The parameter C in the SVM classifier is set as 500. In order to reduce the computational cost for the computation of histograms, we first obtain the integral histogram [49] of I_D, which has a linear complexity to the data length, and then compute the actual histograms quite simply.

We conduct two experiments to evaluate the performance of the proposed method. The first experiment is carried out on three video sequences, *Water* [54], *Watersurface* [55] and *Curtain* [56]. The first two of these are outdoor sequences with their objects being, respectively, close to and distant from the dynamic background, whereas the third one is an indoor sequence with a background of a flapping curtain. Based on the resolutions of the three sequences, the coefficients for the opening operation, ε_{erode} and ε_{dilate}, are set as (8, 6), (5, 3) and (5, 3), respectively. In each case, ε_{erode} is chosen to be greater than ε_{dilate} in an effort for F_{om} not to include background pixels. In the classification module, the choice of a proper window size for the histogram calculation is crucial. Too small a window size may not correctly reflect as to whether the neighboring pixels belong to the moving foreground or to the moving background. On the other hand, a choice of too large a window size may result in losing the local details. Based on this consideration and depending on the frame resolutions of the three sequences, we choose the window sizes of 11×11, 9×9 and 9×9, respectively, for the three sequences.

The results of applying the proposed segmentation method and the methods of [9], [18] and [32] on the *Water*, *Watersurface* and *Curtain* sequences are shown in Figures 3.14, 3.15 and 3.16.

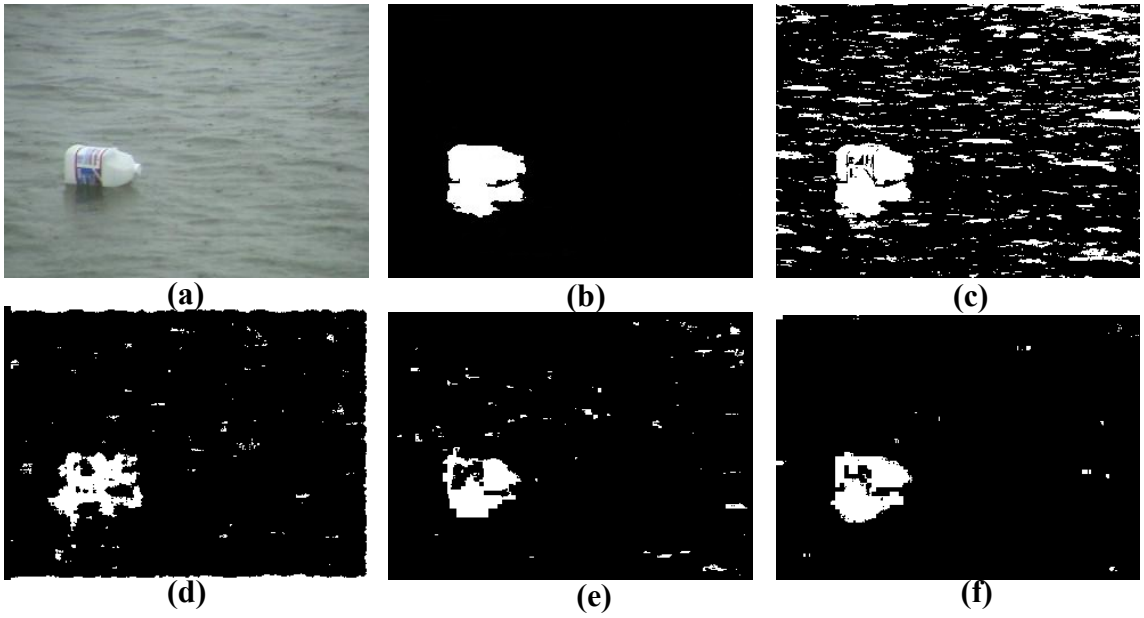


Figure 3.14 (a) Original 190th frame of the Water sequence. (b) Ground truth of foreground. (c) Segmented foreground mask using GMM [9]. (d) Segmented foreground mask using method [18]. (e) Segmented foreground mask using method [32]. (f) Segmented foreground using the proposed method.

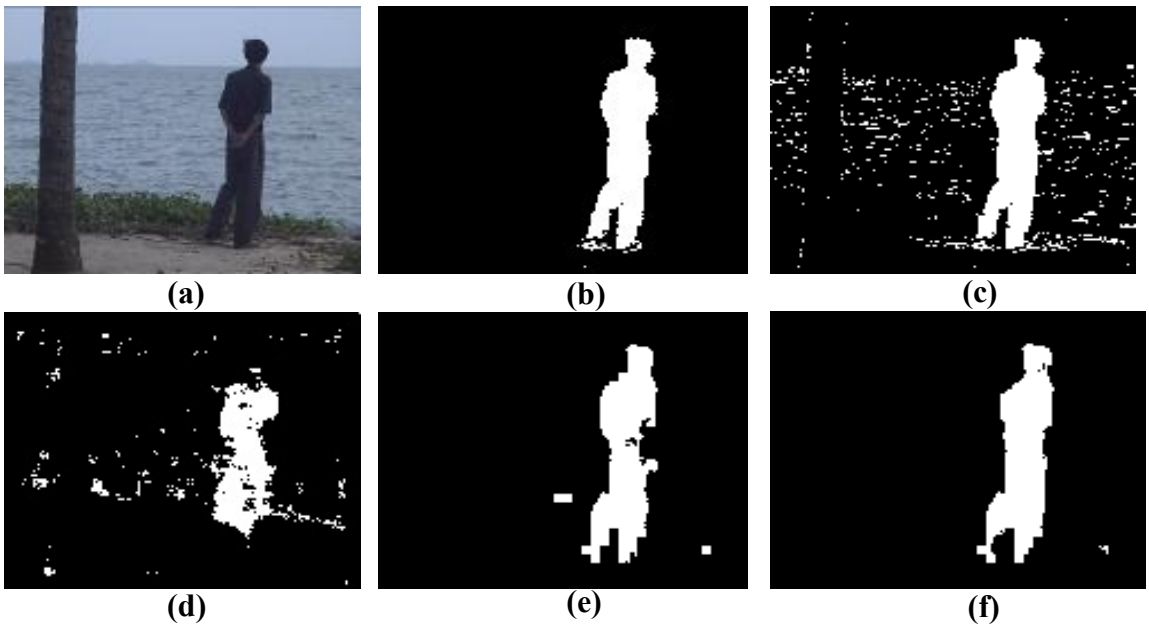


Figure 3.15 (a) Original 578th frame of the Watersurface sequence. (b) Ground truth of foreground. (c) Segmented foreground mask using GMM [9]. (d) Segmented foreground mask using method [18]. (e) Segmented foreground mask using method [32]. (f) Segmented foreground using the proposed method.

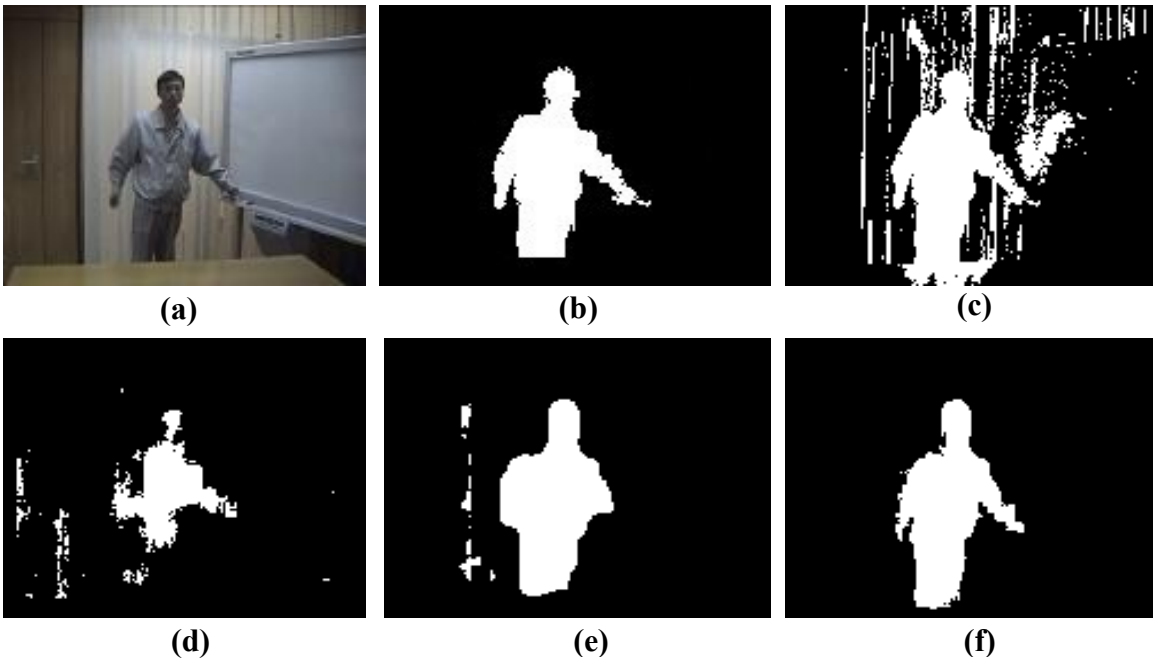


Figure 3.16 (a) Original 882th frame of the Curtain sequence. (b) Ground truth of foreground. (c) Segmented foreground mask using GMM [9]. (d) Segmented foreground mask using method [18]. (e) Segmented foreground mask using method [32]. (f) Segmented foreground using the proposed method.

Figures 3.14(a), 3.15(a) and 3.16(a) show the original 190th, 578th and 882nd frames of the three sequences, whereas Figures 3.14(b), 3.15(b) and 3.16(b) show the respective ground truths of the foregrounds. The images (c), (d), (e) and (f) in the three figures show the results of the foreground segmentation obtained by applying, respectively, the GMM method, the methods of [18] and [32], and the proposed method. It is seen from the illustrations shown in (c) of these figures that the GMM method even though segments almost all the foreground pixels, it includes in its segmentation many of the dynamic background pixels, since the method is quite sensitive to pixel motions. It is seen from illustrations (d) of the figures that even though the method [18], which is based only on the texture feature, is less sensitive to pixel motions, there are a number

of background pixels, both still and moving, as well as a number of foreground pixels that are misclassified. The method in [32] gives relatively better results, as seen from its segmentation results shown in the illustrations (e) of the figures; however, it still removes some of the foreground pixels or includes some of the pixels belonging to dynamic background. The proposed method is seen to provide the best results in terms of the completeness of the segmented foreground and exclusion of most of the background pixels.

For quantitative evaluation of the various segmentation methods, false positive (FP), the number of the background pixels that are detected as foreground, false negative (FN), the number of foreground pixels that are missed, false alarm rate (FAR), tracker detection rate (TRDR) [57] which is also named as sensitivity [69] as well as the specificity [69] are commonly used as performance metrics. In our measures, FP, FN, FAR and TRDR are used as the metrics. The metrics FAR and TRDR are defined as

$$FAR = \frac{FP}{TP+FP} \tag{3.13}$$

$$TRDR = \frac{TP}{TP+FN} \tag{3.14}$$

where TP is true positive, the number of correctly segmented foreground pixels.

For quantitative evaluations of the methods, 25 frames are randomly selected from the set of the frames containing the moving object in each sequence. The ground truth of the foreground corresponding to each of the selected frames is obtained manually. Each segmented foreground mask obtained by using a given method is compared with

the corresponding ground truth in order to obtain the values of the performance metrics and averaged over the 25 frames of each sequence. Table 3.1 and Table 3.2 give, respectively, the average numbers of false positives and false negatives per frame. It is seen from these tables that the proposed method provides the lowest values for the false positives, and the second lowest values, which are next to that provided by GMM, for the false negatives. The reason for the GMM method providing the lowest FN values can be understood in view of the fact that this method is very sensitive to pixel motions, thus classifying all the moving pixels as foreground, as indicated by the very large values of false positives provided by it.

Table 3.1 Average number of false positives per frame

	Method [9]	Method [18]	Method [32]	Proposed Method
Water	8024	3934	424	191
WaterSurface	779	749	246	172
Curtain	1980	529	369	289

Table 3.2 Average number of false negatives per frame

	Method [9]	Method [18]	Method [32]	Proposed Method
Water	214	1127	833	519
WaterSurface	95	596	192	180
Curtain	129	434	274	261

Tables 3.3 and 3.4 give, respectively, the average false alarm rate and tracker detection rate. From these tables, it is seen that the proposed method gives the lowest FAR, which means the ratio of the falsely classified foreground pixel to the total number of pixels classified as foreground pixels is the lowest. The proposed method also has the second highest TRDR, only next to that given by the GMM method. The highest TRDR provided by the GMM method is resulted from its sensitivity to motions of the pixels whether they belong to the foreground or background.

Table 3.3 Average false alarm rate

	Method [9]	Method [18]	Method [32]	Proposed Method
Water	0.69	0.58	0.17	0.08
WaterSurface	0.32	0.42	0.16	0.11
Curtain	0.41	0.24	0.15	0.09

Table 3.4 Average tracker detection rate

	Method [9]	Method [18]	Method [32]	Proposed Method
Water	0.96	0.73	0.75	0.86
WaterSurface	0.97	0.68	0.86	0.90
Curtain	0.98	0.82	0.84	0.89

We next provide the results of visual performance of the various methods by applying them to video sequences with more than one moving object having different shapes, sizes and velocities. In this experiment, we use the *Railway* [58] and *Campus*

[59] sequences, having outdoor scenes with two objects. The parameters (ε_{erode} , ε_{dilate}) for these two sequences are set as (8, 6) and (5, 3), respectively. The window size for the two sequences are chosen to be 11×11and 9×9, respectively. Figures 3.17 and 3.18 illustrate the segmented images for the two sequences produced by applying the proposed method and methods in [9], [18] and [32]. Figures 3.17(a) and 3.18(a) show the original 426th and 826th frames of the two sequences. The ground truths of the foreground segmentation for these two frames are shown in Figures 3.17(b) and 3.18(b), respectively. The images (c), (d), (e) and (f) of the two figures are the segmented images obtained by applying the methods in [9], [18], [32] and the proposed method, respectively. It is seen from these images that, as in the case of the single-object sequence, the proposed method results in a superior segmentation by providing the most complete foreground and suppressing almost all the background pixels.

The computation times of the proposed method and those of [5] and [6] are obtained by applying these methods to the 160 × 128 resolution *WaterSurface* and *Curtain* sequences on a Windows-platform PC with a 2.83GHz Intel Core Quad CPU and 8GB RAM using MATLAB codes. The results are shown in Table 5. It is seen from this table, that the proposed method on an average takes 44% more time than the method of [18] and 10% less than the method of [32]. Thus, the proposed method provides a segmentation performance superior to that of [32] with a reduced computational cost.

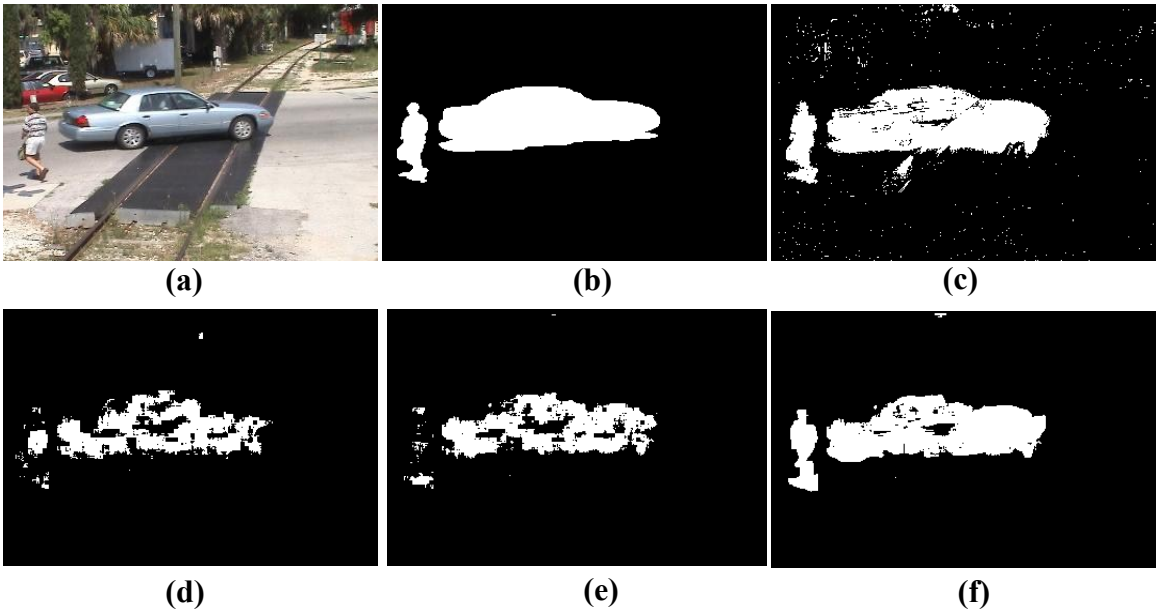


Figure 3.17 (a) Original 426th frame of the Railway sequence. (b) Ground truth of the foreground. (c) Segmented foreground mask using GMM [9]. (d) Segmented foreground mask using method [18]. (e) Segmented foreground mask using method [32]. (f) Segmented foreground using the proposed method.

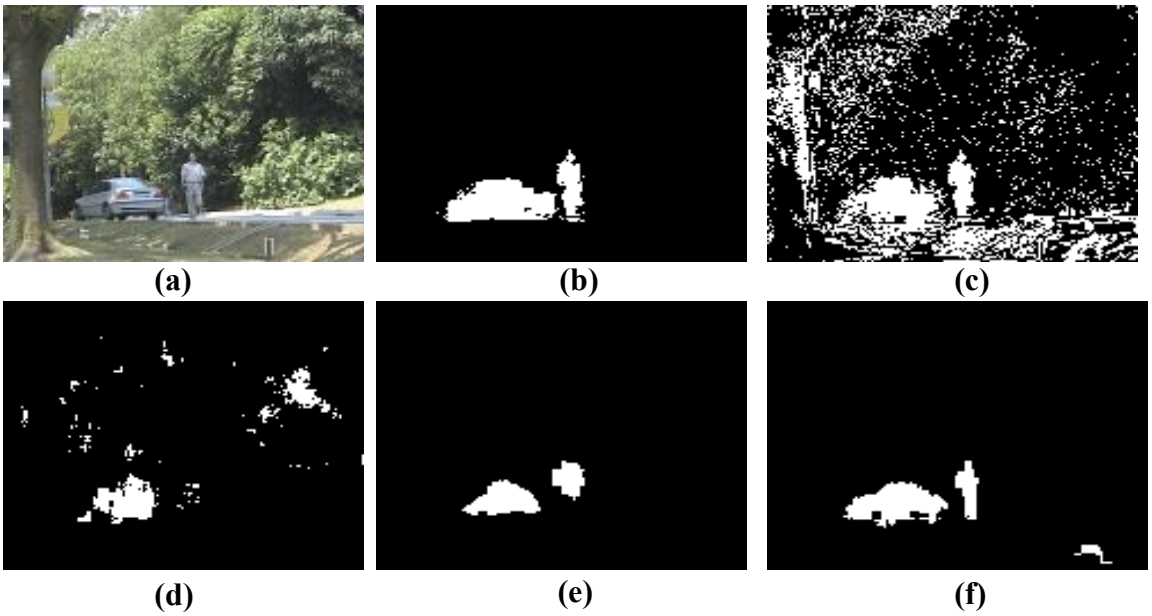


Figure 3.18 (a) Original 826th frame of the Campus sequence. (b) Ground truth of the foreground. (c) Segmented foreground mask using GMM [9]. (d) Segmented foreground mask using method [18]. (e) Segmented foreground mask using method [32]. (f) Segmented foreground using the proposed method.

Table 3.5 Average elapsed time per frame (Second)

	Method [18]	Method [32]	Proposed method
Sequence WaterSurface	7.52	11.75	10.38
Sequence Curtain	5.81	9.59	8.79
Average	6.67	10.67	9.59

3.4. Summary

In this chapter, a novel technique for segmenting the moving foreground from video sequences with a dynamic background has been presented. The segmentation problem has been treated as a problem of classifying the foreground and background pixels of a frame of the sequences using the pixel color components as multiple features of the images. The individual features representing the pixel gray levels, hue and saturation levels are first extracted and then recombined with suitable weights to form a scalar-valued feature image. Multiple features incorporated into a scalar-valued feature image has allowed to devise a simple classification scheme in the framework of the support vector machine classifier. Unlike some other data classification approaches for foreground segmentation in which *a priori* knowledge of the shape and size of moving foreground is essential, in the proposed method, training samples are obtained in an automatic manner. In order to assess the effectiveness of the proposed method, the new scheme has been applied to a number of video sequences with a dynamic background and the results have been compared with those obtained by using other existing

methods. The subjective and objective results have clearly demonstrated the superiority of the proposed scheme in providing a segmented mask that fits more closely with the ground truth than those provided by the other methods.

Chapter 4

Cast Shadow Removal Using Multiple Features

4.1. Introduction

In a video sequence, the cast shadow of a moving object becomes its integral part, since they both carry identical motion information. A simple scheme of removing the cast shadow from a segmented foreground, which consists of the pixels of the object and its cast shadow, is to identify an object pixel in the foreground based on the differences in the feature of the object and shadow, such as gray levels, colors or the gradients of pixel intensities in a neighborhood. However, such an identification becomes almost impossible in situations where the two classes of pixels are not sufficiently distinguishable with respect to the feature used for identification of the object pixels.

In this chapter, a novel shadow removal technique [60] based on using multiple features is developed. The rationale behind the use of multiple features for object segmentation is that an object pixel would not be indistinguishable from a shadow pixel simultaneously with respect to all the features used. In order to minimize the complexity of the proposed method, first, each feature is used individually to produce an incomplete object mask, and then the various incomplete object masks, each produced using a different feature, are merged into a single object mask. The merged

mask is a more complete object mask, since it can be expected that an object pixel not included in an incomplete mask is likely to be included in at least one of the other incomplete masks. In this chapter, the features used for developing the proposed method are gray levels, color and the gradients of pixel intensities. In Section 4.2, modules, each using a different feature, are designed to generate three shadow masks. The three shadow masks are then used to obtain the corresponding incomplete object masks. Finally, the three incomplete object masks are used by the next module to produce a complete object mask. In Section 4.3, simulation results using the proposed scheme are presented and compared with those obtained by using other methods in order to examine the effectiveness of the proposed method.

4.2. Proposed Method

Since the proposed method of cast shadow removal is based on gray levels and color and pixel gradients features of the shadows and objects of the images, the use of realistic modules of these features is essential. In order to develop useful and refined modules of these features, it is first necessary to analyze the objects and shadows in the context of these features.

(1) If the grey levels of the moving objects are similar to that of the shadows, it would be difficult to distinguish objects from their shadows, by using only their grey levels.

(2) If parts of the moving objects have colors similar to those of the corresponding background, then these parts of the objects would get removed when only the color feature is used to distinguish them.

(3) Generally, the gradient values of the object pixels and that of the cast shadow ones are different. However, if the gradient values of the moving object pixels and that of shadow ones are similar, such object pixels would be falsely misclassified as background when only the gradient values is used to distinguish them.

One way of removing the shadow from the foreground is to identify the shadow pixels and remove them from the foreground. From the above analysis, it is clear only one of the features is not sufficient to efficiently remove the shadows, and thus, one needs to use a combination of features for this task. Figure 4.1 shows two basic structures of the schemes in which multiple modules, each of which based on only one of the features, are used for shadow removal. In the architecture of Figure 4.1(a), the multiple modules are used in a sequential manner, whereas that of Figure 4.1(b) they are used in parallel. From the above discussions, it is clear that in either architecture, a useful information about the object could be lost or unnecessary information about the shadow could be retained by a single module. Accordingly, the sequential and parallel architectures of Figure 1 need to be modified so that the overall architecture is able to effectively and accurately remove the shadow by compensating the drawbacks of the individual modules. To this end, the parallel architecture of Figure 4.1(b) seems to be a

better candidate for such a modification, since each module operating on the same original foreground image can more easily produce results complementary to those produced by the other modules. We, therefore, develop a shadow removal scheme that is based on the parallel architecture of Figure 4.1(b).

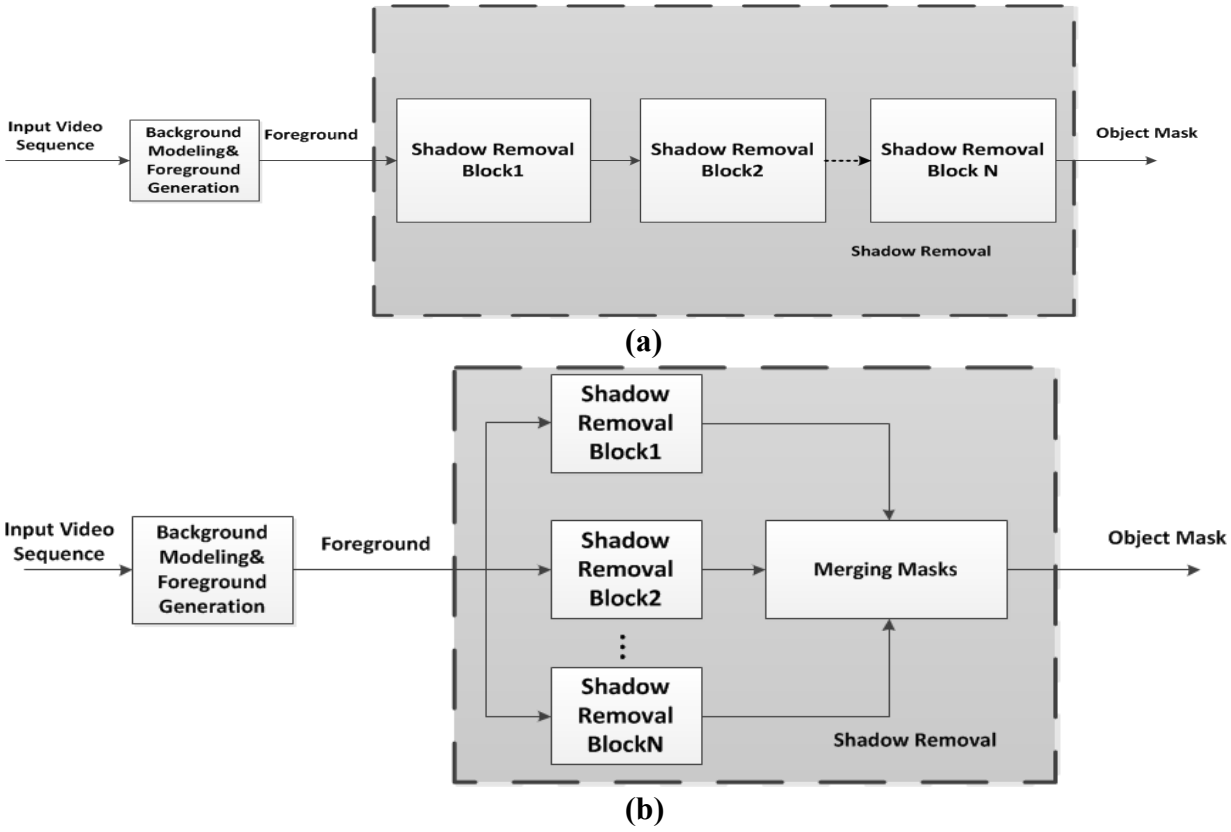


Figure 4.1 Schemes of shadow removal. (a) Sequential. (b)Parallel.

The proposed shadow removal method is shown in Figure 4.2. The input foreground image is assumed to have both moving objects and cast shadow pixels of the objects. First, three binary shadow masks are created by the three modules designed based on the gray level, color and pixel gradients features, respectively, which operate in parallel on the same foreground image. The output shadow masks resulting from

these modules are then used to create three different object masks by subtracting the individual shadow masks from the foreground mask obtained through a binarization operation of the foreground image. Finally, the three object masks are merged by a logical "OR" operation to generate the final object mask.

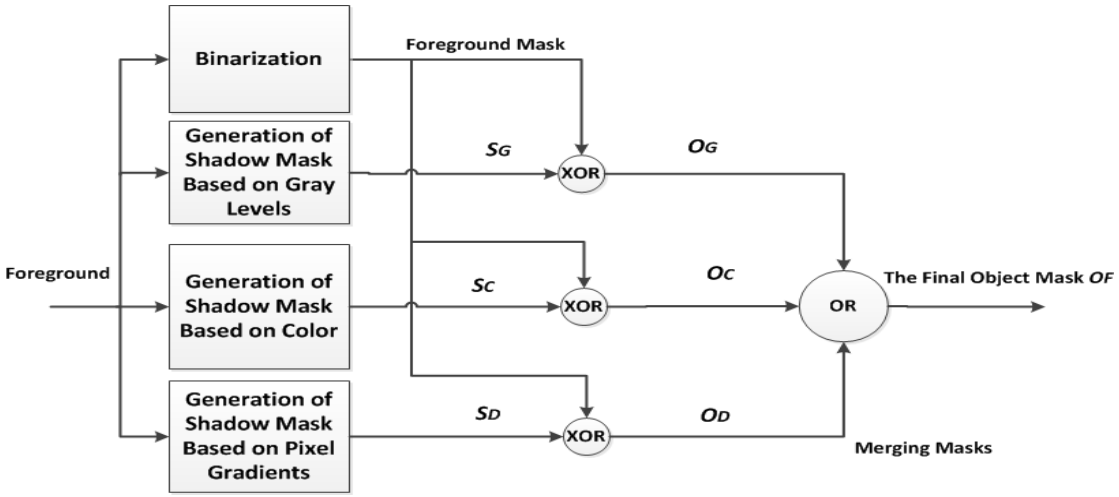


Figure 4.2 Scheme of the proposed method for shadow removal, where S_G, S_C and S_D denote the three shadow masks, and O_G, O_C and O_D denote the three subsequent object masks.

The objective of the proposed method is to generate a complete object mask that captures all the object pixels and does not include any pixel belonging to the shadow. Since the final object mask O_F is obtained via a logical "OR" operation of the three object masks, none of the object masks, O_G, O_C and O_D must include a shadow pixel and an object pixel must be included in at least one of these three object masks. In order to achieve this goal, the three shadow masks S_G, S_C and S_D must meet the following two requirements:

(1) Each shadow mask must capture all the shadow pixels;

(2) An object pixels must not be mistakenly detected as a shadow pixel in all the three shadow masks simultaneously.

In view of the fact that three different features are used to create the three shadow masks, the condition (2) is less stringent in the sense that it would likely be met more easily than the first one. Hence, we should make sure that each shadow mask captures all the shadow pixels even at the expense of some of the object pixels being mistakenly considered as shadow ones.

4.2.1. Detection of Shadow Pixels based on Gray Level

In the proposed method, the detection of shadow pixels using gray level information is based on the luminance enhanced method [39], and the procedure is turned in such a way that all the pixels in the cast shadows should be captured by the gray-level-based module.

It is observed that the gray level difference between the background and shadows is generally smaller than that between the objects and the background. Adding a small positive constant δ, to all the non-zero pixels in the foreground image of the *t* th frame $Y_F^{(t)}$ gives

$$Y_E^{(t)}(i,j) = Y_F^{(t)}(i,j) + \delta \tag{4.1}$$

If δ is appropriately chosen, the gray level difference between the background and shadow will become zero or get reduced, whereas, that between the background and the

objects still remain significant. If B_T denotes the background image of the t th frame, the difference between the modified foreground and the background is given by

$$Y_D^{(t)}(i,j) = |Y_E^{(t)}(i,j) - B_T(i,j)| \tag{4.2}$$

A threshold ε_T is calculated, based on Y_D^t and B_T, as

$$\varepsilon_G(i,j) = \varepsilon_0 * \left(1 + {Y_D^{(t)}(i,j)}/{(B_T(i,j)+1)}\right) \tag{4.3}$$

where ε_0 is a small positive constant. Then, a shadow mask is computed as

$$S_G^{(t)}(i,j) = \begin{cases} 1, & if\ Y_D^{(t)}(i,j) < \varepsilon_G(i,j) \\ 0, & otherwise \end{cases} \tag{4.4}$$

The gray level difference between the background and shadow is generally not a constant, and varies from area to area. Thus, using a very small δ would probably not reduce the difference $Y_D^{(t)}$ given by (4.3) to zero. However, a large value δ may produce a $Y_E^{(t)}$ in which the gray level difference between some pixels of the objects and those of the background is comparable to the gray level difference between the shadow and the background. In other words, $Y_D^{(t)}(i,j)$ at a pixel position in the object may be similar to that in part of the shadow. Ideally, the value of δ should be made signal dependent, instead of keeping it constant. This, however, would increase the complexity of the proposed method. As mentioned previously, the emphasis in designing the shadow mask is on a complete coverage of the shadow pixels, even at the risk of mistaking some of the object pixels as shadow pixels.

4.2.2. Detection of Shadow Pixels Based on Color Information

The shadow mask $S_G^{(t)}$as obtained in Section 4.2.1 has been designed to capture all the shadow pixels; but, it may also include some of the object pixels having gray levels similar to that of the corresponding background. Thus, the subsequent object mask $O_G^{(t)}$resulting from the shadow mask will miss these pixels. The color feature of these pixels, however, may be different from that of the background. Thus, we can make use of the color feature in the formation of the shadow mask without mistaking such object pixels as the shadow ones.

It is known that under normal illumination condition, the color composition of an individual pixel inside the main body of a shadow remains approximately the same as that of the corresponding pixel in the background. This characteristic of shadow formation has been referred to as color invariance property [40]. The ratio of the three color components, R, G and B, of a pixel satisfying this property is the same as that of the corresponding background pixel. However, the pixels on or near the boundary of the shadow, in general, do not satisfy this property. Therefore, a shadow formed under normal illumination condition have two regions: the umbra region consisting of the majority of the pixels in the interior of a shadow that satisfy the color invariance property, and the penumbra region consisting of a minority of the border shadow pixels for which this property is not satisfied. Using the color information of the un-shadowed background, one can identify the umbra shadow pixels in the foreground. In the

proposed method, we use the RGB color space instead of the HSV one in order to simplify the computation.

Representing the three color components of the background (i.e., the scene without the object or shadow) by R_B, G_B and B_B, and those of the foreground of the t th frame by $R_F^{(t)}$, $G_F^{(t)}$, $B_F^{(t)}$, we can calculate the following three ratios corresponding to each of the three color components.

$$R_R^{(t)}(i,j) = {R_F^{(t)}(i,j)}\Big/{R_B(i,j)} \tag{4.5}$$

$$R_G^{(t)}(i,j) = {G_F^{(t)}(i,j)}\Big/{G_B(i,j)} \tag{4.6}$$

$$R_B^{(t)}(i,j) = {B_F^{(t)}(i,j)}\Big/{B_B(i,j)} \tag{4.7}$$

Since the three ratios for a pixel in an object vary considerably from one another than that in the case when the pixel is in the shadow, we can use the variance of the three ratios to distinguish an object pixel from the shadow pixel in the foreground. Denoting the variance of the three ratios corresponding to an (i, j)th pixel in the t th frame by $V^{(t)}(i,j)$, a shadow mask based on color feature can be constructed as

$$S_C^{(t)}(i,j) = \begin{cases} 1, & if\ V^{(t)}(i,j) \le \varepsilon_C \\ 0, & otherwise \end{cases} \tag{4.8}$$

where ε_C is a pre-specified small positive constant. The object pixels having gray levels similar to the corresponding background pixels, and therefore, mistakenly

included in $S_G^{(t)}$ will not be included in the shadow mask $S_C^{(t)}$ as long as they have color compositions different from the corresponding background pixels.

4.2.3. Detection of Shadow Pixels Based on Pixel Gradients

If an object pixel has both the color composition and gray level similar to those of the corresponding background pixel, this pixel will be included in both gray level and color-based shadow masks. Therefore, another technique for a shadow mask construction based on a feature that can handle this kind of situation needs to be developed. We now develop such a technique based on pixel gradients feature of the image.

It is known that the pixel gradient information of the object is usually very different from the background. It is assumed that there are more gray level variations in the object than that in the background. One can, therefore, use the gradient of the foreground image to detect shadow pixels. We use the SOBEL operator to calculate the gradients. Then, the magnitude of the gradient $G^{(t)}$ is used to construct a shadow mask as

$$S_D^{(t)}(i,j) = \begin{cases} 1, & if G^{(t)}(i,j) \leq \varepsilon_D \\ 0, & otherwise \end{cases} \tag{4.9}$$

where ε_D is a pre-specified positive constant.

With this simple method for the construction of the shadow mask $S_D^{(t)}$, the object pixels having low gradient values will be mistakenly included in $S_D^{(t)}$ as shadow pixels,

and therefore, would be missed in the subsequent object mask $O_D^{(t)}$. However, this is acceptable considering that such object pixels are probably included in one or both of the other object masks, $O_G^{(t)}$ and $O_C^{(t)}$.

Remark: As discussed earlier, our objective in the construction of the shadow masks is to capture simultaneously in all of them as may shadow pixels as possible. The achievement of this goal can be facilitated by choosing the threshold parameters ε_0, ε_C and ε_D to have reasonably large values. However, a choice of large values for these parameters may lead to many of the object pixels to be mistakenly included in one or more of the shadow masks. This would be acceptable as long as the same object pixel is not included in the shadow masks S_G, S_C and S_D simultaneously. Note that an object pixel being included in all the shadow masks would happen in the case when gray level and color composition of such a pixel are the same as that of the corresponding background pixel as well as this object pixel has a low gradient value. Our proposed technique is, therefore, based on the assumption that such situation would occur infrequently for most of the real images.

4.2.4. Creation of the Final Object Mask

As shown in Figure 4.2, the three object masks, O_G, O_C and O_D are combined using a logical "OR" operation in order to obtain a single object mask O_F. As stated earlier, our objective is not to miss out a shadow pixel from any of the three shadow masks, so that none of the object masks would have included in it a shadow pixel. Therefore, in

order to include in each of the shadow masks all the shadow pixels, we propose to use reasonably large values for the threshold parameters ε_0, ε_C and ε_D. We now discuss the following two situations in which despite the choices of large values for ε_C and ε_D, the shadow masks S_C and S_D may still miss some of the shadow pixels, and consequently, O_C and O_D will include these corresponding pixels:

(1) In the construction of the shadow mask based on color, we have used the color invariance property, that is, the color composition of a pixel is not affected by the shadow. It means that the three components of the color are modulated by a same positive constant. However, in some cast shadow areas, such as in penumbra regions, the three color components of a pixel are modulated differently because of different light sources forming the projection of the object. A choice of a small value for ε_C may be sufficient to detect most of the umbra pixels, whereas a relatively large value of ε_C may not be adequate to detect some of the penumbra pixels. Such pixels may, therefore, be detected as object pixels and excluded from S_C, and consequently, get included in the subsequent object mask O_C. Figure 4.3 depicts an example of such a situation. Figure 4.3(a) is the foreground mask of a frame in which there are shadows marked as “Left”, “Middle” and “Right”, resulting from the projections of the object from three different light sources. The left shadow is primarily formed by the dominant light source, whereas the middle and right shadows are formed by the other two. The left shadow can be considered as the one formed under normal illumination condition consisting of an interior

umbra region and the boundary penumbra region. On the other hand, the other two shadows are formed, respectively, from the two distant light source situated on the left of the object and their pixel values are affected from the illumination of all three light sources. The R, G and B components of a pixel in these shadows are modulated differently. Thus, such a pixel does not satisfy the color invariance property. As a result, these two shadows are overwhelmingly of the penumbra type. Figure 4.3(b) is the corresponding object mask resulting from the use of the method of forming shadow mask described in Section 4.2.2. It is seen that the interior pixels in the left shadow (umbra region) are almost completely removed, whereas a large number of the pixels in the penumbra regions of all the shadows are still remaining in the object mask. These pixels could be removed if the value of the threshold parameter ε_c is further increased, but this would be done at the expense of losing a very large number of the object pixels from the object mask.

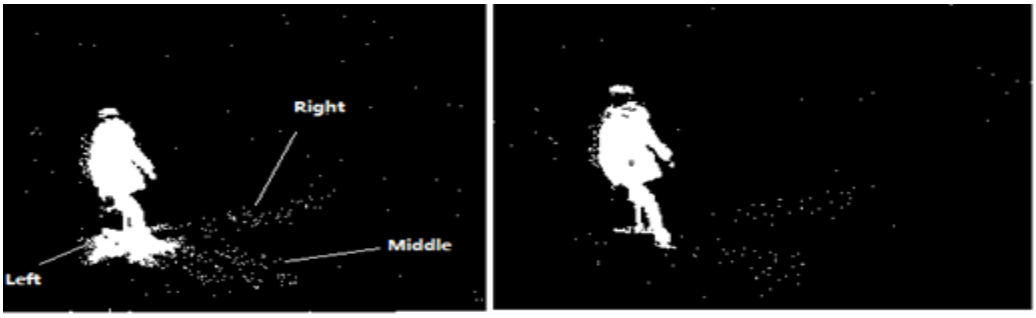


Figure 4.3 (a) Foreground mask of a frame of a video sequence. (b) Object mask resulting from the method described in Section 4.2.2.

(2) The shadow mask S_D has been constructed using the pixel gradients. The construction of this mask is based on the assumption that the variations in gray

levels of the object pixels are larger than that of the shadow pixels. However, in reality, some of the shadow pixels, especially the edge pixels, having relatively large gradients, would be regarded as object pixels and consequently excluded from the shadow mask S_D, included in the subsequent object mask O_D. Figure 4.4 illustrates an example of such a situation. Figure 4.4(a) is an original gray level frame of a sequence and Figure 4.4(b) is the corresponding foreground mask; Figure 4.4(c) shows the object mask resulting from the use of the method of forming a shadow mask described in Section 4.2.3. It is seen that the most shadow pixels have been successfully removed, but there are some shadow pixels with the structure of continuous or discontinuous thin lines still remaining in the object mask. A comparison of Figure 4.4(c) with Figure 4.4(a) shows that these line pixels correspond to the edges within the shadowed background or the boundary edges, where gradient values of the pixels are usually quite large.

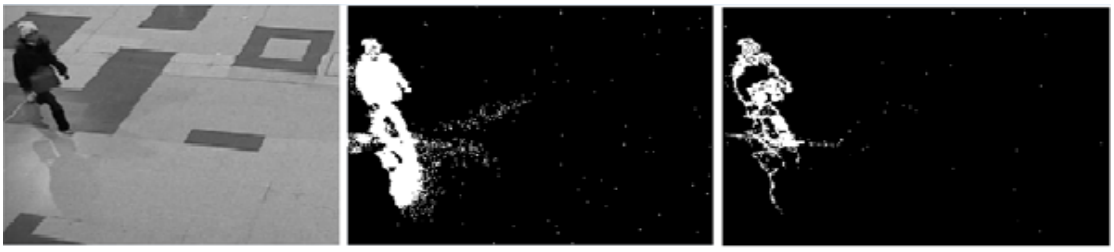


Figure 4.4 (a) Original frame. (b) Corresponding foreground mask. (c) Object mask resulting from the method described in Section 4.2.3.

It is clear that in above two cases, O_C and O_D would be affected in the sense that they will also have some shadow pixels considered as object pixels. Therefore, a direct "OR" operation on the pixels of O_G, O_C and O_D would result in a final object mask O_F

including these shadow pixels. As discussed above, the shadow pixels appearing in O_C and O_D have a structure of thin lines or isolated clusters of some small numbers of pixels. We, therefore, propose to make use of the morphological opening operation [61] in order to remove these artifacts from O_F. To apply the morphological opening operation, the merging operation given in Figure 4.2 is modified as shown in Figure 4.5. Since O_C and O_D could be affected by the inclusion of the two types of shadow pixels, these two masks are first combined into a single object mask O_{CD} using a logical "OR" operation and then subjected to a morphological opening operation with a pre-specified coefficient to generate a subsequent object mask O_{CDM}. This mask is finally combined with O_G using a logical "OR" operation to obtain the final object mask O_F.

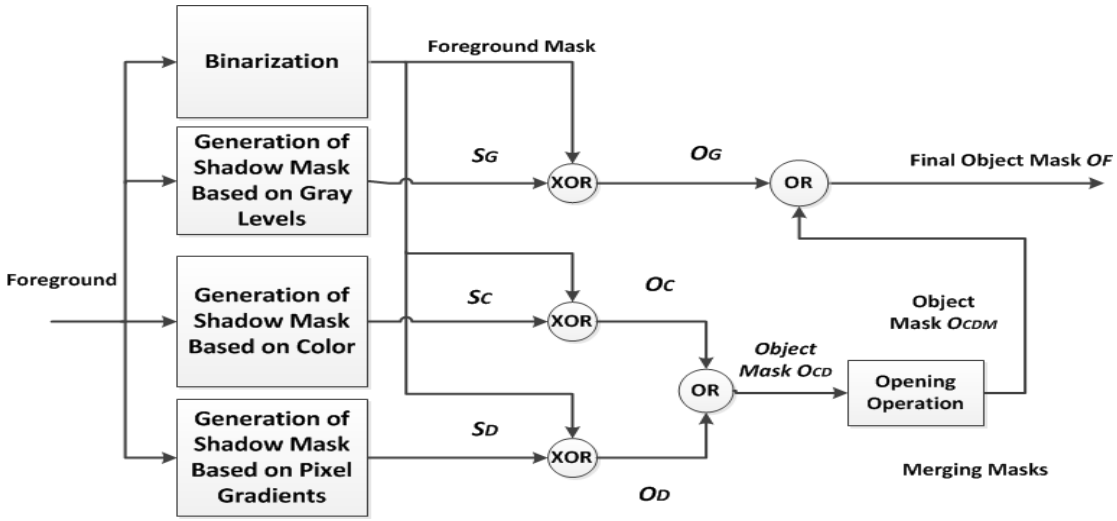


Figure 4.5 Modified scheme of the proposed method.

4.3. Performance Evaluation

The proposed method is applied to remove shadow in a number of video sequences to assess its performance and the results are compared with those obtained by applying the methods in [39] and [41]. Figure 6 illustrates the results of applying the proposed method and those given in [39] and [41] on a video sequence from PETS data sets [62]. The video contains 2370 frames.

Figure 4.6(a) and (b) show original frames 147 and 291 of this sequence and the corresponding foreground masks. Figures 4.6(c), (d) and (e) are the corresponding object masks resulting from the methods of [39] and [41] and that using the proposed method, respectively. It is clear from Figure 4.6(e), that the method of [39] misses many of the object pixels that have gray levels similar to that of the background. Similarly, as seen from Figure 4.6(d), a number of object pixels are missed due to their color or texture being similar to the corresponding background pixels. A comparison of Figures 4.6(c), (d) and (e) shows that, the proposed method is the best among all the three methods considered in capturing object pixels and in removing the shadow ones. The reason for this better performance of the proposed method could be attributed to the fact that it uses three different features in parallel. Therefore, the object pixels missed by one of the features may be recovered by one or both of the other two features.

Figure 4.7 illustrates the results of applying the proposed method and those given in [39] and [41] on two video sequences *Intelligentroom* [63] and *Hall_Monitor* [64].

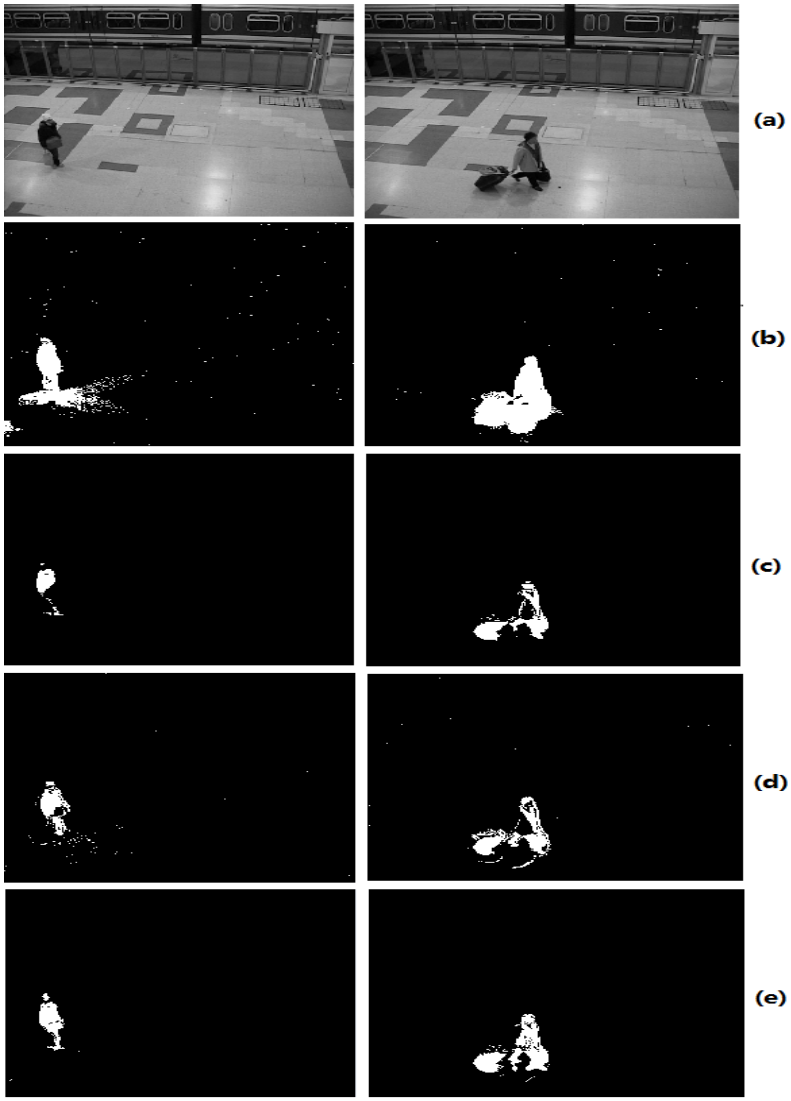


Figure 4.6 (a) Original 147th and 291st frames of a video sequence from the PETS data sets. (b) Foreground masks generated using GMM. (c) Object masks results obtained by applying method [39]. (d) Object masks obtained by applying method [41]. (e) Object masks obtained by applying the proposed method.

Figure 4.7(a) shows the originals of frames 300 and 41, respectively, of the two video sequences. Figure 4.7(b) shows the corresponding foreground masks, and Figures 4.7(c), (d) and (e) are the corresponding object masks resulting from the methods of [39] and [41] and that using the proposed one. It can be clearly seen that, among the three methods, the proposed one gives the most complete object mask.

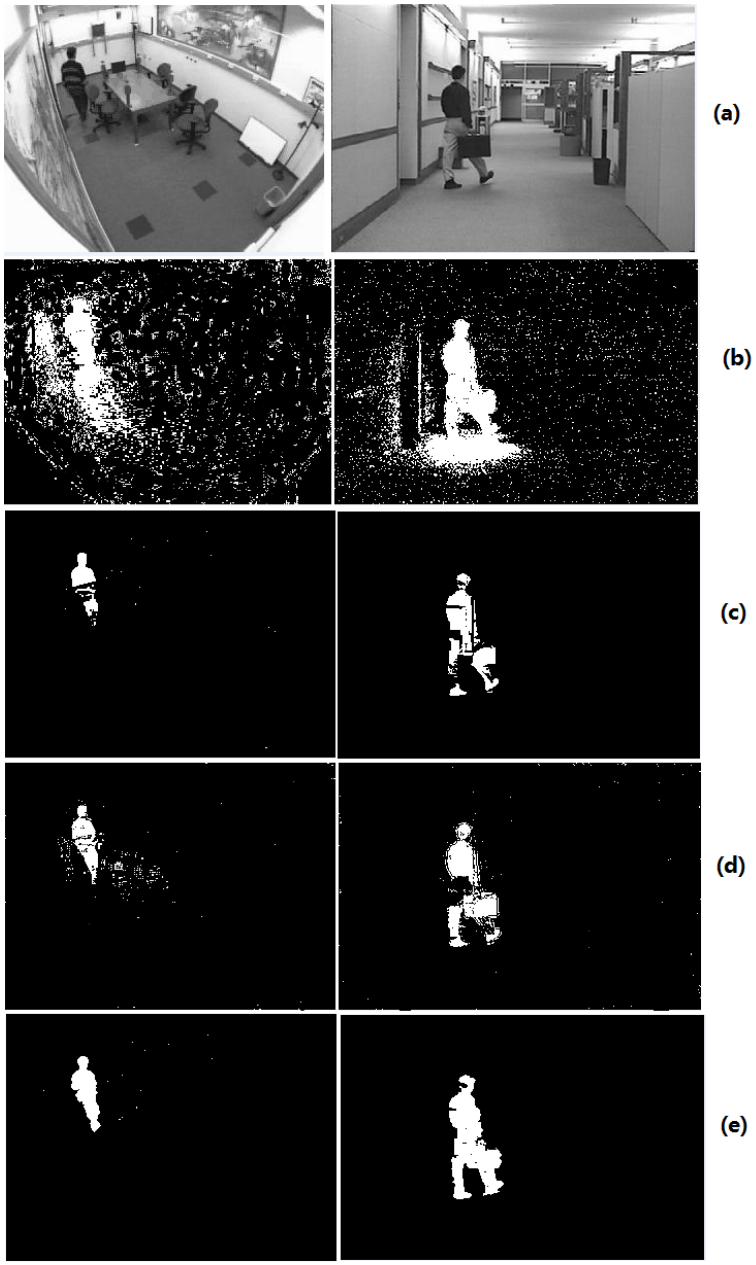


Figure 4.7 (a) Original 300th and 41st frames from sequences Intelligentroom and Hall_Monitor respectively. (b) Foreground masks using GMM. (c) Object masks obtained by applying method [39]. (d) Object masks obtained by applying method [41]. (e) Object masks obtained by applying the proposed method.

In order to provide a quantitative performance evaluation of the proposed and the other two methods considered in this chapter, we use the false alarm rate (FAR) and tracker detection rate (TRDR) [57] defined below as performance measures.

$$FAR = \frac{FP}{TP+FP} \tag{4.10}$$

$$TRDR = \frac{TP}{TP+FN} \tag{4.11}$$

where TP is true positive, FP is false positive, and FN is false negative. Table 4.1 gives these quantitative results of the three methods on the video sequence frames chosen earlier for the illustration of visual performance. From the table, the proposed method applied in frame 291 has a slightly higher FAR than those in [39] and [41], since there are some shadow pixels being detected as object ones mistakenly, as shown in the right image of Figure 4.6(e). However, for the frame 291, the proposed method has the highest TRDR value among the three methods, which means it accurately detects the most object pixels among the three methods. For frame 147, the proposed method gives best performance among the three methods, which means it accurately detect most object pixels and has a smallest ratio of misclassifying the shadow pixels as object ones.

Table 4.1 Results of performance evaluation

	147th frame		291st frame	
	FAR	TRDR	FAR	TRDR
Method [39]	0.289	0.385	0.169	0.625
Method [41]	0.388	0.719	0.137	0.769
Proposed Method	0.196	0.868	0.217	0.876

4.4. Summary

In this chapter, a method for cast shadow removal using the features pixel gray levels, color and gradients has been developed. The objects and shadows of a video sequence frame are first examined from the standpoint of the three features, and then a new method employing three features in parallel for removing the cast shadow has been proposed. The challenges arising from the formation of the cast shadows from multiple illumination sources or from the similarity of texture features between the object and shadow pixels have been discussed and a scheme to overcome these challenges has also been proposed. Subjective and objective results of applying the proposed method to video sequences are given and compared with those of two recently reported methods. These results have demonstrated the effectiveness and superiority of the proposed method for cast shadow removal.

Chapter 5

Conclusion

5.1. Concluding Remarks

Segmentation of moving objects is an essential step in many vision-based applications. Presence of motions and cast shadows in video sequences makes the task of segmentation a difficult problem. The existing segmentation techniques do not perform well in the presence of a rapidly changing background or require *a priori* knowledge of the object's shape and size and a manual selection of training samples to be used by a classifier. This research has been concerned with the development of cost-effective techniques for segmentation of foreground and removal of cast shadows from video sequences with a dynamic background.

In the first part of the thesis, the problem of segmenting a moving foreground from video sequences with a dynamic background has been investigated by treating it as that of classifying the foreground and background pixels of a frame. For the purpose of this classification, a novel feature image has been constructed and used in the framework of a support vector machine. The feature image has been constructed by combining the individual features representing the gray levels, hue and saturation levels of the pixels with suitable weights. An attribute of the feature image leading to the computational simplicity of the proposed segmentation technique is in its ability to represent multiple

features of a pixel with a scalar value. Another distinguishing characteristic of the proposed method is that, unlike some other data classification based approaches for segmentation in which *a priori* knowledge of the object's shape and size is required or a set of training samples needs to be manually selected, the training samples to be employed by the classifier are automatically selected in this method. The use of a scalar-valued feature image incorporating multiple features and the employment of automatically selected training samples in the framework of a support vector machine are shown provide a computationally simple yet accurate method for segmenting the foreground from video sequences with a dynamic background.

Cast shadows are integral parts of moving objects, since pixels belonging to the two regions have identical motions. An algorithm designed for object segmentation very often results in segmenting the entire foreground. Therefore, in order to segment the foreground, the shadow pixels need to be removed from the foreground. Removal of shadow pixels becomes a difficult task in a situation in which a feature of these pixels is similar to that of the object pixels. In the second part of the thesis, a simple yet efficient method has been developed for removing the cast shadow from a foreground using multiple features. The proposed method is based on the premise that an object pixel is less likely to be similar to a shadow pixel simultaneously with respect to all the features used for classifying the two types of pixels. The proposed method does not have the complexity of a shadow removal technique based on using a vector-valued feature.

Object masks, each constructed based on an individual feature, are combined to obtain an accurate overall object mask.

Extensive simulations have been carried by applying the proposed and other techniques of foreground segmentation and shadow removal to video sequences with a dynamic background. The results of the experiments have demonstrated the effectiveness of the proposed methods and their superiority to the existing techniques.

5.2. Scope for Future Work

The research work undertaken in this thesis has been concerned with developing effective techniques for segmentation of foreground in video sequences with a dynamic background and removal of cast shadows of objects in the foreground. Even though the performance of techniques have been shown to be superior to that provided by some of the other techniques in the literature, the ideas and schemes proposed in this thesis can be further investigated from the point view of further increasing their segmentation accuracy and reducing the computational complexity.

The data classification method introduced in this thesis for foreground segmentation has been applied to the segmented foreground obtained by using the GMM method. Any of the object pixels not detected by the GMM method would obviously be absent from the final foreground segmentation. A further study needs to be carried out to resolve this problem.

In the proposed method for segmenting background, a color frame with no moving objects in it is needed for its use as a universal reference frame for the construction of feature images corresponding to the hue and saturation level features of the pixels. A study could be carried out to obtain a reference color background image that is adaptive to the frame under consideration for the foreground segmentation.

In the proposed method for foreground segmentation, the window size has been empirically fixed. A study could be undertaken to devise a technique to make the widow size adaptive to the shape and size of the moving foreground.

References

[1] A. Cavallaro, O. Steiger, and T. Ebrahimi, “Tracking Video Objects in Cluttered Background,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 4, pp. 575–584, 2005.

[2] F.H. Cheng and Y.L. Chen, “Real time multiple objects tracking and identification based on discrete wavelet transform,” Pattern Recognition, vol. 39, no. 6, pp. 1126–1139, 2006.

[3] R. Li, S. Yu, and X. Yang, “Efficient spatio-temporal segmentation for extracting moving objects in video sequences,” IEEE Transactions on Consumer Electronics, vol. 53, no. 3, pp. 1161–1167, Aug. 2007.

[4] A. Colombari, A. Fusiello, and V. Murino,“Segmentation and tracking of multiple video objects,” Pattern Recognition, vol. 40, no. 4, pp. 1307–1317, 2007.

[5] Y.L. Tian and A.Hampapur,“Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance,” Seventh IEEE Workshops on Application of Computer Vision, vol. 2, pp.30-35, Jan. 2005.

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

[7] M. Piccardi, “Background subtraction techniques: a review,” IEEE International Conference on Systems, Man and Cybernetics, 2004: 3099-3104.

[8] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp.780-785, 1997.

[9] C. Stauffer and W.E.L. Grimson,“Adaptive Background Mixture Modules for Real-time Tracking,” Computer Vision and Pattern Recognition, vol.2, pp. 246-252, 1999.

[10] S. Fazli, H. M. Pour and H. Bouzari,“A Novel GMM-Based Motion Segmentation Method for Complex Background,” Fifth IEEE GCC Conference & Exhibition, pp.1-5, March 2009.

[11] T.K. Kim, J.H. Im and J.K. Paik, “Video object segmentation and its salient motion detection using adaptive background generation,” IET, Electronics Letters, vol.45, issue 11, pp. 543-543, May 2009.

[12] H. Greenspan, J. Goldberger and A.Mayer,“Probabilistic Space-Time Video Modeling via Piecewise GMM,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no.3, pp.384-396, March 2004.

[13] D. Zhou and H. Zhang, ” Modified GMM background modeling and optical flow for detection of moving objects,” 2005 IEEE International Conference on Systems, Man and Cybernetics, vol.3, pp. 2224-2229, Oct. 2005.

[14] H. Zheng, Z. Liu and X. Wang, “Research on the Video Segmentation Method with Integrated Multi-features Based on GMM,” 2008 International Conference on Computational Intelligence for Modeling Control & Automation, pp. 260-264, Dec. 2008.

[15] B. Li, Z. Tang, B. Yuan and Z. Miao, “Segmentation of Moving Foreground Objects using Codebook and Local Binary Patterns,” 2008 Congress on Image and Signal Processing, vol.4, pp. 239-243, May 2008.

[16] D. Zhou, H. Zhang and N.Ray, “Texture based background subtraction,” 2008 International Conference on Information and Automation, pp. 601-605, June 2008.

[17] B. Zhang, Y. Gao, B. Zhong, “Complex background modeling and motion detection based on Texture Pattern Flow,” 19th International Conference on Pattern Recognition,pp. 1-4, Dec. 2008.

[18] M. Heikkilä and M. Pietikäinen,“A texture-based method for modeling the background and detecting moving objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 657-662, Apr. 2006.

[19] L. Li, W. Huang, Y. Gu and Q. Tian,"Statistical Modeling of Complex Backgrounds for Foreground Object Detection," IEEE Transactions on Image Processing, vol. 13, no.11, pp. 1459-1472, Nov. 2004.

[20] V. Mahadevan and N. Vasconcelos, "Spatiotemporal Saliency in Dynamic Scenes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no.1, pp. 171-177, Jan. 2010.

[21] M. M. Azab, H.A. Shedeed and A.S. Hussein,"A new technique for background modeling and subtraction for motion detection in real-time videos," 2010 17th IEEE International Conference on Image Processing, pp. 2453-2456, Sept.2010.

[22] L. Cheng and M. Gong, D. Schuurmans and T. Caelli,"Real-Time Discriminative Background Subtraction," IEEE Transactions on Image Processing, vol. 20, no. 5, pp. 1401-1414, May 2011.

[23] Y. Wang, K.-F. Loe, T. Tan and J.-K. Wu, "Spatiotemporal video segmentation based on graphical models," IEEE Transactions on Image Processing, vol. 14, no. 7, pp. 937-947, July 2005.

[24] W. Kim, C. Jung and C. Kim,"Spatiotemporal Saliency Detection and Its Applications in Static and Dynamic Scenes," IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.4, pp. 446-256, April 2011.

[25] Y.-P. Guan,"Spatio-temporal motion-based foreground segmentation and shadow suppression," IET, Computer Vision, vol. 4, no.1, pp. 50-60, March 2010.

[26] R. Pless,"Spatio-temporal Background Models for Outdoor Surveillance," 2005 Journal on Applied Signal Processing, vol. 14, pp. 2281-2291, 2005.

[27] S. D. Babacan and T.N. Pappas, "Spatiotemporal Algorithm for Background Subtraction," 2007 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 1065-1068, April 2007.

[28] S.-C. Huang, “An Advanced Motion Detection Algorithm with Video Quality Analysis for Video Surveillance Systems,” IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.1, pp. 1-14, 2011.

[29] Y. Sheikh and M. Shah,“Bayesian Modeling of Dynamic Scenes for Object Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no.11, pp. 1778-1792, Nov. 2005.

[30] W. Wang, J. Yang and W. Gao,“Modeling Background and Segmenting Moving Objects from Compressed Video,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no.5, pp. 670-681, May 2008.

[31] H. Yang, J. Tian, Y. Chu, Q, Tang and J. Liu, “Spatiotemporal Smooth Models for Moving Object Detection,” IEEE Signal Processing Letters, vol. 15, pp. 497-500, 2008.

[32] B. Zhang, Y. Gao, S. Zhao and B. Zhong, “Kernel Similarity Modeling of Texture Pattern Flow for Motion Detection in Complex Background,” IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.1, pp. 29-38, January 2011.

[33] Corinna, Cortes and V. Vapnik, "Support-vector networks," Machine Learning, Springer Netherlands, vol.20, no.3, pp.273–297, 1995.

[34] H. Li and J. Cao, “Detection and Segmentation of Moving Objects Based on Support Vector Machine,” 2010 Third International Symposium on Information Processing, pp. 193-917, Oct. 2010.

[35] J. Zhang and C. H. Chen, “Moving Objects Detection and Segmentation In Dynamic Video Backgrounds,” 2007 IEEE Conference on Technologies for Homeland Security, pp. 64-69, 2007.

[36] T. Li, X. Cao and Y. Xu, “An Effective Crossing Cyclist Detection on a Moving Vehicle,” 2010 8th World Congress on Intelligent Control and Automation, pp. 368-372, July 2010.

[37] X. Cao, C. Wu, J. Lan, P. Yan X. Li, “Vehicle Detection and Motion Analysis in Low-Altitude Airborne Video Under Urban Environment,” IEEE Transactions on Circuits and Systems for Video Technology, vol.21, no.10, pp. 1522-1533, Oct. 2011.

[38] J. Zhou, D. Gao and D. Zhang, “Moving Vehicle Detection for Automatic Traffic Monitoring,” IEEE Transactions on Vehicular Technology, vol. 56, pp. 51-59, 2007.

[39] C.-T. Chen, C.-Y. Su and W.-C. Kao,“An Enhanced Segmentation on Vision-based Shadow Removal for Vehicle Detection,” 2010 International Conference on Green Circuits and Systems, pp. 679 - 682, 2010.

[40] E. Salvador, A. Cavallaro and T. Ebrahimi,“Cast shadow segmentation using invariant color features,” Computer Vision and Image Understanding, vol. 95, no.2, pp. 238-259, 2004.

[41] C. Wang and W. Zhang, “A Robust Algorithm for Shadow Removal of Foreground Detection In Video Surveillance,” 2009 Asia-Pacific Conference on Information Processing, vol.2, pp.422-425, 2009.

[42] M.-T. Yang, K.-H. Lo, C.-C Chiang and W.-K Tai, “Moving Cast Shadow Detection by Exploiting Multiple Cues,” Image Processing, IET, vol. 2, pp. 95 -104, 2008.

[43] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf, “Support vector machines,” IEEE Intelligent Systems, vol. 13, no. 4, pp. 18–28, July-Aug. 1998.

[44] M. Pontil and A. Verri, “Support vector machines for 3D object recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 6, pp. 637–646, June 1998.

[45] D.J. Sebald and J.A. Bucklew, “Support vector machine techniques for nonlinear equalization,” IEEE Transactions on Signal Processing, vol. 48, no. 11, pp. 3217–3226, Nov. 2000.

[46] C.W. Hsu, and C,J. Lin, “A comparison of methods for multiclass support vector machines,” IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[47] V.N. Vapnik, “An overview of statistical learning theory,” IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 988–999, Sept. 1999.

[48] C. D.Manning, P. Raghavan and H. Schütze,“Introduction to Information Retrieval,” Cambridge University Press. Chap. 15, pp.319-348. 2008.

[49] F. Porikli, “Integral histogram: A fast way to extract histograms in Cartesian spaces,” in Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition, vol. 1. Jun. 2005, pp. 829–836.

[50] http://en.wikipedia.org/wiki/RGB_color_model

[51] Pearson, K. (1895). "Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 186: 343–326.

[52] N. Cristianini,“Training Invariant Support Vector Machines,” Machine Learning, 46, 161–190, 2002 Kluwer Academic Publishers.

[53] C. Tang, M. O. Ahmad and C. Wang, “Foreground Segmentation in Video Sequences with a Dynamic Background Using Multiple Features,” to be submitted for publication to IEEE Transactions on Circuits and Systems for Video Technology.

[54] http://www.cs.bu.edu/groups/ivc/data/DynamicBackgrounds/ICCV2003/water/

[55] http://perception.i2r.a-star.edu.sg/BK_Model_TestData/WaterSurface.m1v

[56] http://perception.i2r.a-star.edu.sg/BK_Model_TestData/Curtain.m1v

[57] J.Black, T. Ellis and P. Rosin, “A Novel Method for Video Tracking Performance Evaluation,” International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 125-132, 2003.

[58] http://www.cs.cmu.edu/~yaser/new_backgroundsubtraction.htm

[59] http://perception.i2r.a-star.edu.sg/BK_Model_TestData/Campus.m1v

[60] C. Tang, M. O. Ahmad and C. Wang, "An Efficient Method of Cast Shadow Removal Using Multiple Features," submitted to a special issue of Springer Journal, "Image and Video Processing for Security".

[61] D. Vernon, Machine Vision, Prentice-Hall, 1991, pp 78-79.

[62] http://www.cvg.rdg.ac.uk/PETS2006/data.html

[63] http://www.openvisor.org/video_details.asp?idvideo=114

[64] http://trace.eas.asu.edu/yuv/

[65] E. Arbel and H.Hel-Or, "Shadow Removal Using Intensity Surfaces and Texture Anchor Points," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no.6 , pp. 1202-1216, 2011.

[66] W.T. Wintringham, Bell Telephone Laboratories: Colorimetry and color television [M]. Inc., Murray Hill, N.J. Current version: 08 January, 2007.

[67] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," Artificial Intelligence Lab. Mass. Inst. Technol., Cambridge, MA, Tech. Rep. 1602, 1997.

[68] A. Barla, F. Odone, and A. Verri, "Histogram Intersection Kernel for Image Classification," in Proc. IEEE Int. Conf. Image process., vol. 3, pp. 513–516, 2003.

[69] D.G. Altman, J.M. Bland, "Diagnostic tests. 1: Sensitivity and specificity," BMJ 308 (6943): 1552. PMC 2540489. PMID 8019315, 1994.