2009

# Adaptive foreground segmentation using fuzzy approach

Huajing Yao
*University of Windsor*

# ADAPTIVE FOREGROUND SEGMENTATION USING FUZZY APPROACH

by
**Huajing Yao**

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2009

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

*Your file Votre référence*
*ISBN:* 978-0-494-57639-7
*Our file Notre référence*
*ISBN:* 978-0-494-57639-7

# Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that if I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

# Abstract

Intelligent visual surveillance which attempts to detect, recognize and track certain objects from image sequences is becoming an active research topic in computer vision community. Background modeling and foreground segmentation are the first two and the most important steps in any intelligent visual surveillance systems. The accuracy of these two steps highly effects performance of the following steps. In this thesis, we propose a simple and novel method which employs histogram based median method for background modeling and a fuzzy k-Means clustering approach for foreground segmentation. Experiments on a set of videos and benchmark image sequences show the effectiveness of the proposed method. Compared with other two contemporary methods - *k*-Means clustering and Mixture of Gaussians (MoG) - the proposed method is not only time efficient but also provides better segmentation results.

# Dedication

To my parents and elder sister

## Acknowledgements

I would like to take this opportunity to express my sincere gratitude to Dr. Imran Ahmad, my supervisor, for his steady encouragement, patient guidance and enlightening discussions throughout my graduate studies. Without his help, the work presented here could not have been possible.

I also wish to express my appreciation to Dr. Boubakeur Boufama, School of Computer Science and Dr. Mehdi Monfared, Department of Mathematics and Statistics for being in the committee and spending their valuable time and Dr. Ziad Kobti, School of Computer Science for serving as the chair of the defense.

Finally, I would like to thank all my friends on and off campus in Windsor; I could not recover that fast from the accident without your help.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

In recent years, intelligent visual surveillance, also called automated visual surveillance has emerged as an important research topic in computer vision community and with good reason. According to New Scientist magazine: "*If the technology takes off it could put an end to a longstanding problem that has dogged CCTV almost from the beginning. It is simple: there are too many cameras and too few pairs of eyes to keep track of them. With more than a million CCTV cameras in the UK alone, they are becoming increasingly difficult to manage.*"[1]

Surveillance cameras are cheap and ubiquitous. In conventional approach of visual surveillance systems, Closed-Circuit Television (CCTV) cameras are installed in public and private areas to capture indoor or outdoor scene information, which is then transmitted and displayed in a terminal or a television where a human observer keeps monitoring and optionally records video information in a storage medium [61].

However, the manpower required for surveillance is not only expensive but may also involves a host of associated problems. From the real-time threat detection perspective, it

[1] New Scientist, 12 July 2003, page 4.

is a well known fact that over a period of time, the human visual attention drops below an acceptable level even when trained personnel are assigned to the surveillance task. Further, it is difficult for one operator to concurrently monitor several places. Currently, most of the video information from the cameras are used merely as an archive to refer back to once an incident is known to have taken place [52].

The purpose of intelligent visual surveillance system is to obtain a description of what is happening in a monitored area and then to take appropriate action based on that interpretation, e.g., alert a human supervisor to reduce human involvement significantly and assist human operators for better monitoring [24], [2].

The essential idea of developing an intelligent visual surveillance system is to detect motion and track objects of interest that generates motion, upon which higher level recognition capabilities can be achieved [42], [24], [25], [27].

A typical method to detect unusual motion is through *background subtraction* - a technique which detects moving regions in an image by taking the difference between the current frame and the background model in a pixel-by-pixel fashion [24], [25]. Background subtraction has two important components: (1) background modeling and (2) foreground segmentation. Background modeling is a representation of the empty scene (without foreground objects), while foreground segmentation is the detection of moving foreground objects within the monitored scene.

In an ideal situation, the background model is known in advance and the background does not change over time. However, in realistic environments, background subtraction methods encounter several problems. Following is a brief description of some of the important ones [29], [35], [43]:

- the monitored area contains moving foreground objects and it is difficult to get an empty background model, i.e., public areas such as airport terminals, shopping malls, etc.

- the appearance of objects in background model changes due to illumination changes in the scene, e.g., changing weather conditions, time of the day, or the switching of lights can lead to entire scene being misclassified as foreground object(s).
- there may be dynamic background objects in the surveillance scene, i.e., flickering computer screen, wavering tree branches, etc.
- some or all of the static background part may convert to a dynamic background, e.g., by turning on a computer screen. A dynamic background pixel can also turn to a static one such as a wavering tree branch when the wind stops.
- the background object turns to the foreground object, e.g., in a parking lot, a car may enter the scene and is parked. Initially, the car needs to be detected as a foreground object but after it has been parked there, should be considered as part of the background, and vice versa.

Background model for general scenes should be able to represent not only the stationary objects, i.e., walls, doors and room furniture, but also the non-stationary (or "dynamic") objects such as waving tree branches, moving escalators. Moreover, background model for general environments should self-evolve to incorporate illumination changes in the scene and changes of background objects (i.e., add, remove and relocate the background objects).

Once background is computed or known, foreground segmentation and detection of moving foreground objects can be done by subtracting and thresholding each incoming video frame from the background model [15]. Some of the difficulties in foreground segmentation [2], [27] are as follows:

- Finding a threshold to differentiate between background and foreground is difficult. If the threshold is too low, some background pixels may be misclassified as foreground pixels. If the threshold is too high then some foreground pixels may be misclassified as background pixels. In the latter case, the foreground region might consist of apertures inside and/ or get segmented into several isolated regions. The inaccurate segmentation results will create problems for later tasks, i.e., tracking the foreground objects and will result in poor performance of the overall surveillance system.

- Shadow introduces several difficulties for the process of detecting foreground, i.e., object shape distortion, object merging or even object losses due to shadow over another object. Shadows are categorized into cast shadows and self shadows. The cast shadow is caused by moving objects on the side of the object while self shadow is caused by the object on itself, i.e., shadows cast by folding of a shirt. During the process of cast shadow removal, some of the self shadow regions may also be deleted which may not be desirable.

- Another major problem in foreground segmentation is the camouflage. When a foreground object has a similar color and intensity as the background, it is difficult to distinguish between them. Part of the foreground object is misclassified as the background, and that may lead to missing or split foreground object regions.

Designing an automated surveillance system has a number of issues and difficulties and some of those have been presented above. The first stage of an automated surveillance system is to build a background model from which the foreground object(s) can be properly separated. Therefore, there is a need to improve background modeling and foreground segmentation processes to improve the overall surveillance task.

In this thesis, we use a fast and effective background modeling scheme that is based on a histogram based median method. It estimates the background model from the image sequence itself without prior knowledge of the scene, i.e., even in the presence of foreground objects in the image sequence [28], and proposes fuzzy k-Means clustering for foreground segmentation, which eliminates the dilemma of choosing a proper threshold.

Rest of the thesis is organized as follows: Chapter 2 is a detailed introduction of the entire intelligent visual surveillance system and its functional components. Chapter 3 provides a brief survey of the related work with an emphasis on two techniques directly related to this thesis: k-Means clustering and Mixture of Gaussians. Chapter 4 provides details of the proposed method. Experimental results and comparisons are presented in Chapter 5 and finally, Chapter 6 provides some concluding remarks.

# Chapter 2

# Intelligent Visual Surveillance

This chapter gives an overview of intelligent visual surveillance. The general framework and the functional components are explained, followed by some applications of the intelligent visual surveillance. Finally, the conventional approaches for motion detection which is the first important step in intelligent visual surveillance are discussed.

## 2.1 Overview of the Intelligent Visual Surveillance

Intelligent visual surveillance attempts to detect, recognize and track certain objects from image sequences. Its general framework includes following stages:

- modeling of environments
- detection of motion
- classification of moving objects
- tracking

- understanding and description of behaviors
- human identification
- fusion of data from multiple cameras

Figure 2.1 is the illustration of the intelligent visual surveillance system [24].

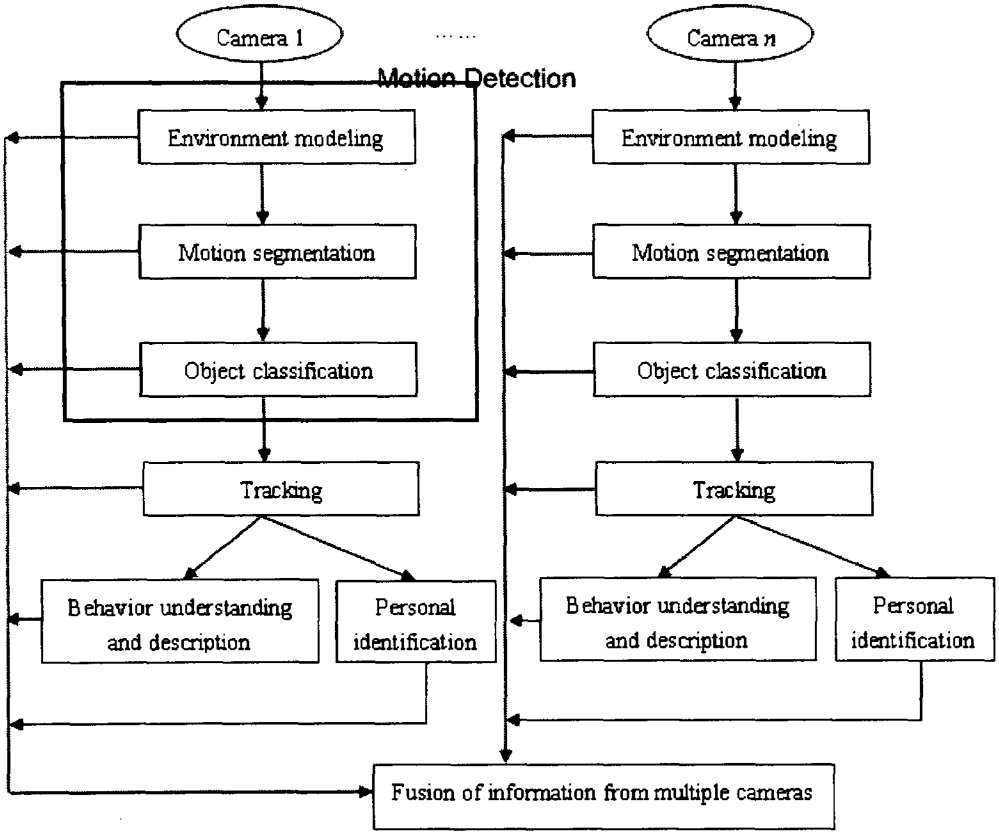


Figure 2.1: General framework of intelligent visual surveillance

Motion detection is the first and the most important step in nearly every visual surveillance system. It aims at segmenting regions corresponding to moving objects from the rest of an image. Subsequent processes such as tracking and behavior recognition are greatly dependent on it. Environment modeling, motion segmentation, and object classification are typical processes in motion detection [24], [57]. Environment modeling is same as the background modeling, and aims to establish a background model for the surveillance scene. Motion Segmentation is carried out to detect regions corresponding to moving objects such as vehicles and humans. Detection of moving regions provides a focus of attention for later processes since only these regions need be considered. Object classification is the determination of the type of object corresponding to different moving regions, i.e., in road traffic scenes, may include humans, vehicles, flying birds and moving clouds.

After motion detection, surveillance systems generally track moving objects from one frame to another in an image sequence. The difficulty of tracking is determining the appearance and location of a particular object in the sequence of frames, i.e., two people walk towards each other, after occlusion, the tracking algorithms have to figure out "who is who?" among all the frames [21]. Understanding of behaviors is the recognition of suspicious motion patterns: the behaviors in the testing sequence are analyzed and compared with typical labeled behaviors. In some applications, description of behaviors using natural language is necessary for a nonspecialist operator involved in visual surveillance [41].

Personal identification deals with the issue "who is now entering the area under surveillance?". It can be treated as a special behavior understanding problem. In such applications, biometric features such as human face and gait are widely used [55]. Fusion of information from multiple cameras can benefit the surveillance task because the surveillance area is expanded and multiple view information can overcome occlusion. The problems in this step range over camera installation (cover the entire scene with the minimum number of cameras), object matching (find the correspondences between the objects in different image sequence taken by different cameras) and data fusion, i.e., to "synthesize the tracking results of different cameras to obtain an integrated trajectory" [24].

From the functional blocks point of view, intelligent visual surveillance system consists of three levels: pixel processing level, object segmentation level and tracking level as shown in Figure 2.2.

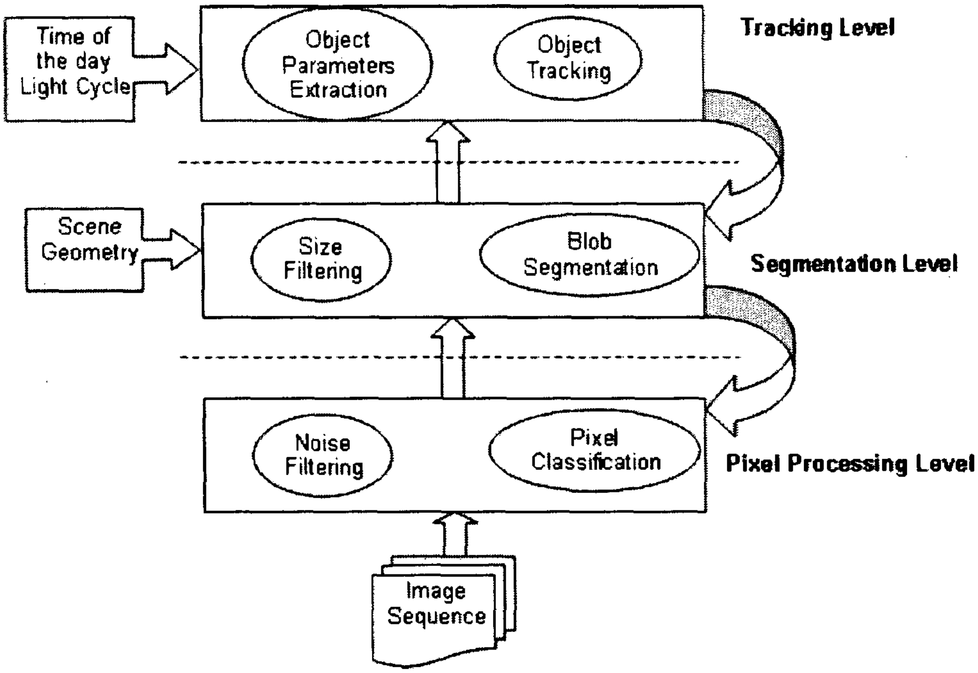


Figure 2.2: Building blocks of intelligent visual surveillance system

Pixel processing level is a basic step for any intelligent visual surveillance system. The classification of foreground and background is based on the pixel itself, i.e., intensity, color. Object segmentation level establishes clear foreground object's blob, which is a spatially coherent group of pixels clustered together. Usually 4-connectivity or 8-connectivity analysis is performed to identify blobs [2]. 4-connected pixels are neighbors to every pixel that touches one of their edges. These pixels are connected horizontally and vertically. In terms of pixel coordinates, every pixel that has the coordinates $(x \pm 1, y)$ or $(x, y \pm 1)$ is connected to the pixel at $(x, y)$. 8-connected pixels are neighbors to every pixel that touches one of their edges or corners. These pixels are connected horizontally, vertically, and diagonally. In addition to 4-connected pixels, every pixel with coordinates $(x \pm 1, y \pm 1)$ or $(x \pm 1, y \mp 1)$ is connected to the pixel at $(x, y)$.

## 2.2 Applications of Intelligent Visual Surveillance

Intelligent visual surveillance has a wide range of applications, such as a security for communities and important buildings, traffic surveillance in cities and expressways, detection of military targets, etc [19]. However, the most investigated application is the surveillance of people or vehicles. Some of the important applications are described as follows:

### 1. Crowd flux statistics and congestion analysis

Using techniques for human detection, visual surveillance systems can automatically compute the flux of people at important public areas such as shopping malls and travel sites, and then provide congestion analysis to assist in the management of people [11]. In a similar way, visual surveillance systems can monitor expressways and junctions in road networks, and further analyze the traffic flow and the status of road congestion, which are of great importance for traffic management [24].

### 2. Identity checking (biometrics)

Video surveillance can be used for access control in some security-sensitive locations such as military bases and important organizations. Identity of a person can be verified by different biometric features such as height, facial appearance and walking gait in real time, and then to decide whether of the visitor can be cleared for entry or not [7].

**3. Suspect identification**

Using intelligent visual surveillance, criminals or suspects can be identified in public or suspected places, e.g., subway stations, casinos by analyzing biometric features of that suspect. If the person detected is a suspect, alarms are given immediately. Such systems along with face recognition capability have already been used at public sites, but the reliability is too low for police requirements [24].

**4. Interactive surveillance using multiple cameras**

For social society, cooperative surveillance using multiple cameras could be used to ensure the security of an entire community, for example by tracking suspects over a wide area by using the cooperation of multiple cameras. For traffic management, interactive surveillance using multiple cameras can help the traffic police discover, track, and catch vehicles involved in traffic offences [24].

**5. Anomaly detection and alarming**

In some circumstances, it is necessary to analyze the behaviors of people and vehicles and determine whether these behaviors are normal or abnormal. For example, visual surveillance systems set up in supermarkets and parking lots can analyze abnormal behaviors indicative of a theft and robbery [14].

## 2.3 Approaches for Motion Detection

Robust tracking of objects and the processes after tracking call for a reliable and effective moving object detection method. Such kind of methods can be characterized by some important features: high precision with two meanings of accuracy in shape detection and reactivity to changes in time, flexibility in different scenarios (indoor, outdoor) or different light conditions, and efficiency, in order for detection to be provided in real-time [19]. There are three conventional approaches in literature for motion segmentation or foreground segmentation [24], [2].

### 2.3.1 Background subtraction

Background subtraction is a popular method for motion segmentation. It is especially useful in situations with a relatively static background. It can detect moving regions in an image by taking the difference between the current image and a reference background image in a pixel-by-pixel fashion. It is a simple and computationally affordable method for real-time applications, but is extremely sensitive to changes in dynamic scenes derived from lighting and extraneous events etc. Therefore, it is highly dependent on a good background model to reduce the influence of these changes [18], [22], [45] in the background model.

The background subtraction method can be written mathematically as:

$$FG_{(x,y)} = \begin{cases} 1, & \text{if } |I_{(x,y)} - BG_{(x,y)}| > T; \\ 0, & \text{otherwise.} \end{cases}$$

where $I_{(x,y)}$ is the color of a pixel at a 2D location $(x,y)$ and $BG_{(x,y)}$ is the color of the same pixel in the background model. If the difference between these two values exceeds some threshold $T$, then this pixel $(x,y)$ in the current frame is classified as the foreground, otherwise it is treated as the background.

### 2.3.2 Temporal differencing

Temporal differencing, also known as the *Frame Differencing* makes use of pixel-wise differences between two or three consecutive frames in an image sequence to extract moving regions [3]. Temporal differencing is very adaptive to dynamic environments, but generally does a poor job of extracting all of the relevant pixels, e.g., there may be holes left inside moving entities.

Mathematically, temporal differencing can be written as:

$$FG_{(x,y)} = \begin{cases} 1, & \text{if } |I^t_{(x,y)} - I^{t-1}_{(x,y)}| > T; \\ 0, & \text{otherwise.} \end{cases}$$

where $I^t_{(x,y)}$ is the intensity of a pixel at the location $(x,y)$ in the current frame captured at time $t$ and $I^{t-1}_{(x,y)}$ is the intensity of the same pixel in a frame captured at time $t-1$. If the difference between these two exceeds the threshold $T$, the pixel $(x,y)$ in current frame is considered as the foreground, otherwise, it is considered as the background. Sometimes more than two consecutive frames are considered to increase the accuracy of detection.

### 2.3.3 Optical flow

Optical flow based motion segmentation methods use characteristics of flow vectors of moving objects over time to detect moving regions in an image sequence. They can be used to detect independently moving objects even in the presence of camera motion. However, most flow computation methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real time without specialized hardware. An illustration of optical flow is shown in Figure 2.3. More detailed discussion of optical flow can be found in [4].

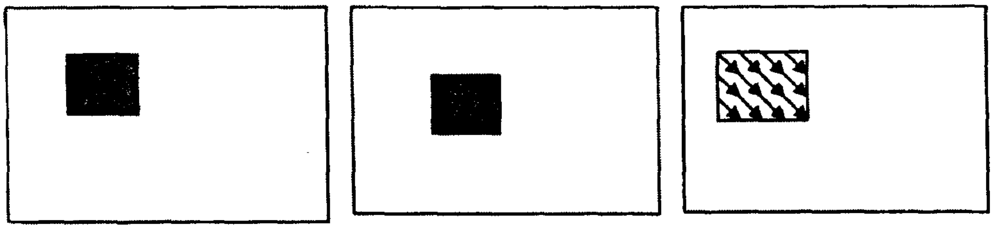


Figure 2.3: Optical flow

## 2.4 Summary

This chapter presents a detailed view of intelligent visual surveillance in terms of its framework and functional components. As one can observe, background modeling and foreground segmentation are at the heart of the intelligent visual surveillance.

# Chapter 3

# Related Work

For background modeling without specific domain knowledge, the background is usually represented by image features at each pixel. The features extracted from an image sequence can be classified into three types [35], [38]: *spectral*, *spatial*, and *temporal* features. *Spectral* features could be gray-scale or color information. *Spatial* features could be gradient or local structure, and *temporal* features could be inter-frame changes at the pixel. Most of the existing methods use *spectral* information, i.e., distribution of intensities or colors at each pixel for background modeling since it is straightforward and suitable for static background pixels [15], [60], [22], [54]. To tolerate the illumination changes, some *spatial* features are incorporated with *spectral* information [37], [29]. *Temporal* features are investigated in [58], [36], [35] to describe the dynamic background pixels such as waving trees, flicking screens. In this thesis, we use color information represented in HSV color space for background modeling and foreground segmentation.

## 3.1 Color Space

The images extracted from video are represented in RGB color space (R: Red; G: Green; B: Blue). In this color space, *spectral* information is equally spread among all of the three components, and it is not perceptually similar to our eyes. Any change in illumination results in changes in values of all of the three components. Therefore, most of the background subtraction approaches convert RGB color space to some other color space such as HSV color space before further processing. HSV stands for Hue, Saturation and Value. It is a widely described color scheme in literature for surveillance [44], [1], [27], [2]. It is a nonlinear transformation of RGB color space.

Figure 3.1 is a pictorial representation of HSV color space. It provides a color decomposition close to human perception where wavelength, brightness and intensity information are separated [34]. The $H$ (Hue) determines the basic color. A hue is referenced as an angle on a color wheel with a range of $[0, 360°]$; the $S$ (Saturation) determines the grey level of the color (or the amount of white light in the color) in the range $[0, 1]$; the $V$ (Value) represents the global intensity of the light with a range of $[0, 1]$.

The advantage of using this color space is that the intensity (V) can be separated from the chromatic information (HS). All of the useful information is present only in the V channel. HSV color space can deal with noise and shadow in the image region very effectively.

The mathematical relationship between RGB and HSV is given as follows [32]:

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360° - \theta & \text{if } B > G \end{cases}$$

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)]$$

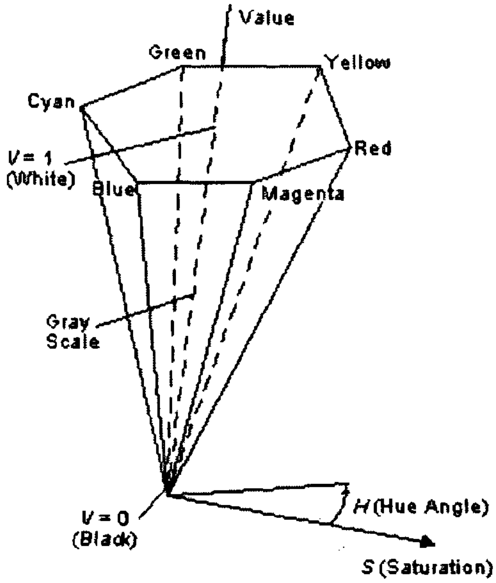


Figure 3.1: Representation of HSV color space in a 3 dimensional space

$$V = \frac{1}{3}(R+G+B)$$

in which

$$\theta = \cos^{-1}\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}$$

Normalized color space is another popular color space [56], [45], [15] used in intelligent visual surveillance. It is independent of the intensity and thus, robust to shadows and varying lighting conditions, with mathematical transformation given as:

$$\begin{pmatrix} I \\ r \\ g \end{pmatrix} = \begin{pmatrix} (R+G+G)/3 \\ R/(R+G+B) \\ G/(R+G+B) \end{pmatrix}$$

in which $r$ and $g$ represent the normalized colors red and green, and $I$ is the intensity of the RGB color pixel.

## 3.2 Previous Work

Up until now, a large amount of work has been proposed to address the issues of background model representation and adaption. These algorithms share the common assumption that an initial background model can be obtained by employing a period of training sequence, in which no foreground objects are present [22], [40], [58].

In some cases it is impractical or impossible to record a separate background sequence without objects' interference, e.g., airport terminals. Further, it is expensive to clear up the scene from foreground objects on a regular basis since this procedure must be repeated every time there is even a small change in the background.

Median filter [17] and Kalman filter [51] were proposed to address the above limitations. They provide a good background model when the scene is not too crowded, i.e., at every pixel, the background should be visible more than fifty percent of the time during the training sequence. If the assumption fails, the output background model contains blending pixels or areas of error.

Long and Yang [40] presented an adaptive smoothness method based on the assumption that the longest and most stable interval of pixel intensity is most likely to represent the background. At each pixel, a moving window along time is employed to search for the stable intervals. However, when the data include multi-modal distributions (i.e., dynamic backgrounds including swaying trees) and when the modes from foreground objects are stationary for a long period of time, a lot of pixels are incorrectly classified.

Gutchess and Trajkovic [20] presented an algorithm, called the *Local Image Flow* algorithm which utilizes the cue of local optical flow fields in the neighborhood of each pixel to compute the probability of each stable interval of intensity being the background and then selects the most probable one to represent the background. Since the result of computation of local optical flow is sensitive to the insignificant movement of other objects in the scenario or the interaction between objects, the algorithm is less robust to deal with complex situations.

Stauffer et al. [53] designed a mixture of Gaussian (MoG) model, currently one of the most popular methods, to effectively deal with the dynamic background and light changes. Each pixel in MoG is modeled by three to five Gaussian distributions, which are adaptively adjusted as each new frame is coming. The main deficiency of MoG is that it is time consuming. Some researches are dedicated in improving the learning rate of MoG [33]. Another serious problem of MoG is that the slow moving objects and zooming objects are easily separated as referenced background, which makes the foreground information lost.

To overcome the problem of inaccurate background model caused by errors in parameter estimation methods such as MoG, Elgammal, Harwood et al. [15] proposed a nonparametric model for background modeling. The model employs a kernel estimator to determine the type of current pixel value based on recently observed values for this pixel. However, several pre-calculated lookup tables for the kernel function values are required to reduce the burden of computation in this approach. Also, this method can not resist the influence of foreground objects in the training stage.

Pixel-based methods mentioned above assume that the time series of observations is independent on each pixel. Region- or frame-based models are recently developed in [9], [58], [46], [13]. These models consider pixel as correlated random variables, and estimate probabilities based on their neighborhood relationships.

Lipton and Haering [39] hold the philosophy that the background pixels share some chromatic and temporal overlap with their neighbors. Such kind of similarities between pixels are served to cast votes for each other in the local region to determine pixel's background period. The disadvantage of this algorithm is that the output performance is highly related to the initial guess of each pixel's background mode.

Oliver et al. [48] proposed an approach called *Eigenbackgrounds* which captures spatial correlations by applying principal component analysis to a set of $N_L$ video frames that do not contain any foreground objects. This results in a set of basis functions of which only the first $d$ basic functions are required to capture the primary appearance characteristics of these frames. A new frame can then be projected into the eigenspace defined by these $d$ basis functions and then back projected into the original image space. Since the basis functions only model the static part of the scene when no foreground objects are present, the back projected image does not contain any foreground objects. As such, it can be used as a background model. The limitation of this approach is that computing the basis functions requires a set of video frames without foreground objects.

Li et al. [36], [37], [35], [38] proposed a generalized framework using Bayesian theory to update the background model by frame-based features. The approach is reported to be robust to illumination changes (gradual and sudden "once-off"), dynamic backgrounds (i.e., waving trees, rippling water surfaces) and changed background (add, remove and relocate background objects). The shortcoming is that if the foreground objects remain motionless in the scene for a while, they are quickly absorbed into the background.

Due to its pervasiveness in various contexts, background subtraction has been investigated by many researchers with plenty of published literature (see surveys in [49], [50]). There is no unique classification of proposed methods. The following include some usually referred dichotomies [43]:

- Parametric versus nonparametric: Parametric models (e.g., [60]) are tightly coupled with underlying assumption, do not always perfectly correspond to the real data, and the choice of parameters can be cumbersome, thus reducing automation. On the other hand, nonparametric models (e.g., [15], [30]) are more flexible but heavily data dependent.

- Unimodal versus multimodal: Basic background models assume that the intensity values of a pixel can be modeled by a single unimodal distribution (e.g., [60], [17], [20]). Such models usually have low complexity, but cannot handle moving backgrounds, while this is possible with multimodal models (e.g., [53], [58]), but at the price of higher complexity.

- Recursive versus nonrecursive: Nonrecursive techniques (e.g., [60], [53]) store a buffer of a certain number of previous sequence frames and estimate the background model based on the temporal variation of each pixel within the buffer, while recursive techniques (e.g., [58], [10]) recursively update a single background model based on each input frame. In nonrecursive techniques, the background well adapts to eventual variations, but memory requirements can be significant while in recursive techniques, space complexity is lower, but input frames from distant past can have an effect on the current background, and therefore any error in the background model is carried out for a long time period.

- Pixel-based versus region-based: Pixel-based methods (e.g., [60], [30], [53]) assume that the time series of observations is independent at each pixel, while region-based methods (i.e., [58], [15]) take advantage of inter-pixel relations, segmenting the images into regions or refining the low-level classification obtained at the pixel level.

This step obviously increases the overall complexity.

## 3.3 k-Means clustering Method

Background estimation using the median intensity value for each pixel was first used in a traffic monitoring system [17]. In [28], a histogram is constructed to determine the confidence of median value as the estimated background pixel.

Figure 3.2 shows a *history map* to represent intensity characteristics of a pixel over a time period. In this figure, the *observed images* are the image frames obtained from the video over a period of time. The *cell string* represents entire intensity variation at each pixel location over all the frames, and represents the *D* dataset. Each pixel has three components H, S and V which are obtained from R, G and B.

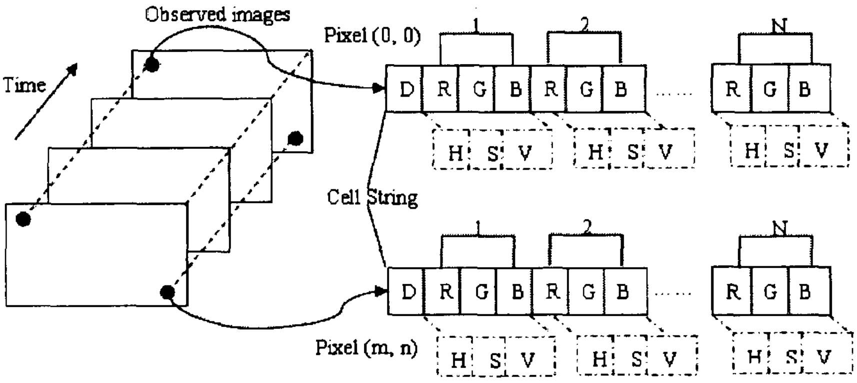


Figure 3.2: History map: A sample overview of how pixels intensities change with time

In general, a histogram displays continuous data in ordered columns or *bins*, also referred to as *bars* [27]. The bars must be adjacent and the intervals are generally of the same size. It is an effective way to represent continuous measures, such as time or intensity. In [28], histogram is used to identify a stable intensity value of each pixel in a certain time interval.

The histogram based median method for background modeling is based on a simple observation that the more often a pixel takes a particular stable color, the more likelihood it belongs to the background. To locate these pixels, the histogram is used to identify how often a particular pixel takes a specific color value.

More specifically, the pixel's HSV values among the training frames are stored, and then the histogram of 4 bins for each pixel's color component is constructed. The median value of the largest bin from the histogram is chosen as the estimated background model's color value. As an example in Figure 3.3, the HSV values for the pixel $(x,y)$ in the background model are $(0.61, 0.19, 0.34)$. The same procedure is applied on all pixels and the background model is then established.

The next step is to separate the foreground objects from the background through segmentation. In [28], k-Means clustering method is used. It can find a group of pixels similar in color, position or a combination of both. The value of k represents the number of clusters to be formed among the pixels of the image. The similarity measure between the feature vectors is calculated by the Euclidean distance function. The detailed steps are as follows:

1. Convert current video frame into HSV color space;

2. Subtract the H, S and V component of background model (obtained using histogram based median method) from the H, S and V components of the video frame and store their absolute values into a difference matrix;

3. Find minimum and maximum values in the difference matrix. The minimum value

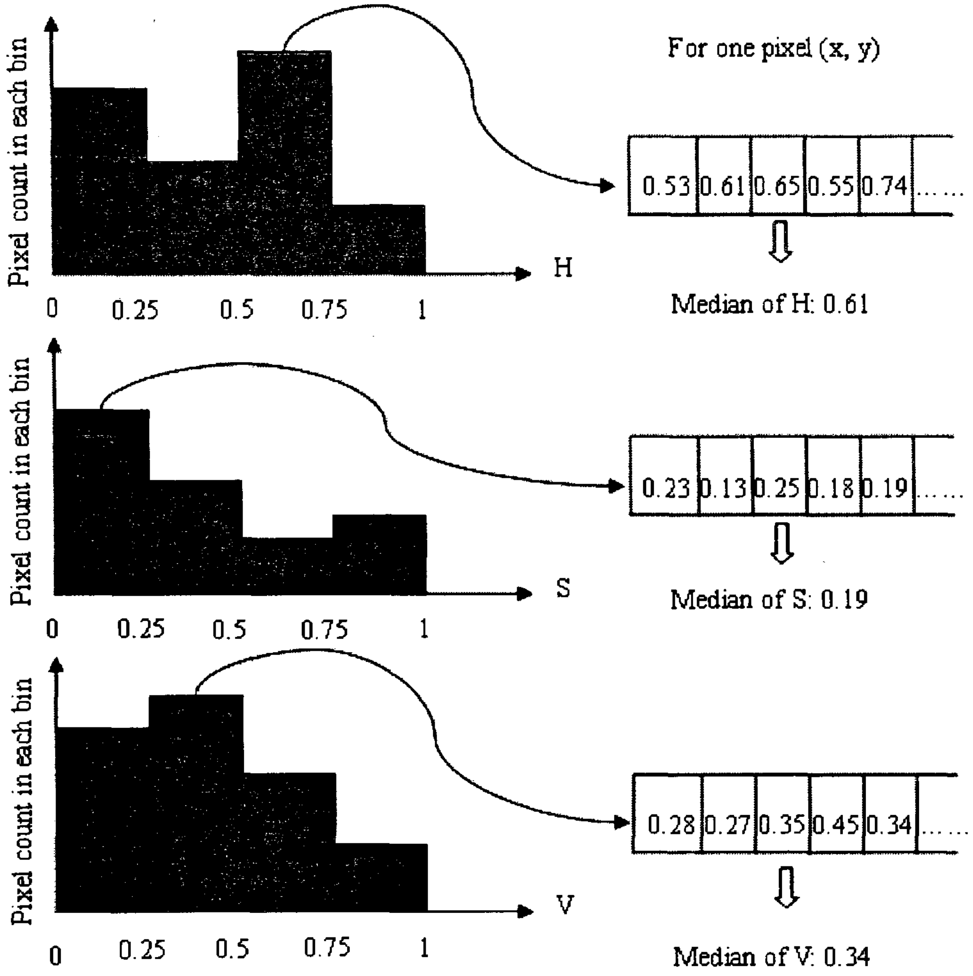


Figure 3.3: Background modeling process using histogram based median method

corresponds to the seed of the background cluster, denoted by M1 whereas the maximum value corresponds to the seed of the foreground cluster, denoted by M2;

4. If $D_{i1} > D_{i2}$, then assign $i$ to foreground cluster, otherwise assign $i$ to the background cluster where $D_{i1}$ is pixel's distance to the centroid of the background cluster whereas $D_{i2}$ is pixel's distance to the centroid of the foreground cluster;
5. Recalculate mean of the background cluster M1 and mean of the foreground cluster M2;
6. Repeat step 4 and step 5 until M1 and M2 do not change significantly (i.e., the difference between the two mean is approximately zero, e.g. 0.001 or so);
7. Report pixels in foreground cluster as the foreground region and also the pixels in the background cluster as the background region.

## 3.4 Mixture of Gaussians

In background modeling, pixel intensity is the most commonly used feature. If we monitor the intensity values of a pixel over time in a relatively static scene, then the pixel intensity can be reasonably modeled with a Gaussian distribution $N(\mu, \sigma^2)$, given that the image noise over time can be modeled by a zero mean Gaussian distribution $N(0, \sigma^2)$. This Gaussian distribution model [60] for the intensity value of a pixel is the underlying model for many background subtraction techniques. However, such single-mode models cannot handle dynamic backgrounds, such as waving trees, lighting changes and shadow casts.

Typically, in outdoor environments with moving trees and bushes, the background scene is not completely static. For example, a particular pixel may belong to sky in one frame, and leaf of a tree in another frame, a tree branch in a third frame, and so on. In each situation, the pixel will have a different intensity or color, so a single Gaussian assumption for the probability density function of the pixel intensity will not hold. Instead, a generalization based on a mixture of Gaussians is proposed by Stauffer and Grimson [18], [53], [54] to model such variations. In this method, the pixel intensity is modeled by a mixture of $K$ Gaussian distributions ($K$ is a small number from 3 to 5). They report good results on outdoor scenes, especially in case of multi-modal backgrounds. The work of [56] is close to Stauffer and Grimson's work. It employs incremental update algorithm which is Expectation Maximization (EM) algorithm for parameters of MoG.

In [56], colors of a pixel in a background object among frames are also described by multiple Gaussian distributions. The *mode*[2] of the 3D RGB color histogram for one pixel over time can be considered as the background color for that pixel.

Due to the fact that the normalized *rg* color space can effectively deal with shadows and varying lightings, 3D color space in [56] is converted to normalized *rg* color space. That's to say in RGB color space, a pixel can be denoted by a triplet $(R,G,B)$ in a frame and now can be denoted by a tuple $(r,g)$. Then the mode of the 2D *rg* color histogram for one pixel over time could be considered as the background color for the pixel. Moreover, the channel $r$ and the channel $g$ are uncorrelated, instead of estimating a 2D *rg*-histogram, two 1D histograms $r$ and $g$ are separately estimated.

The 1D color histograms $r$ and $g$ then can be modeled by the mixture model [6], which is given as below:

[2] In statistics, the mode is the value that occurs the most frequently in a data set or a probability distribution. In Gaussian distribution, the mean, median and mode all coincide. Available at http://en.wikipedia.org/wiki/Mode_(statistics)

$$p(x) = \sum_{j=1}^{K} \pi_j f_j(x;\mu_j,\sigma_j)$$

where $\pi_j$ is the prior of the Gaussian distribution $f_j$ with mean $\mu_j$ and standard deviation $\sigma_j$, and $x$ is the color value $r$ or $g$ and $K$ is the number of Gaussians or *kernels*. Figure 3.4 is an example of Mixture of Gaussians distribution with $K = 3$ used in this scheme.

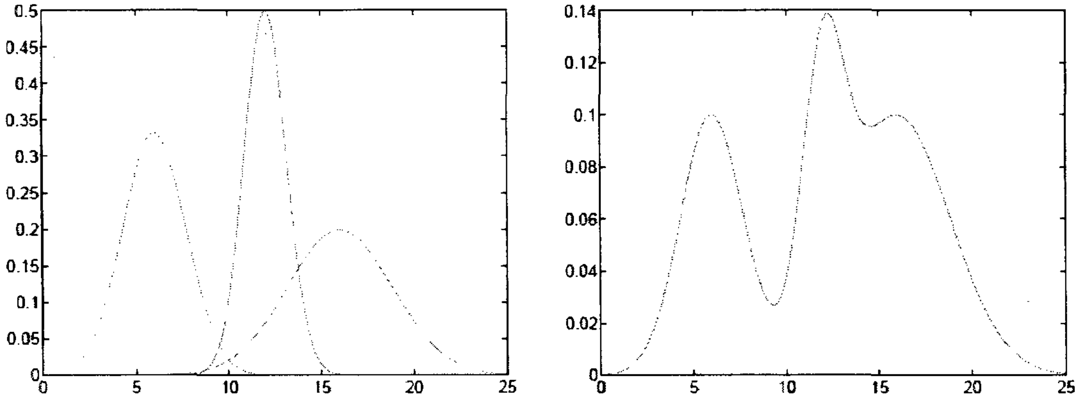


Figure 3.4: One dimensional Mixture of Gaussians model consisting of 3 single Gaussians

At first, the three Gaussians' parameters are set as $\pi_j = 1/K$, $\mu_j = j/(K+1)$ and $\sigma_j = 0.01/K$ which means the $K$ number of kernels are equally distributed over the normalized $r$ and $g$ channels. Then MoG method estimates and updates the parameters $\pi_j$, $\mu_j$ and $\sigma_j$ for both $r$ and $g$ channels in the mixture model formula by Expectation Maximization algorithm [12], which is the incremental variant of the maximum-likelihood estimation:

$$\pi_j \leftarrow \pi_j + \gamma(P\{j|x\} - \pi_j)$$

$$\mu_j \leftarrow \mu_j + \frac{\gamma}{\pi_j} P\{j|x\}(x - \mu_j)$$

$$\sigma_j^2 \leftarrow \sigma_j^2 + \frac{\gamma}{\pi_j} P\{j|x\}((x - \mu_j)^2 - \sigma_j^2)$$

in which

$$P\{j|x\} = \frac{\pi_j f_j(x;\mu_j,\sigma_j)}{p(x)}$$

is the posterior - the chance that the color value $x$ is part of kernel $j$. The variable $\gamma$ is the learning rate and indicates to what amount the parameters are updated. For one complete update of all of the parameters, one should apply all of these formulas over the range $j = \{1,\ldots,K\}$ for both $r$ and $g$ channels of each pixel in the image frame.

Figure 3.5 is the illustration of the background modeling process using MoG method. The mode of $p(x)$ is assumed to be the background color whereas the estimated means of $r$ and $g$ corresponding to the highest priors.

After learning the mode of each background pixel, we can determine a pixel in the current frame to be part of the foreground if the differences between both the normalized color values and the estimated means corresponding to the highest priors are larger than 2.576 $\sigma$ (i.e., 99% confidence).

## 3.5 Summary

In this chapter, we have discussed different approaches proposed in the literature for background modeling and foreground segmentation. The two algorithms - k-Means clustering method which is the basis of the proposed method and Mixture of Gaussians which is widely investigated in literature are discussed in detail.

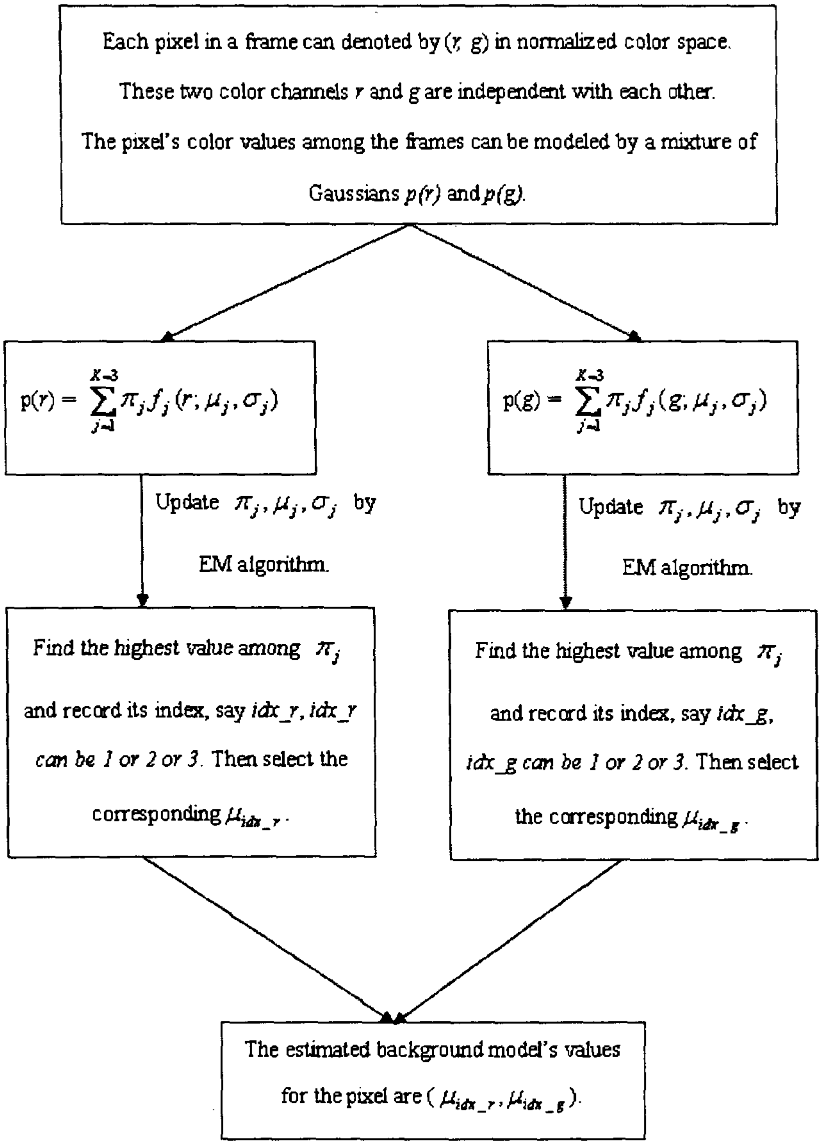

Each pixel in a frame can denoted by (r, g) in normalized color space.
These two color channels r and g are independent with each other.
The pixel's color values among the frames can be modeled by a mixture of Gaussians p(r) and p(g).
p(r) = $\sum_{j=1}^{K=3} \pi_j f_j(r; \mu_j, \sigma_j)$
p(g) = $\sum_{j=1}^{K=3} \pi_j f_j(g; \mu_j, \sigma_j)$
Update $\pi_j, \mu_j, \sigma_j$ by EM algorithm.
Update $\pi_j, \mu_j, \sigma_j$ by EM algorithm.
Find the highest value among $\pi_j$ and record its index, say idx_r, idx_r can be 1 or 2 or 3. Then select the corresponding $\mu_{idx\_r}$.
Find the highest value among $\pi_j$ and record its index, say idx_g, idx_g can be 1 or 2 or 3. Then select the corresponding $\mu_{idx\_g}$.
The estimated background model's values for the pixel are ( $\mu_{idx\_r}, \mu_{idx\_g}$ ).

Figure 3.5: Background modeling process using MoG method

# Chapter 4

# The Proposed Method

This chapter describes details of the proposed method. The background modeling approach using a histogram based median method is presented, which dynamically models the background without prior knowledge of the scene. Then a fuzzy k-Means clustering method is presented to segment the foreground objects from the background.

## 4.1 Statistical Background Modeling

Let $I_t(x,y)$ is the intensity of a pixel $p(x,y)$ in a frame $t$ such that $1 \leq t \leq N'$ where $N'$ is the total number of frames in an image sequence or video. The background model is built by the training sequence $\{I_1(x,y),\ldots,I_N(x,y)\}$ where $N$ is the number of image frames used in the training period such that $N \leq N'$.

To make the task feasible, we make the following assumptions, similar to those in [20], [28]:

1. Each pixel in the image reveals the background for a short interval of the sequence such that one bin out of the four reaches 50% of the total probability.

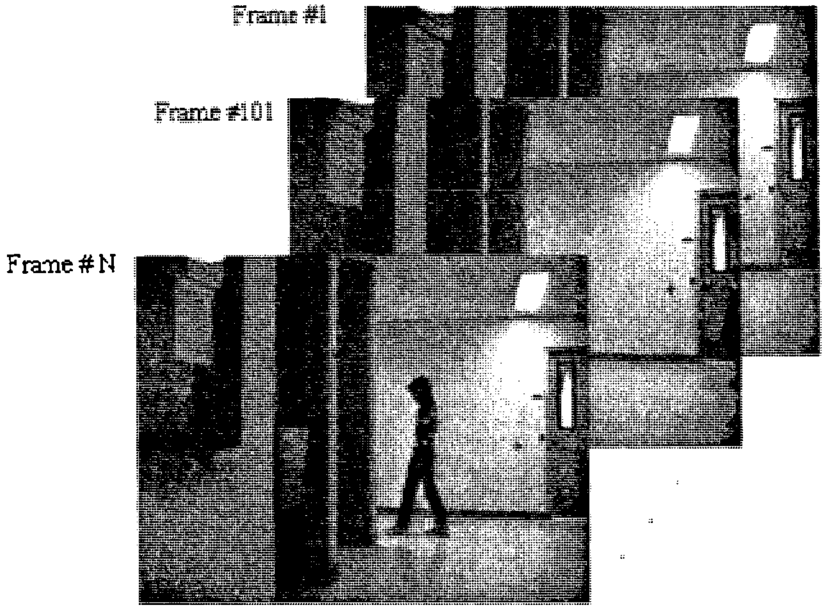
2. The background is nearly stationary with stable and constant illumination.
3. A short processing delay is allowed subsequent to acquiring the training sequence.

Figure 4.1-(a) is an illustration of *N* frames from a video used for background modeling. Figure 4.1-(b) shows an intensity typical plot over time for only one pixel in the image marked by the plus sign in each of these frames. The human object enters the scene around Frame 100 but occludes the marked pixel around Frame 119.

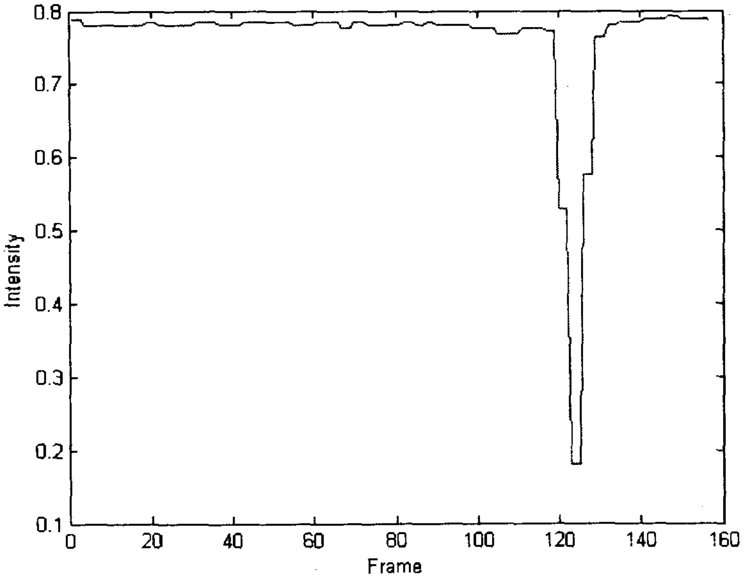
From Figure 4.1-(b), it can be seen that for the same pixel, the intensity is quite stable as a background pixel but it changes significantly when it is occluded by a pixel in the foreground object. The histogram for the intensity values of the same pixel can better illustrate the stable value among the frames as shown in Figure 4.1-(c).

The histogram based median method for background modeling can deal with the so-called *bootstrapping* problem - it can model the background model even in the presence of foreground objects. The use of HSV color space gives advantage of noise removal. The detailed steps of the scheme are as follows [28]:

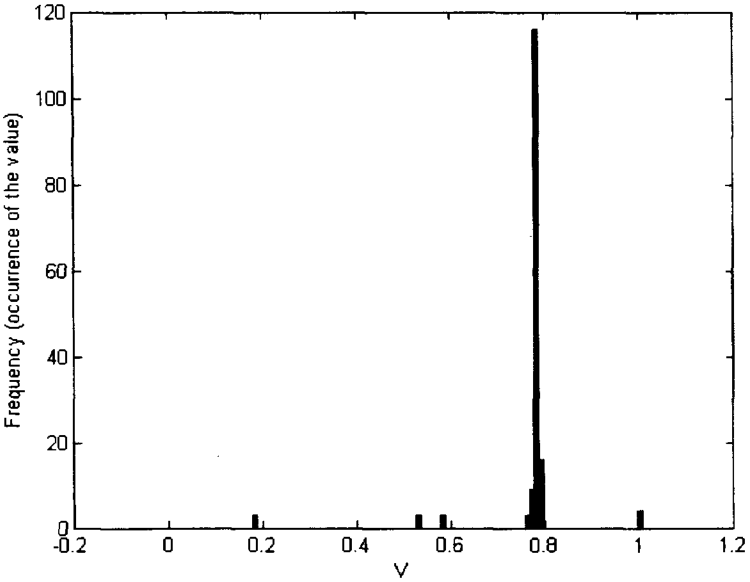
1. Convert each of the *N* frames of the training sequence from RGB color space to HSV color space.
2. For each pixel among the frames, construct a 4-bins histogram on its intensity values, or the V component (For the purpose of displaying the background, all of the three HSV components are needed but for processing, only V component is significant and needed).
3. Find a bin with the highest frequency of intensity values from the histogram.
4. Our objective is to locate the highest bin and pick up the intensity values among frames falling into the highest bin. In the general case, we may scan the intensity



(a) N frames from a sample video



(b)The intensity values of the pixel denoted by a plus sign among the N frames



(c)The histogram of V component of the pixel denoted by the plus sign among the N frames

Figure 4.1: Background modeling of a single pixel using its statistical information

values first, and establish histogram to find out the index of the highest bin. Then we have to scan the intensity values again to record the intensity values fall into the range of the highest bin. Here, we want to scan one time only to find out those intensity values to reduce processing time. We use the idea of *flags* which offer a space-time tradeoff. For each pixel among frames, establish *flags* which record the bin numbers the intensity values fall into. For example, in Frame 1, the intensity value falls into the second bin in V histogram, the flag for this pixel in Frame 1 is 2.

5. Select intensity values whose flags are the same as the highest bin number and store them in a buffer and discard it otherwise.

6. Choose the median of all the intensity values in the buffer and consider it as the intensity value of the pixel in the background model.

7. Perform step 2-6 until all pixels are processed.

After background model is established, segmentation for foreground objects can be done through the background subtraction method as explained in subsequent sections.

## 4.2 Adaptive Foreground Segmentation

Foreground segmentation is the process of classifying each pixel either as the background or the foreground. By background subtraction, one can detect the motion in a given frame. If a pixel is part of the background in current frame, the difference between its intensity value in current frame and that of the background model approximates to zero and is far from zero otherwise. Further, it is important to note that the foreground objects do not enter the scene as isolated pixels but as a group of adjacent pixels and exhibit similarity in color or intensity.

Clustering is the process of dividing data items into classes or clusters. Clustering algorithms attempt to segregate a data set into distinct regions of membership with an essential goal to minimize the disperse between data items within a cluster and maximizing it among the data items in different clusters. Generally speaking, clustering techniques can be classified into two broad categories [16]: (1) exclusive clustering and (2) overlapping clustering.

In exclusive clustering, assignment of a data item to a particular cluster is made on the basis of hard or crisp decision. Therefore, a data item can belong to one and only one cluster. If we need to cluster a set $\chi = \{x_1, x_2, \ldots, x_N\}$ of $N$ data items, the goal of exclusive clustering is to partition $\chi$ into $k$, $2 \leq k \leq N$ disjoint, non-empty clusters $\{C_1, C_2, \ldots, C_k\}$ such that:

$$\chi = C_1 \cup C_2 \cup \ldots \cup C_k$$

where:

$$C_i \cap C_j = 0, i, j \in \{1, \ldots, k\} \quad \textit{and}$$

$$C_i \neq 0, i \in \{1, \ldots, k\}$$

In overlapping clustering, also known as the fuzzy clustering, a data item may belong to more than one cluster with different degrees of memberships. The membership values typically range between 0 and 1 and indicate the extent to which a particular data item may belong to a certain cluster.

Bezdeck [5] proposed a family of fuzzy clustering algorithms. Such algorithms consider each cluster as a fuzzy set while a membership function measures the probability that each training sample belongs to a cluster. As a result, each sample may be assigned to multiple clusters with some degree of certainty, as determined by the membership function. Therefore, partitioning of data is based on soft decision. Fuzzy k-Means clustering [26] facilitates identification of overlapping groups of objects by allowing the objects to belong to more than one group. The essential difference between fuzzy k-Means clustering and standard k-Means clustering is primarily the way data is partitioned among the clusters.

In order for us to be able to properly detect foreground objects, it is important to distinguish between foreground and background pixels. Since the change in intensity value can be either due to change in the characteristics of the background or due to introduction of foreground objects, we believe that the fuzzy k-Means is more appropriate to establish suitability of each pixel as background or foreground due to its varying degree of membership. Further, the membership values of a pixel may change over time as the scene progresses. Fuzzy k-Means in general improves the standard k-Means clustering by finding better centers of clusters and has better time performance for convergence [47].

The fuzzy k-Means algorithm for detection of foreground objects is as followings:

1. Set the number of clusters $k = 2$: a background cluster and a foreground cluster and set the blending factor, or the extent of overlapping between clusters $b = 1.25$ [62].

2. Initialize to compute the clusters' means. Subtract the current frame in HSV color space from the background model. It will give us a difference matrix. From the absolute values of the matrix, select the maximum and the minimum values of V component and use them as the the means $\mu_i$, $i = 1, 2$ of the two clusters $\omega_i$.

3. For each pixel $x_j$, calculate its distance $d_{ij}$ from the foreground mean and the background mean using absolute difference matrix where $i$ is the cluster, $j$ is the pixel such that $i = 1, 2$ and $1 \leq j \leq n$, $n$ is the total number of pixels in a frame. Here

we use Manhattan distance as the distance function since the feature used (i.e., V component) is only 1-dimensional and hence, requires less computations.

4. Compute the memberships of each pixel $x_j$ among the clusters $\omega_i$ as:

$$\hat{P}(\omega_i|x_j) = \frac{(1/d_{ij})^{2/(b-1)}}{\sum_{l=1}^{k}(1/d_{ij})^{2/(b-1)}}$$

5. Compute new means for each of the two clusters:

$$\mu_i = \frac{\sum_{j=1}^{n}[\hat{P}(\omega_i|x_j)]^b x_j}{\sum_{j=1}^{n}[\hat{P}(\omega_i|x_j)]^b}$$

6. If $\mu_i(new) - \mu_i(old) < \gamma$ where $\gamma$ is the established threshold, go to 7; otherwise replace old mean value with the new one and go to 3.

7. If $\hat{P}(\omega_{fg}|x_j) < \hat{P}(\omega_{bg}|x_j)$, $x_j$ belongs to the background cluster; otherwise it belongs to the foreground cluster.

At the end of the classification process, we obtain a binary image in which all of the foreground pixels are set to 1 whereas all of the background pixels are set to 0. Since the background modeling and foreground segmentation are based on pixel-level, the output may contain noise in form of spots, holes and spurs. Therefore, there may be a need to employ some post-processing algorithm to remove such noise or to shrink or fill these holes [56], [63], [23].

## 4.3 Summary

This chapter presents the proposed method and deals with the first two steps of intelligent visual surveillance systems. The background is modeled by statistical method and histogram analysis for bootstrapping in dynamic scene. The foreground segmentation is done by fuzzy k-Means clustering which automatically finds the grouping of foreground pixels and the background pixels. The fuzzy k-Means clustering method is more robust than the approaches which set thresholds to differentiate the foreground and the background pixels since the threshold may vary significantly from scene to scene.

# Chapter 5

# Experimental Results

This chapter presents some of the experimental results of the proposed method. These experiments are conducted on a series of indoor videos and on a set of benchmark videos. We have compared results with two other similar methods discussed before.

## 5.1 Comparative Results

In order to validate the proposed scheme, a series of experiments are conducted using a 2.4GHz Intel quad-core PC running Microsoft Windows XP. For testing purposes, all of the implementation are in MATLAB. All of the videos we recorded have dimensions of either 640x480 or 320x240 pixels. As suggested in [28] and [31], we chose to use 40 frames to establish the initial background model with a threshold for change $\gamma = 0.001$.

Figure 5.1 shows a set of sample frames for a video with 640x480 pixels dimensions. These frames are extracted from an indoor video taken in a corridor and represent a typical environment in video surveillance applications. The background modeling result of this set of images is given in Figure 5.2 - an almost perfect background image.

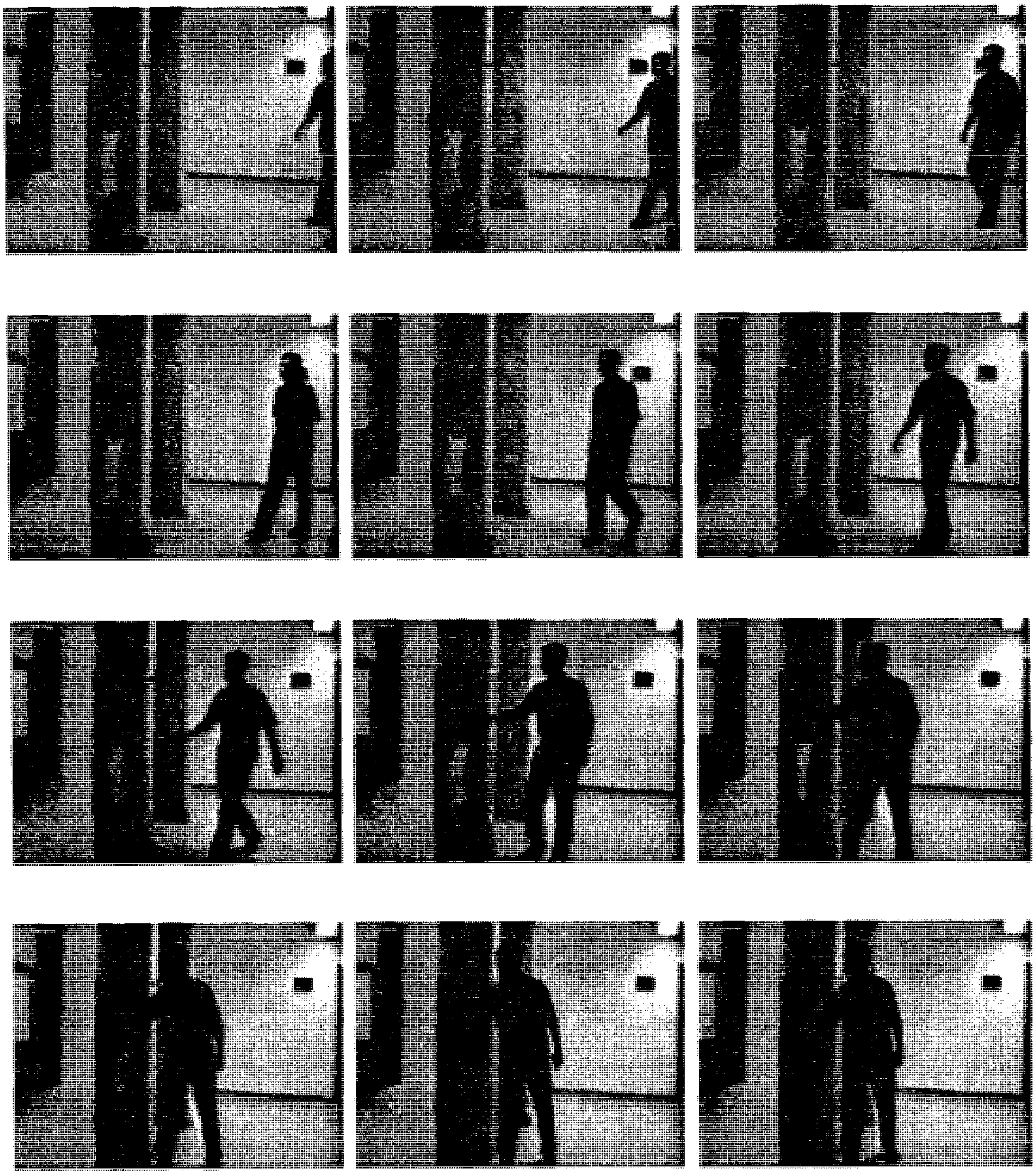


Figure 5.1: The images from an indoor video for background modeling taken in a corridor

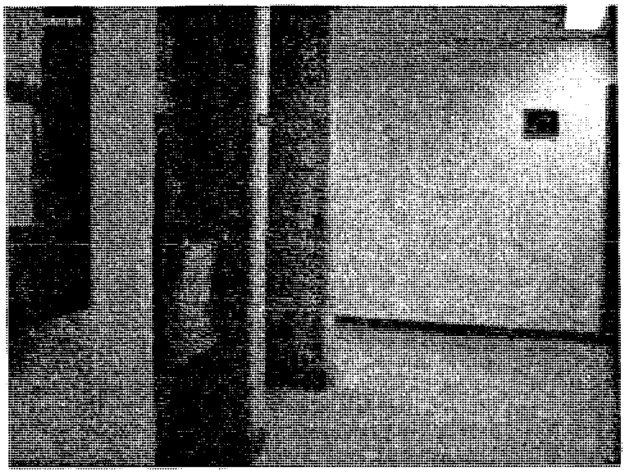


Figure 5.2: The estimated background using image sequence in Figure 5.1

In Figure 5.3, the image set is from a similar video but with a lower dimensions of 320x240 pixels. The established background modeling result for this image set is given in Figure 5.4.

The segmentation results using the discussed three methods can be found in Figure 5.5, Figure 5.6 and Figure 5.7. As one can observe, the proposed method can always provides better results whereas MoG fails. It is due to the underlying assumption of MoG that the color distribution of each pixel among frames can be modeled by a few number of Gaussians which do not always perfectly correspond to the real data. In Figure 5.3, the foreground object has a wide color distribution and that may lead to misclassifications in MoG.

Table 5.1 and Table 5.2 provide the processing time for each of these three methods for both background modeling and foreground segmentation. For background modeling, k-Means clustering method and our proposed method are faster than MoG but our proposed method provides better results. In foreground segmentation step, k-Means clustering is the slowest among all of the schemes compared. Even though the proposed algorithm is a bit slower than MoG but, as can be observed, provides better segmentation results than any of

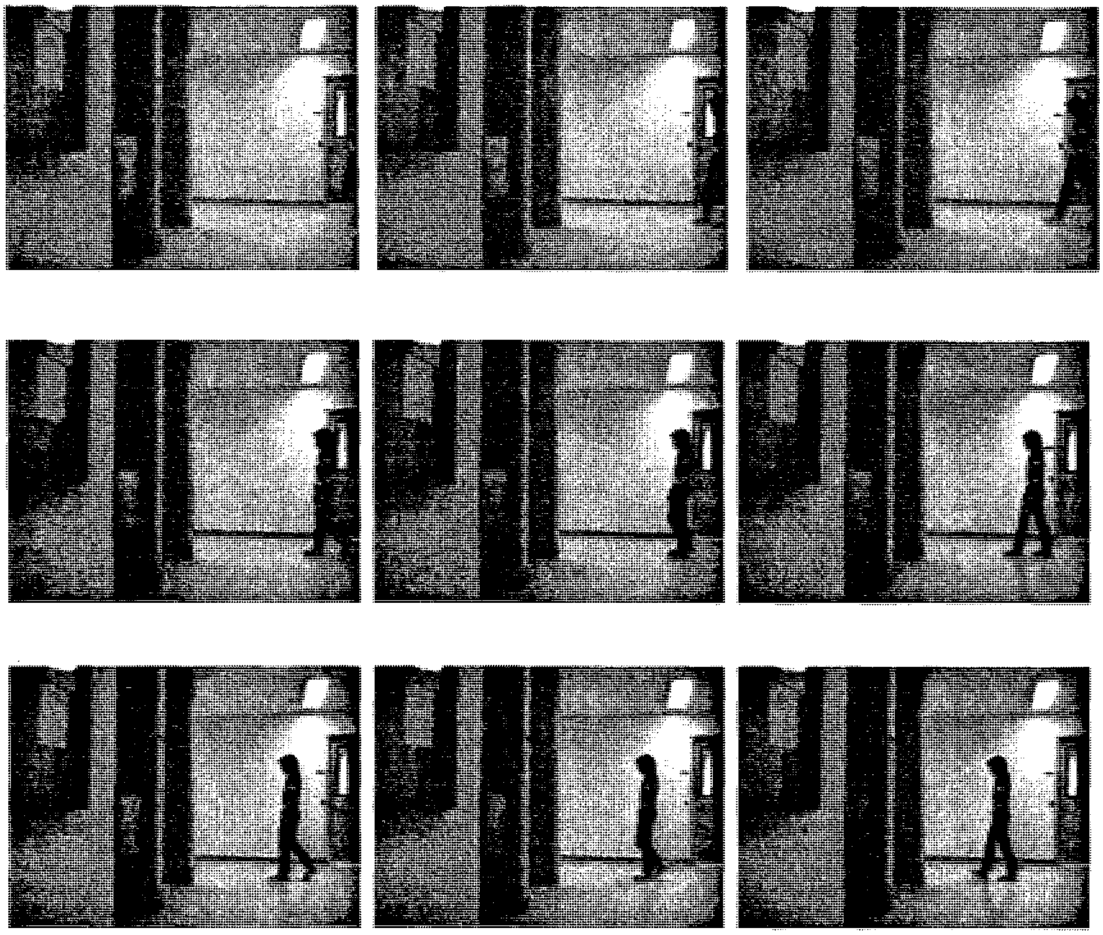


Figure 5.3: Another set of images from an indoor video but with a lower dimension

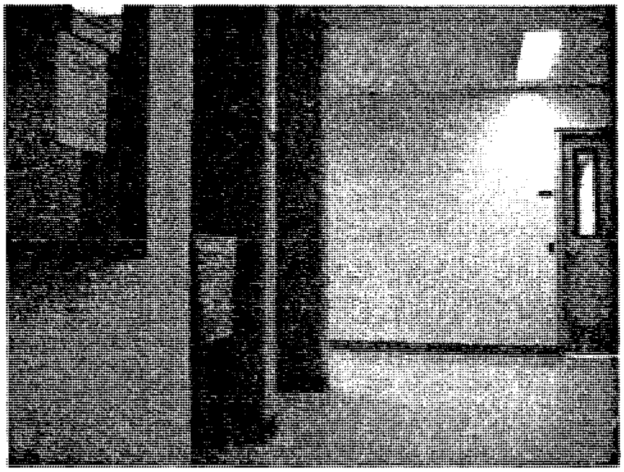


Figure 5.4: The estimated background using image sequence in Figure 5.3

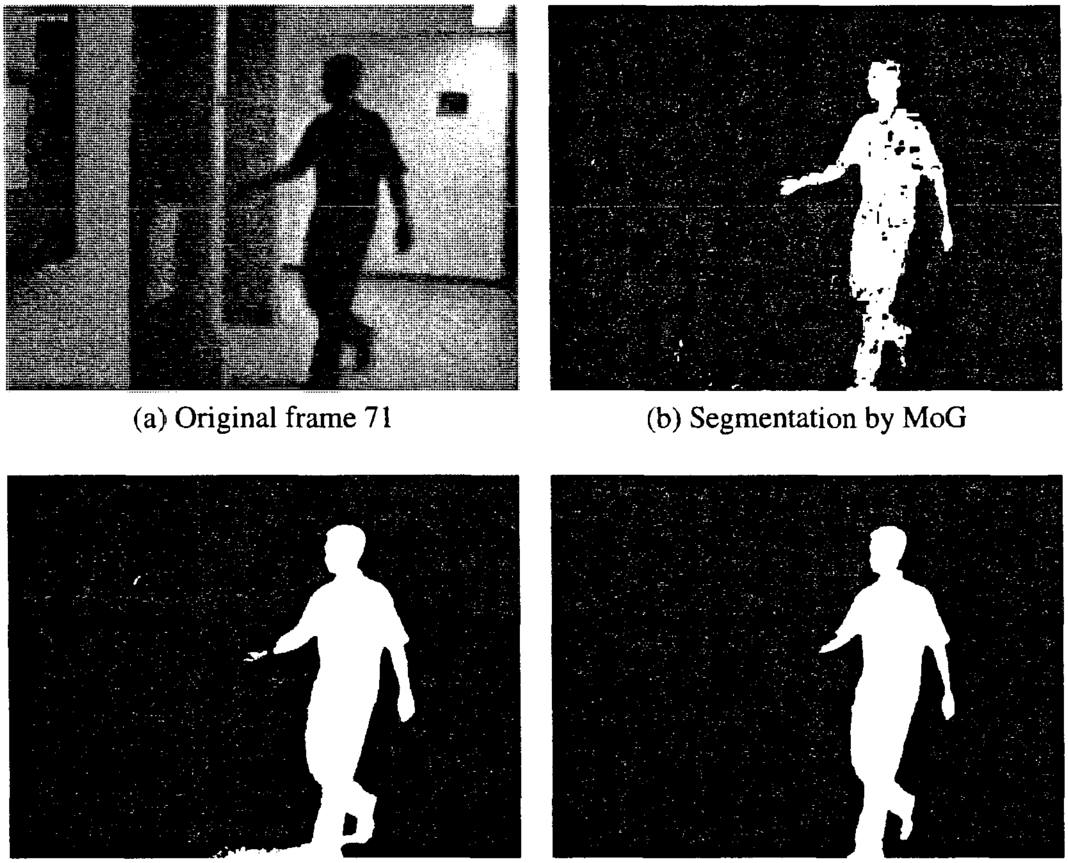
the other algorithms.

Table 5.1: The processing time corresponding to video in Figure 5.1

| Video size<br>640x480 pixels | modeling<br>(sec) | segmentation<br>(sec/frame) |
|---|---|---|
| k-Means Clustering | 354 | 535.0 |
| Mixture of Gaussians | 453 | 0.6 |
| The proposed method | 354 | 15.0 |

In addition to testing our approach on some in-house videos, we also conducted experiments on some of the benchmark videos in PETS series[3]. We conducted our experiments on 1452 frames in PETS2000 video, on 2688 frames in a PETS2001 video and on 2550 frames in PETS2006 video. PETS2000 and PETS2001 videos are outdoor video scenes with human objects and vehicles moving in and out of the scene. Figure 5.8 and Figure 5.9 show the segmentation results on some of the frames. PETS2006 is a video of an indoor scene captured in a subway with group of people. Some of the segmentation results for this

[3] Available at http://ftp.cs.rdg.ac.uk/



(a) Original frame 71

(b) Segmentation by MoG

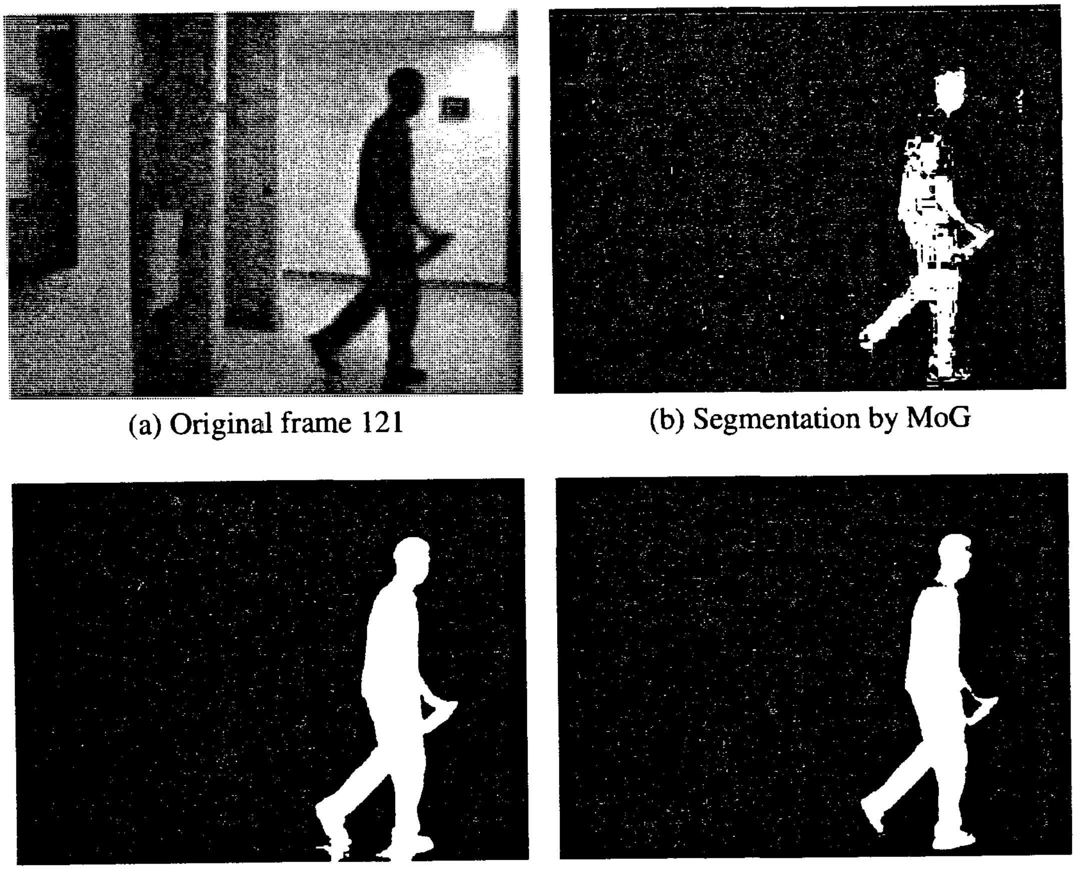
(c) Segmentation by k-Means clustering

(d) Segmentation by the proposed method

Figure 5.5: Segmentation results using 3 different methods on frame 71 of video in Figure 5.1

Table 5.2: The processing time corresponding to video in Figure 5.3

| Video Size<br>320x240 pixels | modeling<br>(sec) | segmentation<br>(sec/frame) |
|---|---|---|
| k-Means Clustering | 92 | 21.0 |
| Mixture of Gaussians | 136 | 0.2 |
| The proposed method | 92 | 2.0 |



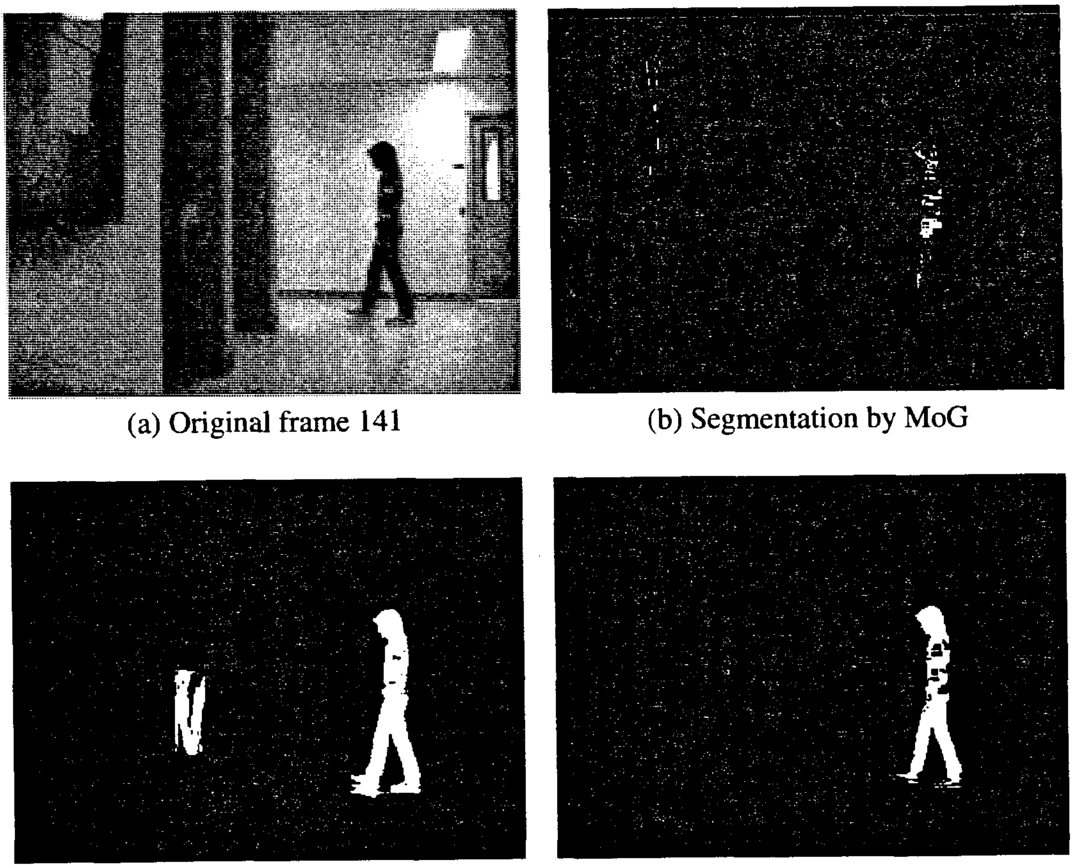
(a) Original frame 121

(b) Segmentation by MoG

(c) Segmentation by k-Means clustering

(d) Segmentation by the proposed method

Figure 5.6: Segmentation results using 3 different methods on frame 121 of video in Figure 5.1



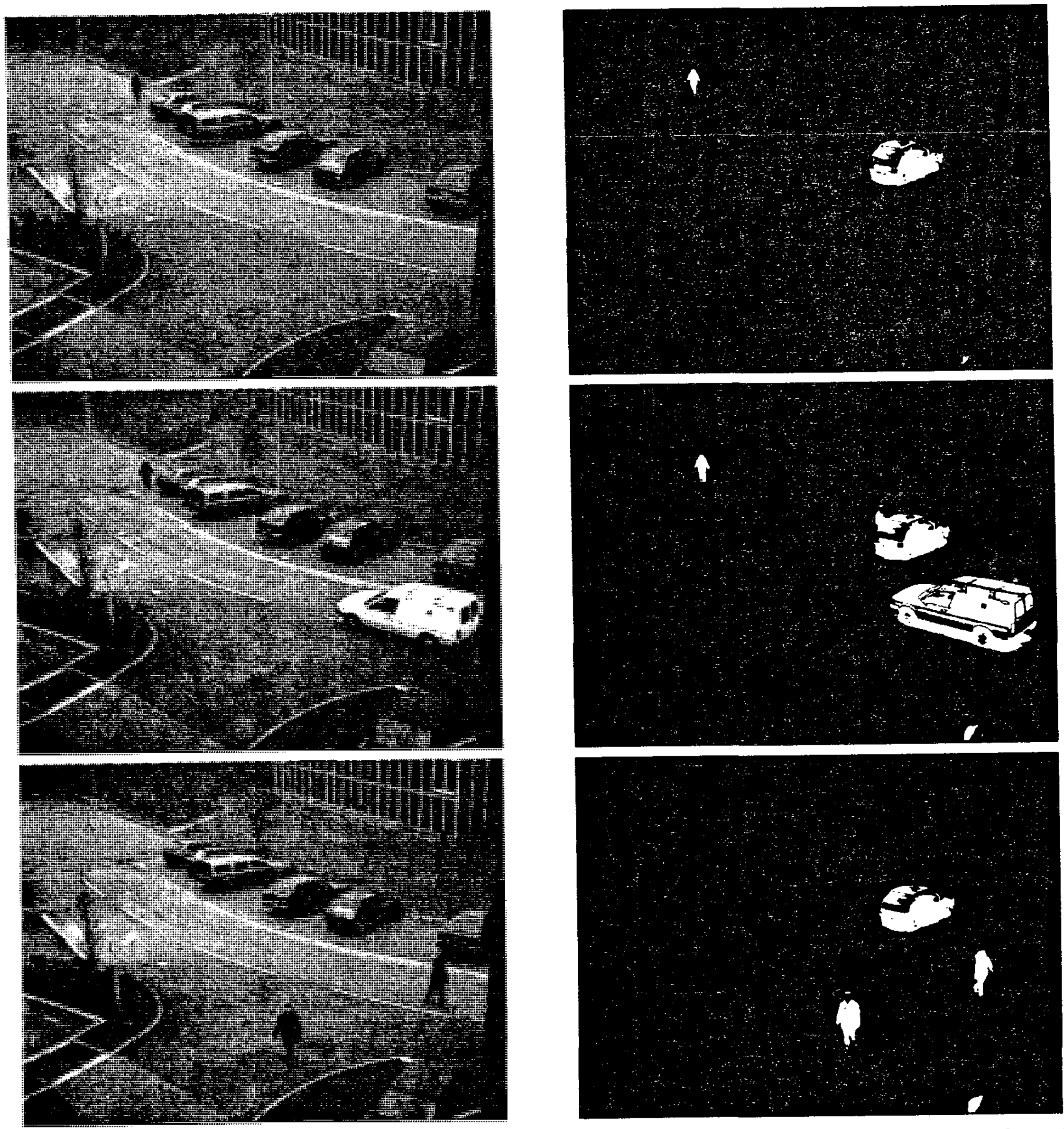
(a) Original frame 141

(b) Segmentation by MoG

(c) Segmentation by k-Means clustering

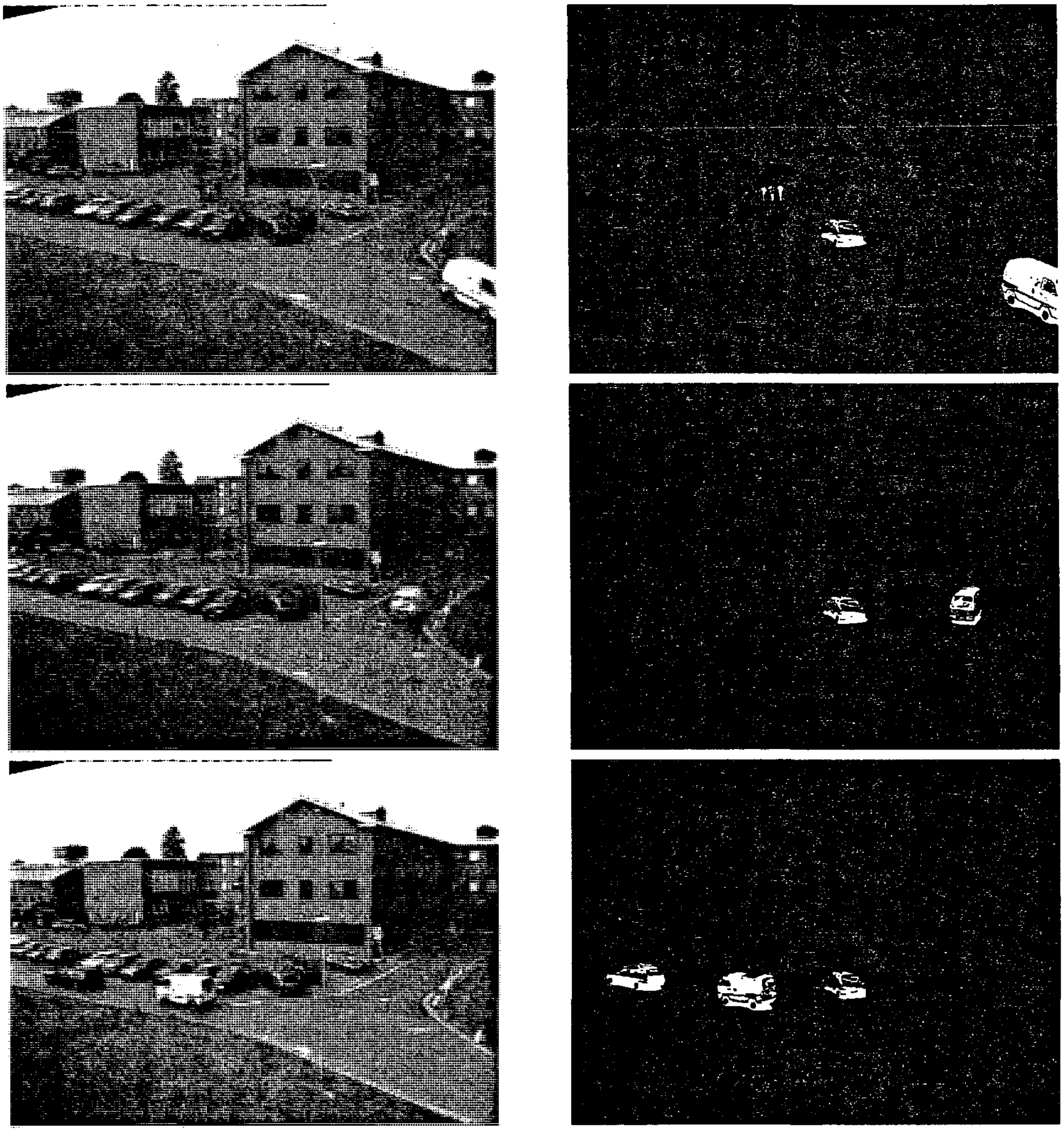
(d) Segmentation by the proposed method

Figure 5.7: Segmentation results using 3 different methods on frame 141 of video in Figure 5.3



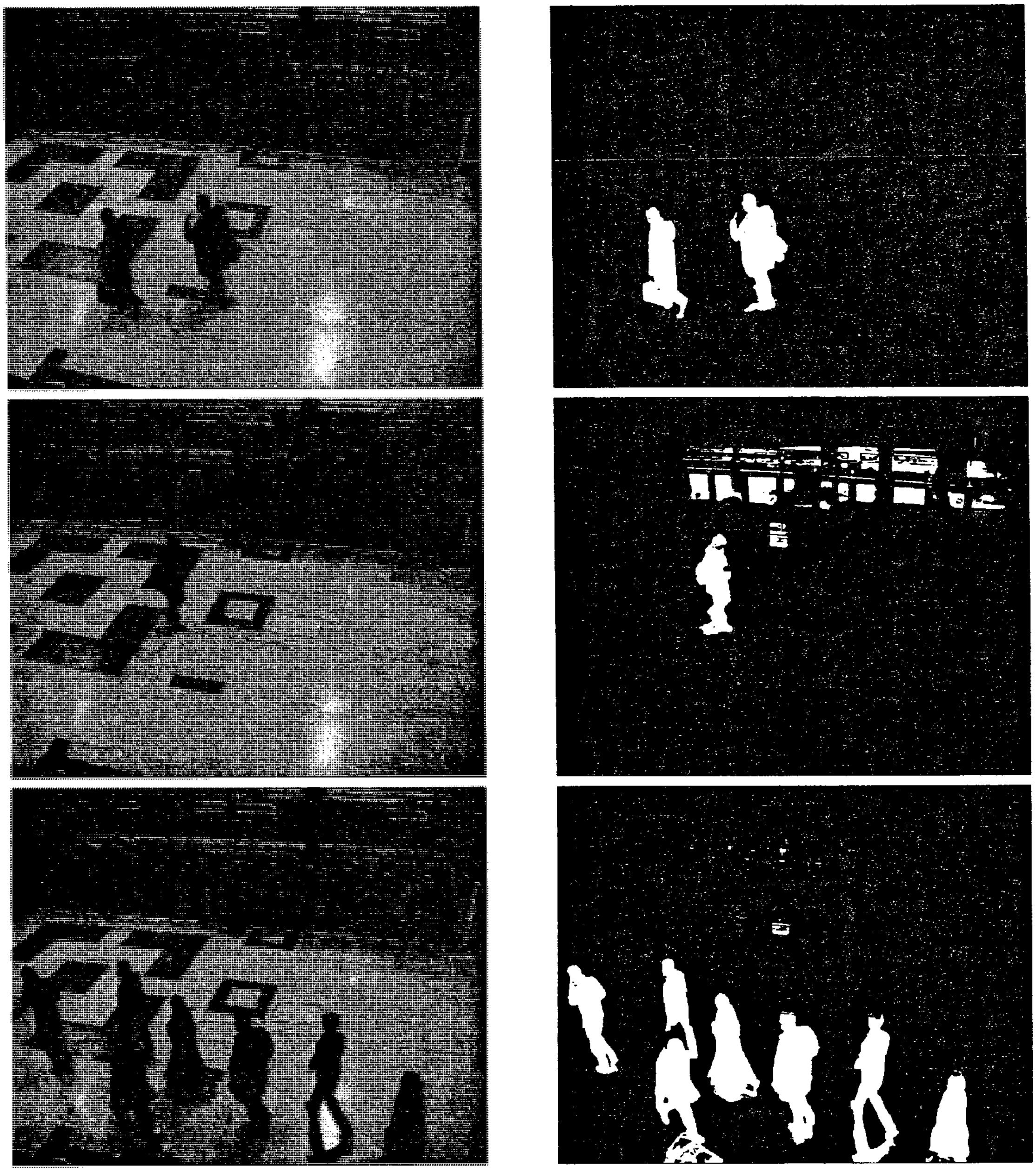
(a) original frames in PETS2000 (b) Segmentation results by the proposed method

Figure 5.8: Segmentation results on PETS2000 using the proposed method



(a) original frames in PETS2001 (b) Segmentation results by the proposed method

Figure 5.9: Segmentation results on a PETS2001 video using the proposed method



(a) original frames in PETS2006 (b) Segmentation results by the proposed method

Figure 5.10: Segmentation results on a PETS2006 video using the proposed method

video are shown in Figure 5.10. The corresponding processing time for the PETS series videos are listed in Table 5.3.

Table 5.3: The processing time of the proposed approach for PETS series videos

| Video size<br>(pixels) | # of frames | modeling<br>(sec) | segmentation<br>(sec/frame) |
|---|---|---|---|
| PETS2000 (768x576) | 1452 | 452 | 20.2 |
| PETS2001 (768x576) | 2688 | 447 | 22.4 |
| PETS2006 (720x576) | 2550 | 422 | 15.8 |

Table 5.4: The processing time of MoG for PETS series videos

| Video size<br>(pixels) | modeling<br>(sec) | segmentation<br>(sec/frame) |
|---|---|---|
| PETS2000 (768x576) | 628 | 0.8 |
| PETS2001 (768x576) | 631 | 0.9 |
| PETS2006 (720x576) | 615 | 0.7 |

Table 5.5: The processing time of k-Means clustering method for PETS series videos

| Video size<br>(pixels) | modeling<br>(sec) | segmentation<br>(sec/frame) |
|---|---|---|
| PETS2000 (768x576) | 452 | 843 |
| PETS2001 (768x576) | 447 | 861 |
| PETS2006 (720x576) | 422 | 789 |

## 5.2 Analysis and Discussion

From the previous section, we can see that the performance of our proposed method is better than the two similar contemporary methods. It can detect foreground objects like vehicles, single and group of humans in a scene. When there is no foreground object in the scene, the proposed method gives very limited isolated white pixels which are caused by image noise. Such kind of image noise can be easily removed by morphological operators, i.e., a pair of open and close [2], [56], [35].

There are some limitations of the proposed method. When the global illumination changes in the scene, e.g., in PETS2001, the whole frame is detected as foreground. The background subtraction technique is based on an accurate background model from which foreground objects can be detected. Thus, it is necessary to include the adaptability of background model, i.e., robust to gradual illumination and sudden illumination changes. The former one is easier than the latter one to deal with. A simple way to tolerate gradual illumination change is IIR (Infinite Impulse Response) filter. One feasible solution dealing with the latter problem is proposed by Li et al. [35].

Secondly, when the foreground object enters the scene and essentially becomes part of the background, for example, gray car (in the first original frame of Figure 5.8, the second car counting from right side in the parking lot) getting parked in Figure 5.8, green car (in the first original frame of Figure 5.9, the third car counting from right side in the parking lot) getting parked in Figure 5.9, the cars are not absorbed into the background. Therefore, in all of the following frames, the gray car and the green car are detected as the foreground objects. However, it is not always necessary to convert the foreground to the background, for example, in Figure 5.10, the luggage was dropped on the floor and alarms should be invoked to get the operator's attention. Different surveillance applications determine the actions surveillance systems should do.

Thirdly, there are some foreground object regions missing after detection, i.e., in Figure 5.8 and Figure 5.9. The vehicles are not totally detected. The reason for that is the intensity of the foreground object is too close to the background, this problem is also known as *camouflage*. More features such as spatial information can be incorporated to get rid of such kind of problems.

Shadows as in Figure 5.10 is another big problem. Due to the color space we used, most of the human objects' shadows are removed. But in the frame with group of persons, one severe shadow (in the third original frame of Figure 5.10, the second person counting from right side) can not be suppressed and needs additional techniques handling it.

Segmentation processing speed or convergence speed is the major problem in k-Means clustering method. The proposed scheme outperforms k-Means clustering method for segmentation processing time. For MoG method, when the background involves a wide distribution of color/ intensity, modeling the background with a mixture of a small number of Gaussian distribution is not efficient. Moreover, when foreground objects are included in the training frames, MoG doesn't work well and it will misclassify [59] as in Figure 5.7. For objects that present a slow motion, only the edges are highlighted. The central parts of the object are absorbed quickly, resulting in a set of sparse points of the object [23]. Finally, the background modeling time of MoG is higher than the k-Means clustering and the proposed method.

One may doubt whether our proposed method can be applied to real-time applications. The answer is yes because of the two reasons. One is the dimension of the image frames or videos. Most of the videos in real-life surveillance have a smaller dimension of 320x240 pixels or 160x120 pixels. Our test image sequences have much higher resolutions, such as 640x480 pixels, 768x576 pixels or 720x576 pixels. These sizes are equal to or more than four times of the size of 320x240 pixels, i.e. $\frac{640x480}{320x240} = 4$. As one can see in Table 5.1 and Table 5.2, when the dimension reduces, the processing time reduces significantly. The other reason is the implementation tool. MATLAB is much slower than other lower level developing tools, such as VC++ and OpenCV. The proposed algorithm developed by other tools will have much higher speed than in MATLAB.

There is one interesting note for the experiments which is the effectiveness using distance as the feature in k-Means clustering method and our proposed method. In k-Means clustering or fuzzy k-Means clustering, calculating the means or *centroids* of the clusters are based on the data set itself, in our case here, it's the difference matrix of V component. After one iteration, we calculate the average of the V values falling into foreground cluster and background cluster separately. However, if we use the average of the distances falling into foreground cluster and background cluster as the new means, the segmentation results are sometimes even better than using difference matrix. The reason behind that is probably the distance is served as weights. The smaller the distance value, the closer the pixel belongs to the cluster. But if there is no foreground object in the scene, this distance scheme may give unpredictable results.

## 5.3 Summary

This chapter presents background modeling results using histogram based median method and the segmentation results using the proposed method. The segmentation results using other two methods on the same video frame are presented too. From segmentation results and the processing time, it is obvious that the proposed method is better than the other two similar contemporary methods.

# Chapter 6

# Conclusions and Future Work

The separation of moving foreground objects from a static background is a crucial step in intelligent visual surveillance. Background subtraction including background modeling and foreground segmentation is the typical approach. The purpose of background modeling is to extract the empty scene model from a short training sequence where foreground objects may be present [8].

In this thesis, background is modeled by the median of every pixel's intensity over the sequence. The output background model is good when each background value appears for more than 50% of the sequence duration. The task of extracting foreground objects from the background is difficult due to the noise of the camera, shadows of the objects and the variations of the illumination even if the background is almost stationary. A fuzzy k-Means clustering method is proposed for segmentation purpose. The experimental results of indoor videos and benchmark videos prove the effectiveness of the proposed method.

Some of the future work includes the following:

1. In certain cases, the 50% assumption for background modeling may not be true because the video sequence to be initialized cannot be selected arbitrarily. Therefore,

there is a need to investigate a more robust background modeling approach.

2. Evaluation of the segmentation using quantitative measures as given in [35]. The segmentation results are compared with the ground truth in terms of the following two measures: *false negative error* - the number of foreground pixels that are missed; *false positive error* - the number of background pixels that are misdetected as foreground.

3. In current stage, we use one constant background model for foreground segmentation. If there is any change in the scene, i.e.,gradual illumination changes or changed backgrounds, our model can not learn the changes from the incoming frames. Background subtraction maintenance to deal with such problems needs to be investigated.

# Bibliography

[1] F. Alexandre and M. Gerald. Adaptive color background modeling for real-time segmentation of video streams. In *Proceedings of International Conference on Image Science, System and Technology*, pages 227–232, 1999.

[2] Mohammad Ahsan Ali. Feature-based tracking of multiple people for intelligent video surveillance. Master's thesis, University of Windsor (Canada), 2006. M.Sc.

[3] S. Apewokin, B. Valentine, L. Wills, and A. Gentile. Multimodal mean adaptive backgrounding for embedded Real-Time video surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[4] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[5] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algoritms.* Plenum Press, New York, 1981.

[6] C. M. Bishop. *Neural Networks for Pattern Recogntion.* Clarendon Press, Oxford, U.K., 1995.

[7] R. Chellappa, A.K. Roy-Chowdhury, and A. Kale. Human identification using gait and face. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–2, 2007.

[8] Chia-Chih Chen and J.K. Aggarwal. An adaptive background model initialization algorithm with objects moving at different depths. In *15th IEEE International Conference on Image Processing*, pages 2664–2667, 2008.

[9] M. Cristani, M. Bicego, and V. Murino. Integrated region- and pixel-based approach to background modelling. In *Proceedings of Workshop on Motion and Video Computing*, pages 3–8, 2002.

[10] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1337–1342, 2003.

[11] B. Coifman D. Beymer, P. McLauchlan and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[13] Xiaoyu Deng, Jiajun Bu, Zhi Yang, Chun Chen, and Yi Liu. A block-based background model for video surveillance. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1013–1016, 2008.

[14] J. Dever, N. da Vitoria Lobo, and M. Shah. Automatic visual recognition of armed robbery. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 1, pages 451–455, 2002.

[15] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *Computer Vision ECCV 2000*, pages 751–767. 2000.

[16] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–780, 1989.

[17] Brian Gloyer, Hamid K. Aghajan, Kai-Yeung Siu, Thomas Kailath, Robert L. Stevenson, and Sarah A. Rajala. Video-based freeway-monitoring system using recursive vehicle tracking. In *Image and Video Processing III*, volume 2421, pages 173–180, San Jose, CA, USA, March 1995. SPIE.

[18] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[19] Hongxing Guo, Yaling Dou, Ting Tian, Jingli Zhou, and Shengsheng Yu. A robust foreground segmentation method by temporal averaging multiple video frames. In *International Conference on Audio, Language and Image Processing*, pages 878–882, 2008.

[20] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A.K. Jain. A background model initialization algorithm for video surveillance. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, volume 1, pages 733–740, 2001.

[21] I. Haritaoglu, D. Harwood, and L.S. Davis. Hydra: multiple people detection and tracking using silhouettes. In *Proceedings of the International Conference on Image Analysis and Processing*, pages 280–285, 1999.

[22] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[23] A. Hochuli, L. Oliveira, A. Britto, and A. Koerich. Detection and classification of human movements in video scenes. In *Advances in Image and Video Technology*, pages 678–691. 2007.

[24] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004.

[25] Tianci Huang, Jingbang Qiu, T. Sakayori, S. Goto, and T. Ikenaga. Motion detection based on background modeling and performance analysis for outdoor surveillance. In *International Conference on Computer Modeling and Simulation*, pages 38–42, 2009.

[26] Ming-Chuan Hung and Don-Lin Yang. An efficient fuzzy C-Means clustering algorithm. In *Proceedings of IEEE International Conference on Data Mining*, pages 225–232, 2001.

[27] S. Indupalli. Background modeling for intelligent video surveillance system. Master's thesis, University of Windsor (Canada), 2006. M.Sc.

[28] S. Indupalli, M.A. Ali, and B. Boufama. A novel Clustering-Based method for adaptive background segmentation. In *The 3rd Canadian Conference on Computer and Robot Vision*, page 37, 2006.

[29] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Proceedings of the Workshop on Motion and Video Computing*, pages 22–27, 2002.

[30] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, June 2005.

[31] Taekyung Kim and Joonki Paik. Adaptive background generation for video object segmentation. In *Advances in Visual Computing*, pages 871–880. 2006.

[32] P. Kumar, K. Sengupta, and A. Lee. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. In *Proceedings of IEEE 5th International Conference on Intelligent Transportation Systems*, pages 100–105, 2002.

[33] Dar-Shyang Lee. Effective gaussian mixture learning for video background subtraction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):827–832, 2005.

[34] Boris Lenseigne, Thierry Dorval, Arnaud Ogier, and Auguste Genovesio. A new color representation for intensity independent pixel classification in confocal microscopy images. In *Advanced Concepts for Intelligent Vision Systems*, pages 597–606. 2007.

[35] Liyuan Li, Weimin Huang, Irene Yu-Hua Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[36] Liyuan Li, Weimin Huang, I.Y.H. Gu, and Qi Tian. Foreground object detection in changing background based on color co-occurrence statistics. In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, pages 269–274, 2002.

[37] Liyuan Li and M.K.H. Leung. Integrating intensity and texture differences for robust change detection. *IEEE Transactions on Image Processing*, 11(2):105–112, 2002.

[38] Qing-Zhong Li, Dong-Xiao He, and Bing Wang. Effective moving objects detection based on clustering background model for video surveillance. In *Congress on Image and Signal Processing*, volume 3, pages 656–660, 2008.

[39] A.J. Lipton and N. Haering. ComMode: an algorithm for video background modeling and object segmentation. In *7th International Conference on Control, Automation, Robotics and Vision, ICARCV*, volume 3, pages 1603–1608, 2002.

[40] W. Long and Y. H. Yang. Stationary background generation: An alternative to the difference of two images. In *Pattern Recognition*, volume 23, pages 1351–1359. 1990.

[41] Jiangung Lou, Qifeng Liu, Tieniu Tan, and Weiming Hu. Semantic interpretation of object activities in a surveillance system. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 777–780, 2002.

[42] Si Luo and Li Zhang. A background model estimation algorithm based on analysis of local motion for video surveillance. In *2005 Fifth International Conference on Information, Communications and Signal Processing*, pages 700–704, 2005.

[43] L. Maddalena and A. Petrosino. A Self-Organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[44] S. J. Mckenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. In *Image and Vision Computing*, volume 17, pages 225–231, 1999.

[45] Stephen J. McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, and Harry Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, October 2000.

[46] A. Monnet, A. Mittal, N. Paragios, and Visvanathan Ramesh. Background modeling and subtraction of dynamic scenes. In *Proceedings of Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1305–1312, 2003.

[47] S. Nasser, R. Alkhaldi, and G. Vert. A modified fuzzy k-means clustering using expectation maximization. In *IEEE International Conference on Fuzzy Systems*, pages 231–235, 2006.

[48] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[49] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, 2004.

[50] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *Image Processing, IEEE Transactions on*, 14(3):294–307, 2005.

[51] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using Kalman-Foltering. In *Proceedings of International Conference on Recent Advances in Mechatronics (ICRAM)*, pages 193–199, 1995.

[52] Dick Anthony Robert and Brooks Michael John. Issues in automated visual surveillance. In *Proceedings of the 7th Biennial Australian Pattern Recognition Society Conference*, pages 195–203, 2003.

[53] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, page 252, 1999.

[54] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[55] J. Steffens, E. Elagin, and H. Neven. PersonSpotter-fast and robust system for human detection, tracking and recognition. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 516–521, 1998.

[56] Gijs Stijnman and Rein van den Boomgaard. Background Extraction of Colour Image Sequences using a Gaussian mixture model. Technical report, ISIS technical report series, University of Amsterdam, 2000.

[57] Peng Suo and Yanjiang Wang. An improved adaptive background modeling algorithm based on gaussian mixture model. In *The 9th International Conference on Signal Processing*, pages 1436–1439, 2008.

[58] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 255–261 vol.1, 1999.

[59] Hanzi Wang and David Suter. A novel robust statistical method for background initialization and visual surveillance. In *Computer Vision ACCV 2006*, pages 328–337. 2006.

[60] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[61] L. Q. Xu, J. L. Landabaso, and B. Lei. Segmentation and tracking of multiple moving objects for intelligent video analysis. *BT Technology Journal*, 22(3):140–150, July 2004.

[62] Wei Yang, L. Rueda, and A. Ngom. A simulated annealing approach to find the optimal parameters for fuzzy clustering microarray data. In *The 25th International Conference of the Chilean Computer Science Society*, pages 45–54, 2005.

[63] Qi Zang and R. Klette. Robust background subtraction and maintenance. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 90–93, 2004.

# Vita Auctoris

Huajing Yao was born in 1983 in Jiangsu, P. R. China. She earned Bachelor of Engineering (Computer Science and Technology) and Master of Engineering (Computer Application Technology) from Hohai University in 2004 and 2007 respectively. She joined the University of Windsor in September 2007 and currently a candidate for the Master's degree in Computer Science at the University of Windsor. Her research interests include computer vision, pattern recognition and image processing.

**Publication:**

- Huajing Yao, Imran Shafiq Ahmad, "Adaptive Foreground Segmentation Using Fuzzy Approach", to appear in the *Proceedings of 4th IEEE International Conference on Digital Information Management (ICDIM'09)*, Ann Arbor, Michigan, USA, November 1 - 4, 2009.