DYNAMIC TEXTURE ANALYSIS IN VIDEO WITH APPLICATION TO FLAME, SMOKE AND VOLATILE ORGANIC COMPOUND VAPOR DETECTION

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Osman Günay

July 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. A. Enis Çetin (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Uğur Güdükbay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Onay Urfalıoğlu

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of the Institute

ABSTRACT

DYNAMIC TEXTURE ANALYSIS IN VIDEO WITH APPLICATION TO FLAME, SMOKE AND VOLATILE ORGANIC COMPOUND VAPOR DETECTION

Osman Günay
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. A. Enis Çetin
July 2009

Dynamic textures are moving image sequences that exhibit stationary characteristics in time such as fire, smoke, volatile organic compound (VOC) plumes, waves, etc. Most surveillance applications already have motion detection and recognition capability, but dynamic texture detection algorithms are not integral part of these applications. In this thesis, image processing based algorithms for detection of specific dynamic textures are developed. Our methods can be developed in practical surveillance applications to detect VOC leaks, fire and smoke. The method developed for VOC emission detection in infrared videos uses a change detection algorithm to find the rising VOC plume. The rising characteristic of the plume is detected using a hidden Markov model (HMM). The dark regions that are formed on the leaking equipment are found using a background subtraction algorithm. Another method is developed based on an active learning algorithm that is used to detect wild fires at night and close range flames. The active learning algorithm is based on the Least-Mean-Square (LMS) method. Decisions from the sub-algorithms, each of which characterize a certain property of the texture to be detected, are combined using the LMS algorithm

to reach a final decision. Another image processing method is developed to detect fire and smoke from moving camera video sequences. The global motion of the camera is compensated by finding an affine transformation between the frames using optical flow and RANSAC. Three frame change detection methods with motion compensation are used for fire detection with a moving camera. A background subtraction algorithm with global motion estimation is developed for smoke detection.

Keywords: VOC leak detection, flame detection, smoke detection, night-fire detection, computer vision, dynamic textures, hidden Markov models, least-mean-square (LMS) algorithm, active learning, optical flow, motion compensation, RANSAC.

ÖZET

VİDEODA DİNAMİK DOKU ANALİZİ VE ALEV, DUMAN, UÇUCU ORGANİK BİLEŞİK BUHARI BULMAYA UYGULANMASI

Osman Günay
Elektrik ve Elektronik Mühendisliği Bölümü, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. A. Enis Çetin
July 2009

Dinamik dokular hareketli görüntü dizilerinden oluşan, zaman içinde durgun karakteristik gösteren ateş, duman, uçucu organik bileşik (VOC) gazları, gibi maddelerdir. Hareket algılama ve tanıma, görüntü tabanlı güvenlik sistemlerinin yerleşik bir parçası haline gelmesine rağmen, dinamik dokuları algılama ve sınıflandırma henüz çoğu güvenlik sisteminde entegre edilmemiştir. Bu tezde dinamik dokuların ayrıştırılması ve sınıflandırması için resim işlemeye dayalı algoritmalar geliştirilmiştir. Geliştirdiğimiz yöntemler güvenlik sistemlerinde VOC sızıntıları, ateş ve duman algılamak için kullanılabilir. Infrared videolarda VOC sızıntılarını algılamak için geliştirilen yöntem, yükselen VOC gazlarını bulmak için bir değişiklik tespit algoritması kullanır. Gazların zaman içinde yükselme karakteristiği bir Gizli Markov Modeli (HMM) kullanılarak algılanır. Bu gazların sızıntı yapan ekipmanda oluşturduğu koyu bölgeler arkaplan çıkarma algoritması ile bulunur. Gece çıkan orman yangınlarını ve yakın mesafeli alevleri algılamak için etkin bir öğrenme algoritması kullanan bir görüntü işleme yöntemi geliştirilmiştir. Etkin öğrenme algoritması, en az ortalama kare (LMS) yöntemini kullanır. Dinamik dokuların belirli özelliklerini karakterize eden alt-yordamların

her biri nihai karara ulaşmak için LMS algoritması ile birleştirilmiştir. Başka bir görüntü işleme yöntemi de hareketli kamera kullanarak ateş ve duman algılamak için geliştirilmiştir. Kameranın çerçeveler arasında yaptığı hareket optik akış ve RANSAC kullanılarak bulunan bir ilgin dönüşün ile telafi edilmiştir. Hareketli bir kamera ile ateş bulmak için üç kare değişim algılama yöntemi kullanılır. Duman algılama için kamera hareket tahmini kullanan bir arka plan çıkarma algoritması geliştirilmiştir.

Anahtar Kelimeler: VOC algılama, alev algılama, duman algılama, gece ateş bulma, bilgisayarlı görü, dinamik doku, gizli Markov modeli, en küçük-ortalama-kare (LMS) yordamı, aktif öğrenme, optik akış, hareket telafisi, RANSAC.

ACKNOWLEDGMENTS

I would like to express my gratitude to Prof. Dr. A. Enis Çetin for his supervision, suggestions and encouragement throughout the development of his thesis.

I am also indebted to Assoc. Prof. Uğur Güdükbay and Dr. Onay Urfalıoğlu for accepting to read and review this thesis.

I would also like to thank Behçet Uğur Töreyin and Yiğithan Dedeoğlu who developed the algorithms and application that provide the basis of our work. I want to thank Dennis Akers for drawing our attention to VOC leak detection problems in video, and Assist. Prof. Dr. Mehmet Bayındır for providing IR cameras for our tests.

I wish to thank all of my friends and colleagues at our department for their collaboration and support.

I would also like to thank TÜBİTAK for providing scholarship throughout my graduate study (BIDEB-2228).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Video based surveillance systems have been used since the closed circuit television systems (CCTV) became widely available for security applications. During the recent decades powered by the developments in the computer vision technology several image processing algorithms have been developed for object recognition, classification and event analysis. Today most video based surveillance systems are already equipped with these image processing modules. Surveillance systems used in industrial plants, refineries, chemical manufacturers, etc. also need algorithms for automatic detection and classification of dynamic textures. Dynamic textures are regions of moving image frames that display some sort of "temporal stationarity". Dynamic textures include fire, smoke, waves, gas plume, etc.

There are several algorithms in the literature developed for recognition and segmentation of dynamic textures. Vidal and Ghoreyshi used second order Ising descriptors to model the spatial statistics of dynamic textures [2]. They try to solve the dynamic texture segmentation problem using this model by minimizing the temporal variance of the stochastic part of the model. This approach is proven to handle intensity and texture based image segmentation. Another

method to segment dynamic textures is to use optical flow features. The features used for this method usually describe local image distortions in terms of curl or divergence. In [3] one normal and four complete optical flow algorithms are compared in terms of performance.

Saisan et al. used stochastic methods to learn and model dynamic textures [4]. Sequences of moving images are analyzed as signals. A closed form solution to the learning problem for a second-order model is proposed based on total likelihood or prediction error criteria. They proposed a method for recognition of textures which uses the observation that similar textures tend to cluster in model space. Dynamic textures are modeled using the spatio-temporal autoregressive model (STAR) in [5]. In this model each pixel is expressed as a linear combination of surrounding pixels lagged both in space and in time. Least squares method is used to estimate model parameters for large, causal neighborhoods using a large number of parameters. There are several other dynamic texture recognition and modeling methods in the literature ([6] -[12]) that are aimed at segmenting general dynamic texture characteristics.

In this thesis, practical methods for recognizing certain types of dynamic textures are developed. These methods use computationally efficient real-time algorithms to work in a surveillance system. An image processing based algorithm is designed to detect and segment volatile organic compounds (VOC) leaking from industrial equipments using video streams captured by an infrared (IR) camera. Video frames are automatically analyzed to detect VOC plume and also the residues that the VOC leak leaves on the leaking equipment. VOC plumes are detected and tracked using a change detection method and then darker regions that are formed on the leaking equipment are found using a background subtraction algorithm.

An active learning method is used for dynamic texture recognition. The algorithm combines the decision results of sub-algorithms that characterize a different aspect of the analyzed texture. Individual decisions of the sub-algorithms are combined together using a least-mean-square (LMS) based decision fusion approach, and texture/no-texture decision is reached by an active learning method. This method is also applied to wildfire detection [13]. We present the results of the application of the algorithm to wildfire detection at night and close range flame detection. The method for wildfire detection at night comprises three sub-algorithms: (i) slow moving video object detection, (ii) bright region detection, and (iii) detection of objects exhibiting periodic motion. Each of these sub-algorithms characterizes an aspect of fire captured at night by a visible range PTZ camera. For flame detection four sub-algorithms are developed: (i) detection of flame colored moving objects, (ii) temporal, and (iii) spatial wavelet analysis for flicker detection and (iv) contour analysis of fire colored region boundaries. Each algorithm yields a continuous decision value as a real number in the range [-1,1] at every image frame of a video sequence. Decision values from sub-algorithms are fused using an adaptive algorithm in which weights are updated using the Least Mean Square (LMS) method in the training (learning) stage.

An optical flow based algorithm is developed to segment moving objects from a continuously moving camera. The algorithm is applied to flame detection and wildfire smoke detection from a panning camera. Two different methods are developed for texture segmentation from a moving camera. The first method uses three consecutive frames to register camera motion between the reference middle frame and the other two frames. After motion estimation the previous and next frames are warped into the reference frame using the estimated affine transformation. This approach can be used for segmenting flames from moving camera video sequences. To segment smoke from a moving camera we use a background subtraction method similar to panoramic backgrounds usually employed for using background subtraction with moving cameras.

1.1 Thesis Outline

The outline of the thesis is as follows. In Chapter 2, a method for detecting VOC plumes in video is developed. An LMS based active learning algorithm is used for wild fire detection at night and close range flame detection in Chapter 3. In Chapter 4, a method for segmenting fire and smoke from moving camera image sequences is developed. Chapter 5 concludes the thesis by providing an overall summary of the results.

Chapter 2

VOC Leak Detection in IR Videos

In this chapter, an image processing based algorithm is designed to detect and segment volatile organic compounds (VOC) leaking from industrial equipments using video streams captured by an infrared (IR) camera. In the U.S, the leak detection and repair program (LDAR) requires petroleum refineries and organic chemical manufacturers to check for possible VOC leaks around process equipments and perform repairs if necessary since 1980s [14]. In the current detection framework a portable flame ionization detector (FID) is used to monitor the seals around the components for VOC leaks. Since there are a large number of process components in a single facility this method is quite costly to perform even if it is done only four times a year [15]. In recent years, petroleum refining and petrochemical industries started using IR cameras to detect volatile organic compounds (VOC) leaking out of process equipment. This method is a low cost alternative to the FID procedure [16, 17]. The IR cameras work at a predetermined wavelength that absorbs VOC leaks.

Different substances create VOC emissions with different characteristics. Diesel and propane have vapor similar to smoke coming out of a pile of burning wood while gasoline vapor is transparent and wavy [18]. The common characteristic of all VOC emissions is the fact that during the initial stages of VOC leaks the temperature of the leaking equipments and the leak drops which causes temperature difference between the surrounding air and the leak. The temperature difference causes intensity difference on the image that is created by the detector array of the camera [15]. Another common characteristic of the VOC plumes is the absorption of IR light at a specific wavelength. Different VOC plumes have different absorption characteristics. The IR spectra of Ammonia, Methane, Butane and Propane are shown in Fig. 2.1. The absorption amount of sources is shown against the wavelength of the IR light. The FLIR camera we used has spectral range in $7.5\mu m$ to $13\mu m$. Therefore methane and ammonia can easily be recognized as dark regions in the IR videos obtained with this camera. In our approach we first detect and track VOC plumes using a change detection method and then find the darker regions that are formed on the leaking equipment using a background subtraction based algorithm [19].

2.1 Related Work

There are only a few implementations that use image processing for detection of VOC emissions. Most methods use sensor arrays or FLIR and LWIR cameras for detection. The method in [20] uses piezoelectric acoustic wave sensors for the detection of VOCs. When a molecular material is added or subtracted from the surface of the acoustic wave sensor, a change in its resonant frequency occurs. Specifically they use 6 quartz crystal microbalances (QCMs) as a sensor array and the responses of the sensors are analyzed to separate different patterns of VOCs. Detection of VOCs is performed with polymer-coated cantilevers in [21]. The change in the resonance frequency of the cantilevers when they absorb VOCs

is used as detection mechanism. The system in [22] uses a tin oxide gas-sensors array and artificial neural network (ANN) for the identification of some of the volatile organic compounds relevant to environmental monitoring. An array of SnO_2-based thick-film gas sensors is used to generate the response patterns and back propagation neural network is used for the classification. In [15] an FFT based image alignment method is used to register IR video frames as a preprocessing module for other possible video processing methods.

Our method is similar to the method described in [18] that uses only image processing techniques to detect VOC emissions. In [18], videos captured by a visible-range camera are used to detect leaking VOC plume from a damaged component using the observation that edges present in image frames lose their sharpness around the regions where VOC emissions occur. In this method the decrease of high frequency energy of the scene is detected using the one level wavelet transform of the current and the background images which are obtained using a background subtraction algorithm. The regions with VOC plume in image frames are analyzed in low-band sub-images, as well. Image frames are compared with their corresponding low-band images for the decrease in wavelet energy. One drawback of this implementation is the use of a visible-range camera, by using an infrared camera we can better monitor the temperature difference that is caused by VOC plume during the initial stages of emission.

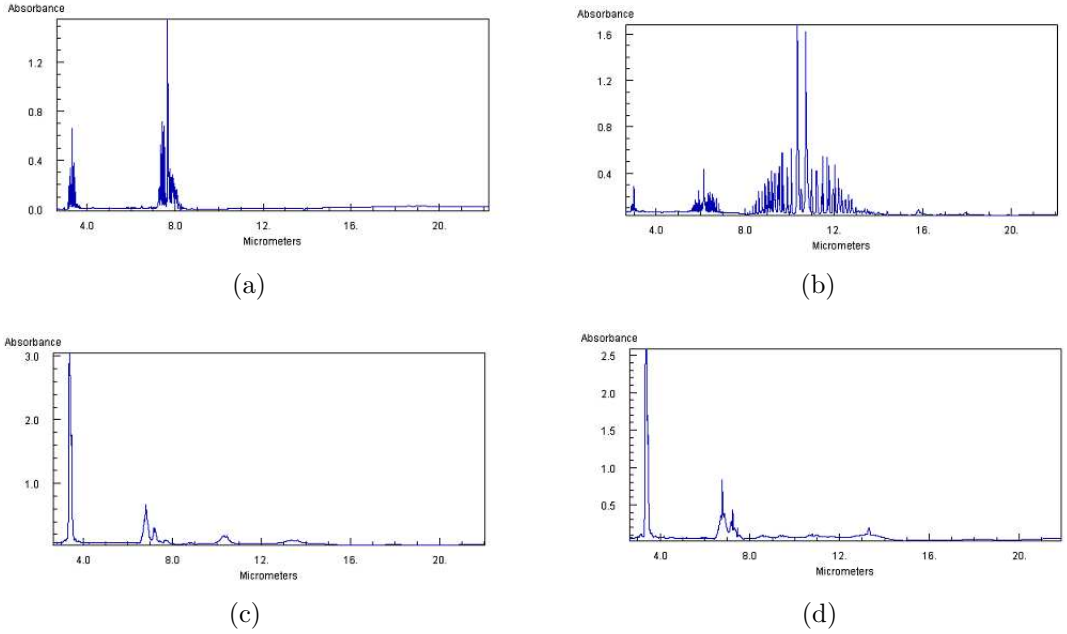


Figure 2.1: Infrared spectrum of various VOC sources: (a) Methane IR spectrum; (b) Ammonia IR spectrum; (c) Butane IR spectrum; (d) Propane IR spectrum [1].

2.2 Detection Algorithm

2.2.1 Detection of Slow Moving Objects

When process equipments in petroleum or organic chemical factories leak VOC plume the plume has lower temperature than the surrounding air during the initial stages of emission or the vapor absorbs the IR light and the region appears darker then the background. This also lowers the temperature on the part of the leaking equipment where the leak occurs. Therefore these regions become darker in IR camera images. To detect these regions we use a background subtraction algorithm that uses double backgrounds for finding left or removed objects that are stationary but have different characteristics than the background. Let $I(x, n)$ represent the intensity value of the pixel at location x in the n^{th} video frame. Assuming the camera is fixed, two background images, $B^{fast}(x, n)$ and $B^{slow}(x, n)$

corresponding to the scene with different update rates are estimated [23, 24], from the video images $I(x,n)$. Initially, $B^{fast}(x,0)$ and $B^{slow}(x,0)$ can be taken as $I(x,0)$.

In [19] a background image $B(x,n+1)$ at time instant $n+1$ is recursively estimated from the image frame $I(x,n)$ and the background image $B(x,n)$ of the video as follows:

$$B(x,n+1) = \begin{cases} aB(x,n) + (1-a)I(x,n) & \text{if } x \text{ is stationary} \\ B(x,n) & \text{if } x \text{ is a moving pixel} \end{cases}, \qquad (2.1)$$

where the time constant a is a parameter between 0 and 1 that determines how fast the new information in the current image $I(x,n)$ supplants old observations. The image $B(x,n)$ models the background scene.

Stationary and moving pixel definitions are given in [19]. Background images $B^{fast}(x,n)$ and $B^{slow}(x,n)$ are updated as in Eq. (2.1) with different update rates. In our implementation, $B^{fast}(x,n)$ is updated at every frame and $B^{slow}(x,n)$ is updated once in a second with $a = 0.7$ and 0.9, respectively. The update parameter of $B^{fast}(x,n)$ is chosen smaller than $B^{slow}(x,n)$ because we want more contribution from the current image $I(x,n)$ in the next background image $B^{fast}(x,n+1)$

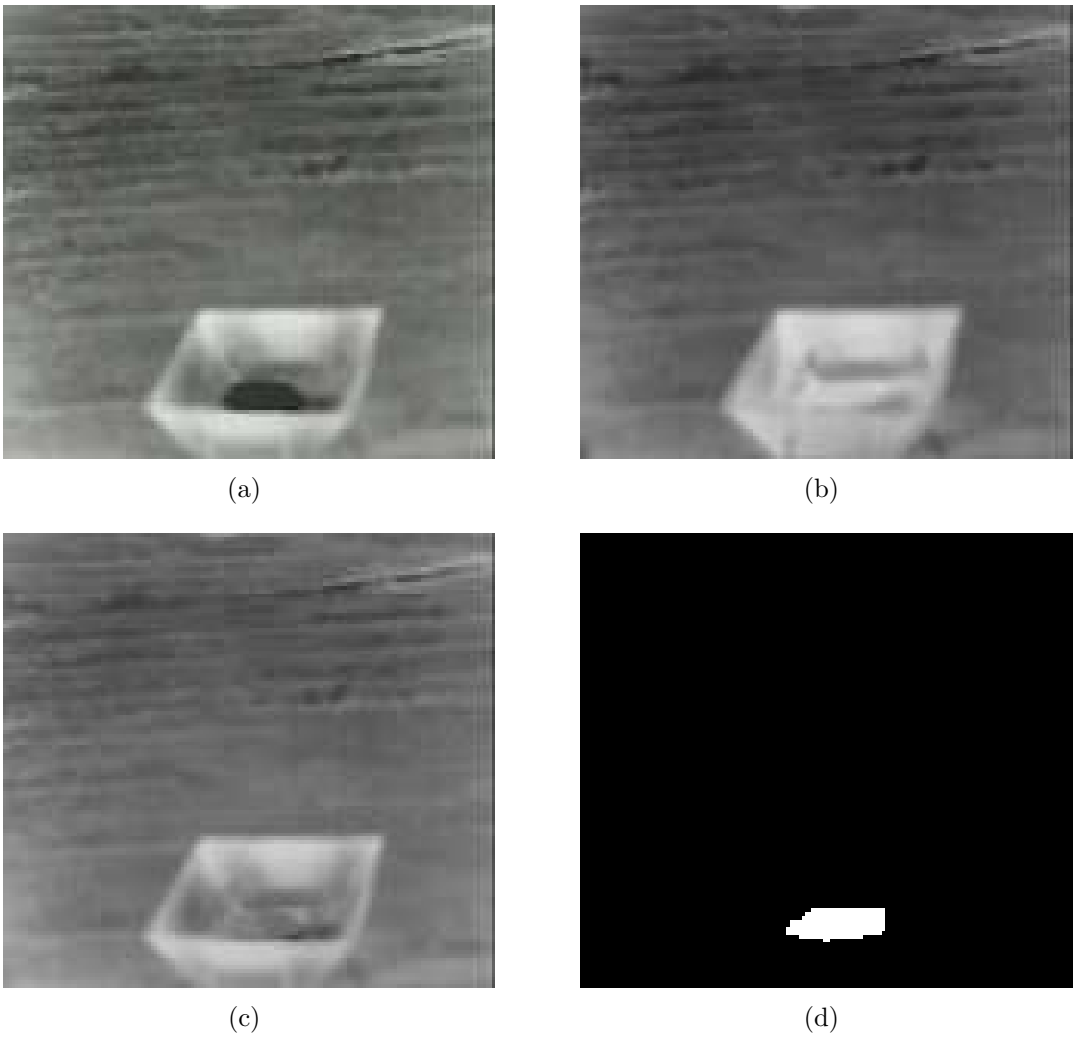
By comparing background images, B^{fast} and B^{slow} slow moving objects are detected [23, 24, 25] because B^{fast} is updated more often than B^{slow}. If there exists a substantial difference between the two images for some period of time, then an alarm for slow moving region is raised, and the region is marked. Apart from the slow moving object constraint we also impose intensity conditions on the detected regions considering the temperature drop on caused VOC emission

as follows:

$$D(x,n) = \begin{cases} 1 & \text{if } (B^{fast}(x,n) - I(x,n)) > T_C \text{ and } I(x,n) < T_I \\ 0 & \text{else} \end{cases}, \qquad (2.2)$$

where $D(x,n)$ is a binary image that has value 1 for pixels that satisfy intensity requirements and 0 for others. T_C and T_I are experimentally determined thresholds.

In Fig. 2.2 the foreground, slow and fast backgrounds and the detection result for a frame of an IR video sequence are shown.



(a) (b) (c) (d)

Figure 2.2: Foreground and background images of a frame of a video sequence: (a) current image; (b) slow background; (c) fast background; (d) detection result.

2.2.2 Detection of VOC Plume

After detecting the color change on the leaking equipment we also detect the moving VOC plume. VOC plume is fast moving and nearly transparent for most leaking gases. Thus plume cannot be detected using background subtraction. A change detection approach is used to detect VOC plume and background subtraction is used to segment and discard ordinary moving objects. Moving objects are detected by subtracting the current image $I(x,n)$ from the background and thresholding by an adaptively updated threshold image $T(x,n)$ as shown in Eq. 2.3. $M(x,n)$ is a binary image that has high values for moving regions in the current frame:

$$M(x,n) = \begin{cases} 1 & \text{if } |B(x,n) - I(x,n)| > T(x,n) \\ 0 & \text{else} \end{cases} \tag{2.3}$$

$T(x,n)$ is a recursively updated threshold at each frame n, describing an intensity change at pixel position x:

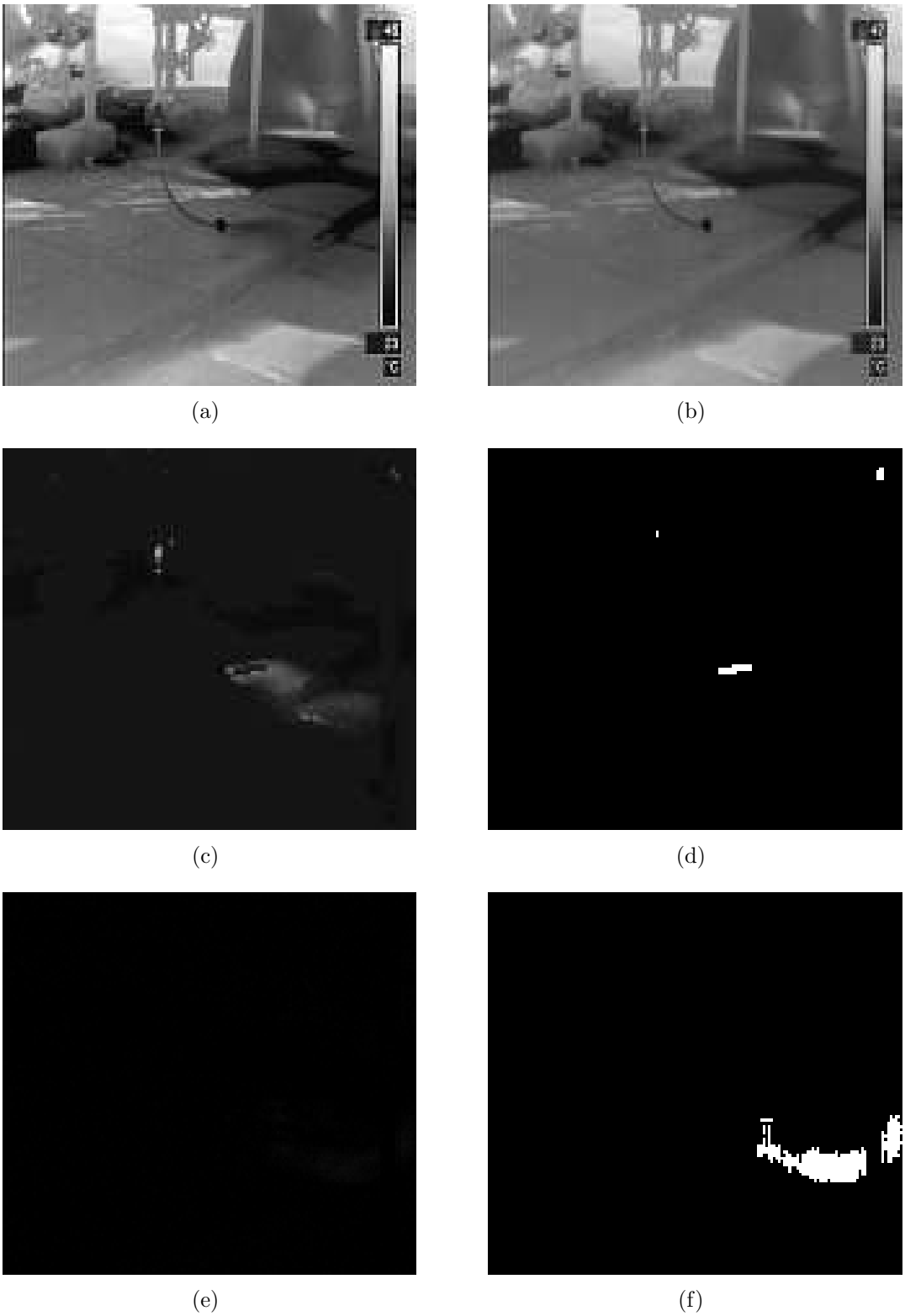
$$T(x,n{+}1) = \begin{cases} aT(x,n) + (1-a)(c|I(x,n) - B(x,n)|) & \text{if } x \text{ is stationary} \\ T(x,n) & \text{if } x \text{ is a moving pixel} \end{cases}, \tag{2.4}$$

where c is a real number greater than one and the update parameter a is a positive number close to one.

Possible VOC plume regions are found by thresholding the difference between the current frame, $I(x,n)$, and previous frame $I(x,n-1)$ and discarding the results of background subtraction as follows:

$$C(x,n) = \begin{cases} 1 & \text{if } T_L < |I(x,n) - I(x,n-1)| < T_H \text{ and } M(x,n) < 1 \\ 0 & \text{else} \end{cases}, \tag{2.5}$$

where $C(x, n)$ is a binary image that has value 1 for possible plume regions. Fig. 2.3 shows the application of the algorithm for detecting butane plume.



(a) (b) (c) (d) (e) (f)

Figure 2.3: VOC plume segmentation using change detection: (a) current frame; (b) background; (c) threshold; (d) background motion; (e) frame difference; (f) detection result.

2.2.3 Rising Plume Detection

VOC plume regions tend to rise up from the equipment at the early stages of emission. This characteristic behavior of plumes is modeled with three-state hidden Markov models (HMM). Temporal variation in row number of the center pixel belonging to a VOC plume regions found by change detection is used as a one dimensional (1-D) feature signal, $F = f(n)$, and fed to the Markov models shown in Fig. 2.4. One of the models (λ_1) corresponds to genuine VOC smoke regions and the other one (λ_2) corresponds to regions with other moving objects. Transition probabilities of these models are estimated off-line from actual VOC leaks and test smokes. The state S1 is attained, if the row value of the center pixel in the current image frame is smaller than that of the previous frame (rise-up). If the row value of the center pixel in the current image frame is larger than that of the previous frame, then S2 is attained and this means that the region moves-down. No change in the row value corresponds to S3 [13].

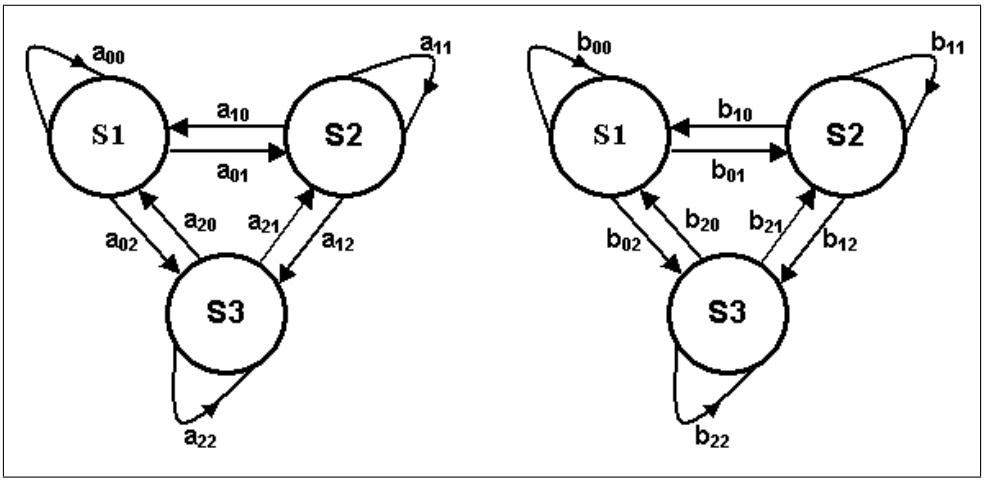


Figure 2.4: Markov model λ_1 corresponding to VOC plume (left) and the Markov model λ_2 of ordinary moving objects (right). Transition probabilities a_{ij} and b_{ij} are estimated off-line.

A possible plume region is classified as a rising region when the probability of obtaining the observed feature signal $F = f(n)$ given the probability model λ_1 is greater than the probability of obtaining the observed feature signal $F = f(n)$

given the probability model λ_2, i.e., when the center pixel belonging to a slow moving region tends to exhibit a rising characteristic [13]:

$$p_1 \;=\; P(F|\lambda_1) \;>\; p_2 \;=\; P(F|\lambda_2), \tag{2.6}$$

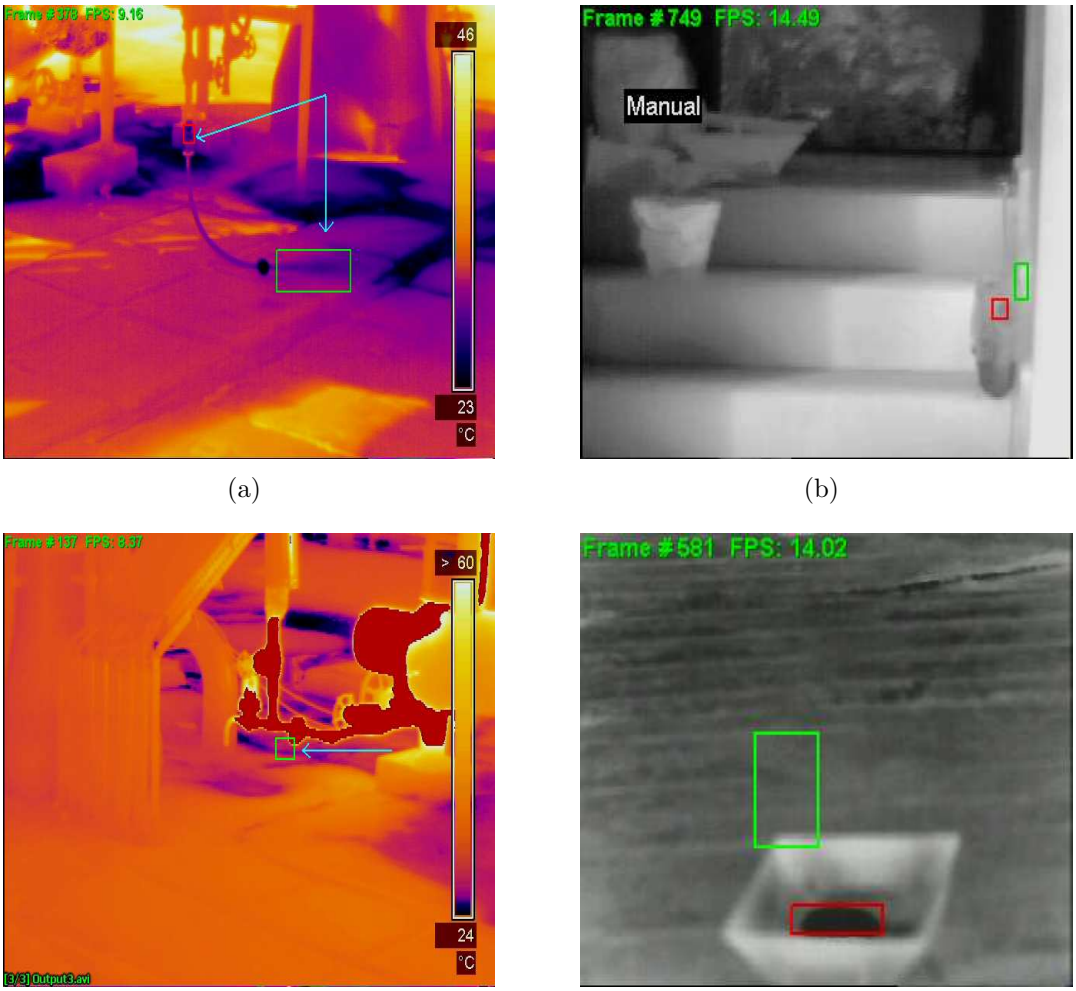
where F is the observed feature signal, λ_1 and λ_2 represent the Markov models for VOC plume and other objects respectively.

2.3 Experimental Results

The proposed method was implemented in C++ programming language and tested with various VOC plume types. The HMMs used in the temporal analysis step were trained using indoor and outdoor IR video clips with VOC emissions and ordinary moving objects. Some of the video clips are recorded at TÜPRAŞ (Türkiye Petrol Rafinerileri A.Ş.). We used a total of 8 video clips with total 7000 frames. The FLIR camera used to record some of the videos is Thermovision A40 shown in Fig. 2.5. Image frames with detection results from some of the clips are shown in Fig. 2.6. The green rectangles show the moving VOC plume and the blue rectangles indicate a temprerature change in the equipment.

Figure 2.5: Thermovison A40 FLIR camera.

Table 2.1 summarizes the detection results for IR videos with different types of VOC emissions. The first comparison is made between the total number of frames in the video with VOC plume and the number of frames detected by the algorithm. the algorithm was able to detect and track most of the plume between the frames. A second comparison is carried out between the frame number of the first image that the temperature change on the leaking equipment became visible and the frame number of the image for which the first alarm is issued by the algorithm. According to the results of the experiment, the method was able to detect temperature change at most after 75 frames.



(a) (b)

(c) (d)

Figure 2.6: Detection Results for different VOC emissions from various sources: (a) Butane; (b) Gasoline; (c) Water Vapour; (d) Ammonia.

Table 2.1: Detection results for various VOC types. Number of frames with VOC plume and the number of frames detected by the algorithm are compared. The frame number when the first temperature change in the equipment occurs and the frame number of the first detection of temperature change are displayed.

VOC Type	# of Frames with Plume	First Frame of Temp. change	# of detected frames	First Alarm Frame for Temp. Change
Ammonia	-	100	-	122
Gasoline	17	780	11	830
Butane+Propane	280	530	196	605
Water Vapour	870	-	500	-
Ethylene	170	-	110	-
Ammonia	-	300	-	310
Ammonia	-	273	-	294
Ammonia	-	365	-	380

2.4 Summary

A novel method to detect VOC emissions in IR videos is developed. The algorithm detects both moving VOC plume and the temperature change on the leaking equipment. The algorithm uses a background subtraction that uses double backgrounds to detect slow moving or stationary objects. Intensity restrictions are defined to further analyze the temperature change that occurs during the initial stages of emission. Moving VOC plume is detected using a change detection approach. Hidden Markov models are trained offline using temporal frame information to detect the rising nature of VOC plume. The method can be used for both indoor and outdoor VOC emission detection applications.

Chapter 3

Fire Detection Using LMS Based Active Learning

In this chapter, an active learning method for dynamic texture recognition is described. The algorithm combines the decision results of sub-algorithms that characterize a different aspect of the analyzed texture. Individual decisions of the sub-algorithms are combined together using a least-mean-square (LMS) based decision fusion approach, and texture/no-texture decision is reached by an active learning method.

We present the results of the application of the algorithm to wildfire detection at night and close range flame detection. The method for wildfire detection at night comprises three sub-algorithms: (i) slow moving video object detection, (ii) bright region detection, and (iii) detection of objects exhibiting periodic motion. Each of these sub-algorithms characterizes an aspect of fire captured at night by a visible range PTZ camera. In our system, we detect smoke during day-time and switch to the night-fire detection mode at night. Because smoke becomes visible much earlier than flames in Mediterranean Region. In Fig. 3.1, a day-time wildfire at an initial stage is shown. This fire was detected by our

system in the summer of 2009. A snapshot of a typical night fire smoke captured by a look-out tower camera from a distance of 3 km is shown in Fig. 3.2. Even the flame flicker is not visible from long distances. Therefore, one cannot use the flame flicker information in [26] for long distance night-fire detection.

For the flame detection problem, four sub-algorithms are used (i) detection of flame colored moving objects, (ii) temporal, and (iii) spatial wavelet analysis for flicker detection and (iv) contour analysis of fire colored region boundaries. Each algorithm yields a continuous decision value as a real number in the range [-1,1] at every image frame of a video sequence.

Decision values from sub-algorithms are fused using an adaptive algorithm in which weights are updated using the Least Mean Square (LMS) method in the training (learning) stage.

3.1 Related Work on Active Learning

The active learning method used in this thesis is similar to classifier ensembles used in pattern recognition, in which decisions from different classifiers are combined using a linear combiner [27]. A multiple classifier system can prove useful for difficult pattern recognition problems especially when large class sets and noisy data are involved, because it allows the use of arbitrary feature descriptors and classification procedures at the same time [28]. The dynamic texture recognition problem can also be formulated as a joint application of multiple classifier decisions.

The studies in the field of collective recognition, which were started in the middle of the 1950s, found wide application in practice during the last decade, leading to solution to complex large-scale applied problems [29]. One of the first examples of the use multiple classifiers was given by Dasarathy in [27] in which

he introduced the concept of composite classifier systems as a means of achieving improved recognition system performance compared to employing the classifier components individually. The method is illustrated by studying the case of the linear/NN(Nearest Neighbor) classifier composite system. Kumar and Zhang used multiple classifiers for palmprint recognition by characterizing the user's identity through the simultaneous use of three major palmprint representations and achieve better performance than either one individually [30]. A multiple classifier fusion algorithm is proposed for developing an effective video-based face recognition method [31]. Garcia and Puig present results showing that pixel-based texture classification can be significantly improved by integrating texture methods from multiple families, each evaluated over multisized windows [32]. The proposed technique consists of an initial training stage that evaluates the behavior of each considered texture method when applied to the given texture patterns of interest over various evaluation windows of different size.

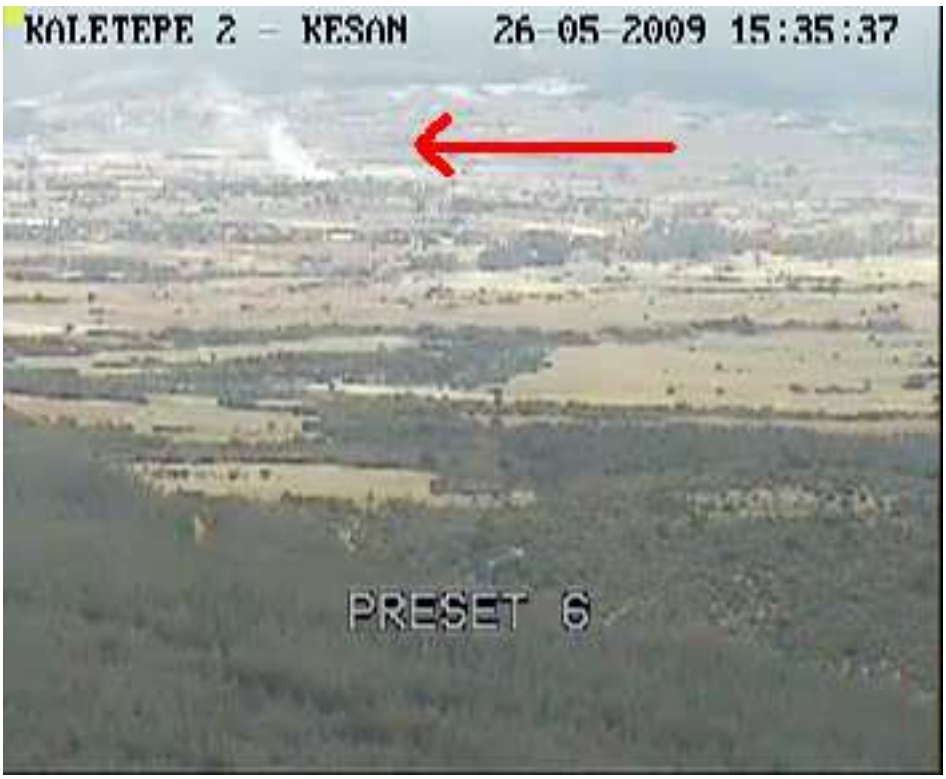


Figure 3.1: A snapshot of a typical forest fire smoke at the initial stages captured by a forest watch tower which is 5 km away from the fire (fire region is marked with an arrow).

Figure 3.2: A snapshot of a typical night fire captured by a forest watch tower which is 3 km away from the fire (fire region is marked with an arrow).

3.2 Related Work on Fire Detection

There are several publications on computer vision based fire detection ([33] - [41]). Most fire and flame detection algorithms are based on color and motion analysis in video. However, all of these algorithms focus on either day-time flame detection or smoke detection. Fires occurring at night and at long distances from the camera have different temporal and spatial characteristics than daytime fires, as shown in Figs. 3.1 and 3.2. This makes it necessary to develop explicit methods for video based fire detection at night.

The proposed automatic video based night-time fire detection algorithm is based on four sub-algorithms: (i) slow moving video object detection, (ii) bright region detection, and (iii) detection of objects exhibiting periodic motion. Each sub-algorithm separately decides on the existence of fire in the viewing range of the camera. Decisions from sub-algorithms are linearly combined using an adaptive active fusion method. Initial weights of the sub-algorithms are determined

from actual forest fire videos and test fires. They are updated using the Least-Mean-Square (LMS) algorithm during initial installation [42]. The error function in the LMS adaptation is defined as the difference between the overall decision of the compound algorithm and the decision of an oracle. In our case, the oracle is the security guard in the forest watch tower. The system asks the guard to verify its decision whenever an alarm occurs. In this way, the user actively participates in the learning process.

The active learning method based on LMS algorithm is a also applied to close range flame detection in visible range video. The sub-algorithms are modified versions of the some of the previous works [40, 26, 38, 39] which include fire detection algorithms that use temporal and spatial wavelet analysis of the video in a Hidden Markov Models framework to determine the existence of fire. In this paper, we use an LMS based on-line learning algorithm to combine the decisions of sub-algorithms, obtained using wavelet analysis and Markov models, in an efficient manner.

Moving objects are determined using a background subtraction algorithm for flame detection and fire colored moving objects are determined using Hidden Markov Models. Temporal and spatial wavelet analysis are carried out on flame boundaries and inside the fire region. An increase in energy of wavelet coefficients indicate an increase in high frequency activity. Contours of moving objects are also analyzed by estimating the boundaries of moving fire colored regions in each image frame. This spatial domain clue is also combined with temporal clues to reach a final decision. The proposed automatic video based fire detection algorithm is based on four sub-algorithms: (i) detection of fire colored moving objects, (ii) temporal and (iii) spatial wavelet analysis for flicker detection, and (iv) contour analysis of flame boundaries. Each sub-algorithm separately decides on the existence of fire in the viewing range of the camera.

3.3 Adaptation of Sub-algorithm Weights

In the proposed method each sub-algorithm has its own decision function. Decision values from sub-algorithms are linearly combined and weights of sub-algorithms are adaptively updated. Sub-algorithm weights are updated according to the Least-Mean-Square (LMS) algorithm which is the most widely used adaptive filtering method [43, 44]. The individual decision algorithms do not produce binary values 1 (correct) or -1 (false), but they produce a zero-mean real number. If the number is positive (negative), then the individual algorithm decides that there is (not) fire in the viewing range of the camera. The higher the absolute value, the more confident the sub-algorithm.

Let the compound algorithm be composed of M-many detection algorithms: $D_1, ..., D_M$. Upon receiving a sample input x, each algorithm yields a zero-mean decision value $D_i(x) \in \mathbb{R}$. The type of the sample input x may vary depending on the algorithm. It may be an individual pixel, or an image region, or the entire image depending on the sub-algorithm of the computer vision problem.

Let $\mathbf{D}(x,n) = [D_1(x,n)...D_M(x,n)]^T$, be the vector of confidence values of the sub-algorithms for the pixel at location x of input image frame at time step n, and $\mathbf{w}(n) = [w_1(n)...w_M(n)]^T$ be the current weight vector.

We define

$$\hat{y}(x,n) = \mathbf{D^T}(x,n)\mathbf{w}(n) = \sum_i w_i(n)D_i(x,n) \tag{3.1}$$

as an estimate of the correct classification result $y(x,n)$ of the oracle for the pixel at location x of input image frame at time step n, and the error $e(x,n)$ as $e(x,n) = y(x,n) - \hat{y}(x,n)$. Weights are updated by minimizing the mean-square-error (MSE):

$$\min_{w_i} E[(y(x,n) - \hat{y}(x,n))^2], \quad i = 1, ..., M, \tag{3.2}$$

where E represents the expectation operator. After solving the MSE problem the following normalized weight update equation is obtained:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(x,n)}{||\mathbf{D}(x,n)||^2} \mathbf{D}(x,n), \tag{3.3}$$

where the μ is an update parameter in the range $0 < \mu < 2$. Initially the weights can be selected as $\frac{1}{M}$. The adaptive algorithm converges, if $y(x,n)$ and $D_i(x,n)$ are wide-sense stationary random processes and when the update parameter μ lies between 0 and 2 [45, 43, 46, 13]. Eq. (3.3) is a computable weight-update equation. Whenever the oracle provides a decision, the error $e(x,n)$ is computed and the weights are updated according to Eq. (3.3).

The sub-algorithms described in the previous section are devised in such a way that each of them yields non-negative decision values, D_i's, for pixels inside fire regions. The final decision which is nothing but the weighted sum of individual decisions must also take a non-negative value when the decision functions yield non-negative values. This implies that, in the weight update step of the active decision fusion method, weights, $w(n) \geq 0$, should also be non-negative. In the proposed method, the weights are updated according to Eq. (3.3) and negative weights are reset to zero complying with the non-negative weight constraint.

3.4 Application to Wild Fire Detection at Night

3.4.1 Building Blocks of Fire Detection Algorithm

Fire detection algorithm is developed to detect the existence of fire within the viewing range of visible range camera monitoring forestal areas at night. The proposed fire detection algorithm consists of three main sub-algorithms: (i) slow moving object detection in video, (ii) bright region detection, and (iii) detection

of objects exhibiting periodic motion, with decision functions, $D_1(x,n)$, $D_2(x,n)$ and, $D_3(x,n)$ respectively, for each pixel at location x of every incoming image frame at time step n.

The decision functions $D_i, \quad i = 1, ..., M$ of sub-algorithms either produce binary values 1 (correct) or -1 (false), or zero-mean real numbers for each incoming sample x. If the number is positive (negative), then the individual algorithm decides that there is (not) fire in the viewing range of the camera. Output values of decision functions express the confidence level of each sub-algorithm. Higher the value, the more confident the algorithm.

Detection of Slow Moving Objects

The slow moving object detection algorithm used in Section 2.2.1 is also used here. When a fire starts at night it appears as a bright spot in the current image $I(x,n)$ and it can be detected by comparing the current image with the background image. However, one can also detect headlights of a vehicle or someone turning the lights of a building, etc. because they also appear as bright spots in the current image. On the other hand we can distinguish night fire from headlights by using two background images with different update rates. Contribution of headlights of vehicles into the background image $B^{fast}(x,n)$ will not be high but the night fire will appear in $B^{fast}(x,n)$ over time. $B^{slow}(x,n)$ is updated once a second therefore contribution of the night fire will be slower in this image.

The update parameter of $B^{fast}(x,n)$ is chosen smaller than $B^{slow}(x,n)$ because we want more contribution from the current image $I(x,n)$ in the next background image $B^{fast}(x,n+1)$By comparing background images, B^{fast} and B^{slow} slow moving objects are detected because B^{fast} is updated more often than B^{slow} [23, 24, 25]. If there exists a substantial difference between the two images for some period of time, then an alarm for slow moving region is raised, and

the region is marked. The decision value indicating the confidence level of the first sub-algorithm is determined by the difference between background images. Decision function $D_1(x,n)$ is defined as:

$$D_1(x,n) = \begin{cases} -1 & \text{if } |B^{fast}(x,n) - B^{slow}(x,n)| \leq T_{low} \\ 2\frac{|B^{fast}(x,n)-B^{slow}(x,n)|-T_{low}}{T_{high}-T_{low}} - 1 & \text{if } T_{low} \leq |B^{fast}(x,n) - B^{slow}(x,n)| \leq T_{high} \\ 1 & \text{if } T_{high} \leq |B^{fast}(x,n) - B^{slow}(x,n)| \end{cases} , \quad (3.4)$$

where $0 < T_{low} < T_{high}$ are experimentally determined threshold values [13]. The threshold T_{low} is simply determined according to the noise level of the camera. When the pixel value difference is less than $T_{low} = 10$ we assume that this difference is due to noise(pixel values are between 0 and 255 in 8-bit grayscale images) and the decision function takes the value $D_1(x,n) = -1$ when the difference between the pixel values at location x of the image increases the value of the decision function increases as well. When the difference exceeds $T_{high} = 30$, we are sure that there is a difference between two images and the decision function $D_1(x,n) = 1$. On the average, $30/(255 \div 2)$ corresponds to %25 difference between the two pixels.

In our implementation, T_{low} (T_{high}) is taken as 10 (30) on the luminance (Y) component of video images. The decision function is not sensitive to the threshold value T_{high} because night fire appears as a bright spot in a dark background. In all the test sequences that contain wild fire the decision function takes the value 1.

Confidence value is 1 (-1), if the difference $|B^{fast}(x,n) - B^{slow}(x,n)|$ is higher (lower) than threshold T_{high} (T_{low}). The decision function $D_1(x,n)$ takes real values in the range [-1,1] if the difference is in between the two threshold values.

Detection of Bright Regions

In this sub-algorithm, image intensity analysis is carried out on slow moving objects to detect bright regions. Long distance wild fires detected at night appear

as bright regions and do not carry much color information. Commercial visible range PTZ cameras that we used cannot capture color information from miles away at night as shown in Fig. 3.2. Therefore it is difficult to implement fire detection methods that depend on RGB information. Confidence value corresponding to this sub-algorithm should account for these characteristics.

The decision function for this sub-algorithm $D_2(x,n)$ takes values between 1 and -1 depending on the value of the $Y(x,n)$ component of the YUV color space. The decision function $D_2(x,n)$ is defined as:

$$D_2(x,n) = \begin{cases} 1 - \frac{255-Y(x,n)}{128}, & \text{if } Y(x,n) > T_I \\ -1, & \text{otherwise} \end{cases}, \tag{3.5}$$

where $Y(x,n)$ is the luminance value of the pixel at location x of the input image frame at time step n. The luminance component Y takes real values in the range [0,255] in an image. The threshold T_I is an experimentally determined value and taken as 180 on the luminance (Y) component. The luminance value exceeded T_I=180 in all test fires we carried out. The confidence value of $D_2(x,n)$ is -1 if $Y(x,n)$ is below T_I. The decision value approaches 1 as luminance value increases and drops down to -1 for pixels with low luminance values.

Our system is developed for Mediterranean area and in this area the weather is clear and humidity is low in summer season when most of the wild fires occur. It is very unlikely that a wildfire will start in a humid day [47]. Our test videos are captured in a clear day with low humidity level.

Detection of Periodic Regions

The main sources of false alarms in a fire detection scenario at night conditions are flashing lights on vehicles and building lights in residential areas. Most of these light sources exhibit perfect periodic behavior which can be detected using

frequency based analysis techniques. The removal of objects exhibiting periodic motion eliminates some of the false alarms caused by artificial light sources. The decision function for this sub-algorithm $D_3(x,n)$ is used to remove periodic objects from candidate fire regions. The candidate regions are determined by thresholding the previous two decision functions $D_1(x,n)$ and $D_2(x,n)$ as follows:

$$A(x,n) = \begin{cases} 1, & \text{if } D_1(x,n) > T_{D_1} \text{ and } D_2(x,n) > T_{D_2} \\ 0, & \text{otherwise} \end{cases} , \tag{3.6}$$

where the T_{D1} and T_{D2} are experimentally determined thresholds and $A(x,n)$ is a binary image having value 1 for pixels corresponding to candidate regions and 0 for others. The candidate pixels are grouped into connected regions and labeled by a two-level connected component labeling algorithm [48]. The movement of the labeled regions between frames is also observed using an object tracking algorithm [25]. The mean intensity values of tracked regions are stored for 50 consecutive frames corresponding to 2 sec of video captured at 25 fps. The resulting sequence of mean values is used to decide the periodicity of the region. Average magnitude difference function (AMDF) methods are used for detection of objects exhibiting periodic motion.

AMDF is generally used to detect pitch period of voiced speech signals [49]. For a given sequence of numbers $s[n]$, AMDF is calculated as follows:

$$P(l) = \sum_{n=1}^{N-l+1} |s[n+l-1] - s[n]|, \quad l = 1, 2, ..., N \ , \tag{3.7}$$

where N is the number of samples in $s[n]$.

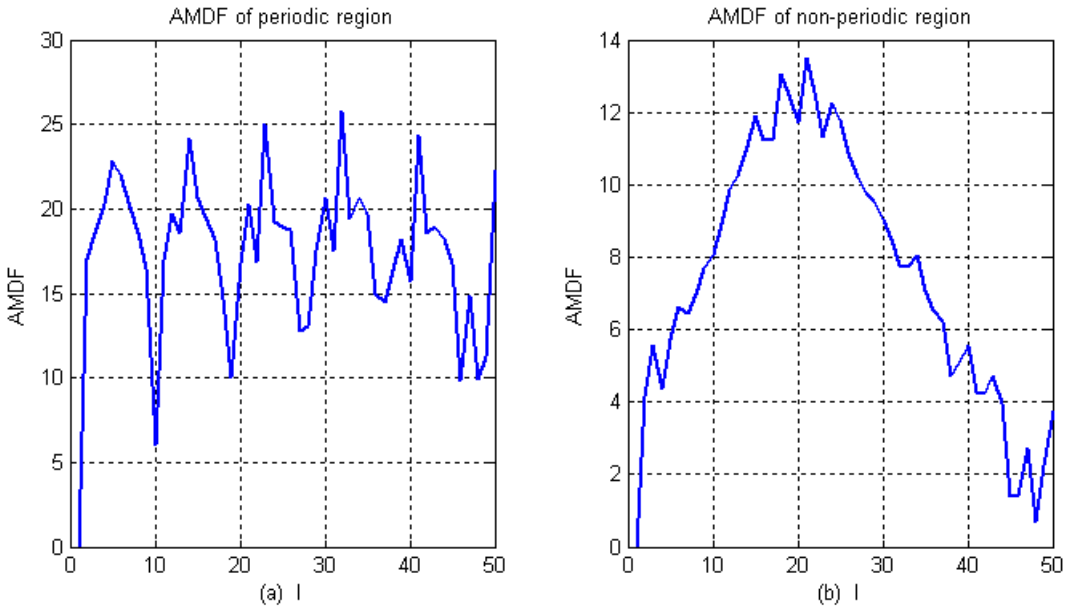


Figure 3.3: AMDF graphs for (a) periodic flashing light and (b) non-periodic bright region in video.

In Eq. 3.7, $s[n]$ represents the intensity value of each candidate region . N is selected as 50 in 25 fps video. For periodic regions, the graph of AMDF also shows a periodic character as shown in Fig. 3.5. If the AMDF of $s[n]$ is periodic we define P_{AMDF}=1, otherwise we set P_{AMDF}=-1.

The decision function for the third sub-algorithm is determined in the following manner:

$$D_3(x,n) = \begin{cases} 1, & P_{AMDF} = 1 \\ -1, & \text{otherwise} \end{cases} \tag{3.8}$$

3.4.2 Experimental Results

The proposed fire detection scheme with LMS based active learning method is implemented in C++ programming language and tested with forest surveillance recordings captured from cameras mounted on top of forest watch towers near Antalya and Mugla regions in Turkey. For detection tests we used an analog PTZ camera and an IP PTZ camera. The analog camera we used is Samsung

SCC-641P. The camera supports 4CIF (704x576) and CIF (352x288) resolutions, with minimum illumination of 0.1 lux in color mode and 0.003 lux in black and white mode. Samsung camera also provides a 22X optical zoom. The IP camera we used is Axis 232D dome camera. This camera provides resolutions maximum, 768x576 (PAL)/704x480 (NTSC) and minimum, 176x144 (PAL)/160x120 (NTSC), with 18X optical zoom and minimum illumination of 0.3 lux (color mode)/0.005 lux (black and white mode). Actually these cameras' features are similar to any other commercially available PTZ camera, therefore any camera with minimum CIF resolution and capable of producing more than 10 fps video frame rate would suffice for our detection method. The Samsung camera mounted on the forest watch tower is shown in Fig. 3.4.

Figure 3.4: Samsung analog camera mounted on the watch tower.

We have 9 actual fire videos recorded at night. The proposed algorithm was able to detect fires in 2 to 20 seconds after they became visible. The results of the

algorithm is compared with the non-adaptive version of the method. The results are summarized in Table 3.1. Fig. 3.5 shows a sample of a detected fire from video file V1. The other bright object in this frame is caused by the headlights of a fire truck. The proposed algorithm was able to separate the two and issue a correct alarm. Figs. 3.6 and 3.7 display detection results on videos that contain actual forest fires. In all test fires, an alarm is issued in less than 10 seconds after the start of the fire. The proposed adaptive fusion strategy significantly reduces the false alarm rate of the fire detection system by integrating the feedback from the guard (oracle) into the decision mechanism by using the active learning framework described in Section 3.3.

Figure 3.5: Correct alarm for a fire at night and elimination of fire-truck headlights.

Figure 3.6: Detection results on an actual forest fire at night.

Figure 3.7: Detection results on an actual forest fire at night.

A set of video clips containing various artificial light sources is used to generate Table 3.2. The snapshots from four of the videos are shown in Fig. 3.8. These videos contain an ice skating ring, seaside buildings, seaport and airport at night. Number of false alarms issued by different methods are presented. The

Table 3.1: Two different methods (LMS based, and non-adaptive) are compared in terms of frame numbers at which an alarm is issued for fire captured at various ranges and fps. It is assumed that the fire starts at frame 0.

Video Seq.	Range (km)	Frame Rate (fps)	Frame number of first alarm	
			LMS Based	Non-Adaptive
V1	5	25	221=10 sec	241
V2	6	25	100=4 sec	115
V3	6	25	216=8 sec	730
V4	7	25	151=6 sec	724
V5	1	25	83=4 sec	184
V6	0.5	25	214=8 sec	204
V7	0.1	30	59=2 sec	241
V8	0.1	30	74=3 sec	194
V9	0.1	30	56=2 sec	211

Table 3.2: Two different methods (LMS based, and non-adaptive) are compared in terms of the number of false alarms issued to video sequences that do not contain fire.

Video Seq.	Frame Rate (fps)	Duration (frames)	Number of false alarms	
			LMS Based	Non-Adaptive
V10	15	3000	1	4
V11	15	1000	0	2
V12	15	2000	0	3
V13	15	1000	0	2
V14	10	1900	0	1
V15	10	1200	0	5

proposed LMS based method produces the lowest number of false alarms in our data set. The proposed method produces a false alarm only to the video clip V10. On the other hand, the other method produces false alarms in all the test clips. In real-time operating mode the PTZ cameras are in continuous scan mode between predefined preset locations. They stop at each preset and run the detection algorithm for some time before moving to the next preset. By calculating separate weights for each preset we were able to reduce false alarms.

Figure 3.8: Snapshots from videos that are used for false alarm tests. (a) Ice skating ring at night, (b) seaside building lights at night, (c)seaport at night, (d) airport at night.

3.5 Application to Close Range Flame Detection

3.5.1 Sub-algorithms of Flame Detection Algorithm

Flame detection algorithm is developed to locate flame regions within the viewing range of visible range camera. Four sub-algorithms that make up the composite detection algorithm are: (i) detection of fire colored moving objects, performing (ii) temporal wavelet analysis, (iii) spatial wavelet analysis, and (iv) contour analysis of flame boundaries. The respective decision functions, $D_1(x, n)$, $D_2(x, n)$,

$D_3(x,n)$ and $D_4(x,n)$, are defined, for each pixel at location x of every incoming image frame at time step n.

i) Detection of Flame Colored Moving Objects

Moving Region Detection: For moving object detection the background subtraction algorithm developed in [19] is used. Let $I(x,n)$ represent the intensity value of the pixel at location x in the $n-th$ video frame I and let $B(x,n)$ denote the estimated background intensity value at the same pixel position. $T(x,n)$ is a recursively updated threshold at each frame n. The formulations for update equations of background and threshold can be found in [19, 40, 39].

It is assumed that regions significantly different from the background are moving regions. Estimated background image is subtracted from the current image to detect moving regions which corresponds to the set of pixels satisfying:

$$|I(x,n) - B(x,n)| > T(x,n) \tag{3.9}$$

are determined. These pixels are grouped into connected regions (blobs) and labeled by using a two-level connected component labeling algorithm [48].

Detection of Flame Colored Pixels: Markov models shown in Fig. 3.9 are used to detect flame in color video. Two models are trained off-line for both flame and non-flame pixels. States of the Markov models are determined according to color information as in [39].

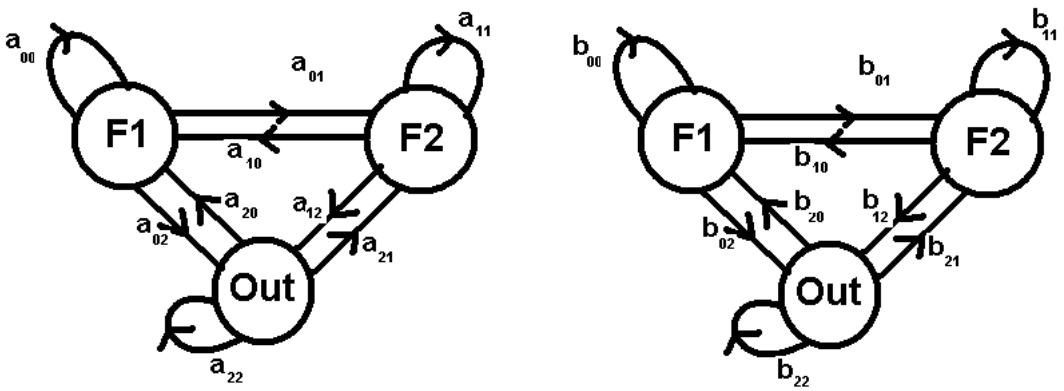


Figure 3.9: Three-state Markov models for flame (left) and non-flame (right) moving pixels.

The fire and flame color model of [39] is used for defining the flame-pixels. Namely; $R > R_T$, $R > G > B$, $S > (255 - R) * S_T / R_T$ where R,G, and B denote the color channels of RGB color space, S_T is the value of saturation when the value of R channel is R_T. In flame color classification, both values of R_T and S_T are defined according to various experimental results, and typical values range from 40 to 60 and 170 to 190, for S_T and R_T, respectively.

The three-state Markov model used for flame detection is presented in Fig. 3.9. The state $F1$ corresponds to a pixel having a fire color. The state $F2$ also corresponds to a pixel having a fire color but the fire color range of $F2$ is different from $F1$. The state called as Out is reserved for non-fire colored pixels. Temporal variation in RGB values of each pixel belonging to a moving region is used as a one dimensional (1-D) feature signal, $F = f(n)$, and fed to the Markov models shown in Fig. 3.9.

A moving pixel is classified as a fire pixel when the probability of obtaining the observed feature signal $F = f(n)$ given the probability model λ_1 is greater than the probability of obtaining the observed feature signal $F = f(n)$ given the probability model λ_2, i.e., when the pixel has fire color characteristics [13]:

$$p_1 = P(F|\lambda_1) > p_2 = P(F|\lambda_2), \tag{3.10}$$

where F is the observed feature signal, λ_1 and λ_2 represent the Markov models for fire and ordinary moving objects, respectively.

As the probability p_1 (p_2) gets a larger value than p_2 (p_1), the confidence level of this sub-algorithm increases (decreases). A decision function, $D_1(x,n)$, is defined describing the Markov Model based flame colored region detection sub-algorithm. The zero-mean decision function $D_1(x,n)$ is determined by the normalized difference of Markov Model probabilities [13]:

$$D_1(x,n) = \begin{cases} \frac{p_1-p_2}{p_1+p_2}, & \text{if x is a moving pixel} \\ -1, & \text{otherwise} \end{cases} \tag{3.11}$$

When a moving pixel is classified as a fire colored pixel, i.e., $p_1 \gg p_2$, $D_1(x,n)$ is close to 1. Otherwise, the decision function $D_1(x,n)$ is close to -1.

ii) Temporal Wavelet Analysis for Flicker Detection

The second sub-algorithm analyzes the frequency history of pixels in flame colored moving regions. Each pixel $I(x,n)$ at location x belonging a fire colored moving object in the image frame at time step n is fed to a two stage-filter bank. The signal $\tilde{I}_n(x)$ is a one-dimensional signal representing the temporal variations in color values of the pixel $I(x,n)$ at location x in the n^{th} image frame. Temporal wavelet analysis is carried out using the red channel values of the pixels. The two-channel subband decomposition filter bank is composed of half-band high-pass and low-pass filters with filter coefficients $\{-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\}$ and $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$, respectively.

Three-state Markov models are trained off-line for both flame and non-flame pixels to represent the temporal behavior (Fig. 3.10). These models are trained by using first-level wavelet coefficients $d_n(x)$ corresponding to the intensity values

$\tilde{I}_n(x)$ of the flame-colored moving pixel at location x as the feature signal. The number of zero crossings of the subband signal d_n in a few seconds can be used to discriminate between a flame pixel and an ordinary fire colored object pixel [26, 38, 39].

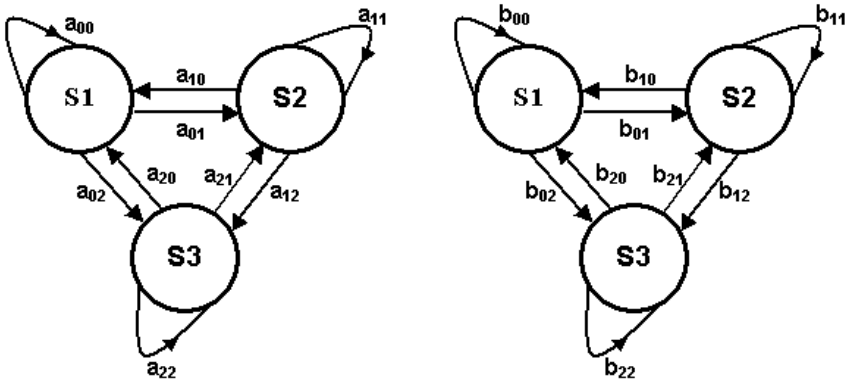


Figure 3.10: Three-state Markov models for flame (left) and non-flame (right) moving flame-colored pixels.

The states of HMMs are defined as follows: at time n, if $|w(n)| < T_1$, the state is in $S1$; if $T_1 < |w(n)| < T_2$, the state is $S2$; else if $|w(n)| > T_2$, the state $S3$ is attained. Here $|w(n)|$ denotes the absolute value of the wavelet coefficient corresponding to the currently analyzed pixel. $T_1 < T_2$ are experimentally determined thresholds. During the recognition phase, the HMM based analysis is carried out in pixels near the contour boundaries of flame-colored moving regions. The state sequence of length 20 image frames is determined for these candidate pixels and fed to the flame and non-flame pixel models [26, 38, 39].

Let $p1$ and $p2$ denote the probabilities obtained from the models for flame and non-flame pixels respectively. As the probability p_1 (p_2) gets a larger value than p_2 (p_1), the confidence level of this sub-algorithm increases (decreases). Therefore, the zero-mean decision function $D_2(x, n)$ is determined by the normalized difference of these probabilities:

$$D_2(x, n) = \frac{p_1 - p_2}{p_1 + p_2} \tag{3.12}$$

When a fire colored moving region is classified as a fire pixels according to frequency history, i.e., $p_1 \gg p_2$, $D_2(x,n)$ is close to 1. Otherwise, the decision function $D_2(x,n)$ is close to -1 [13].

The probability of a Markov model producing a given sequence of wavelet coefficients is determined by the sequence of state transition probabilities. Therefore, the flame decision process is insensitive to the choice of thresholds T_1 and T_2, which basically determine if a given wavelet coefficient is close to zero or not.

iii) Spatial Wavelet Analysis

The third sub-algorithm is the spatial wavelet analysis of moving regions containing fire colored pixels to capture color variations in pixel values. In an ordinary fire-colored object there will be little spatial variations in the moving region. On the other hand, there will be significant spatial variations in a fire region. The spatial wavelet analysis of a rectangular frame containing the pixels of fire-colored moving regions is performed. A decision parameter describing spatial variance is defined for this step, according to the energy of the wavelet subimages [26, 39]:

$$\xi = \frac{1}{M \times N} \sum_{k,l} |I_{lh}(k,l)| + |I_{hl}(k,l)| + |I_{hh}(k,l)|, \tag{3.13}$$

where $I_{lh}(k,l)$ is the low-high subimage, $I_{hl}(k,l)$ is the high-low subimage, and $I_{hh}(k,l)$ is the high-high subimage of the wavelet transform, respectively, and $M \times N$ is the number of pixels in the fire-colored moving region. If the decision parameter of the fourth step of the algorithm, ξ, exceeds a threshold, then it is likely that this moving and fire-colored region under investigation is a fire region. The decision function for this sub-algorithm is determined as follows:

$$D_3(x,n) = \begin{cases} 2\frac{\xi}{\xi_{max}} - 1, & \text{if } \xi \geq \xi_T \\ -1, & \text{otherwise} \end{cases}, \tag{3.14}$$

where ξ_{max} and ξ_T are experimentally determined parameters from videos containing flames. ξ_{max} is the largest value that ξ can take and ξ_T is a predefined threshold. The threshold determines the definite non-fire cases. The decision function is not sensitive to this threshold. One can also use $D_3(x,n) = 2\frac{\xi}{\xi_{max}} - 1$ as the decision function without the dependence on the threshold.

iv) Wavelet Domain Analysis of Object Contours

The fourth sub-algorithm of the proposed method analyzes the contours of flame colored objects. A one-dimensional (1-D) signal $x(\theta)$ is obtained by computing the distance from the center of mass of the object to the object boundary for $0 \leq \theta < 2\pi$. To determine the high-frequency content of a curve, we use a single scale wavelet transform. The feature signal $x[l]$ is fed to a filterbank and the low-band signal

$$c[l] = \sum_m h[2l - m]x[m] \tag{3.15}$$

and the high-band subsignal

$$w[l] = \sum_m g[2l - m]x[m] \tag{3.16}$$

are obtained. Coefficients of the lowpass and the highpass filters are $h[l] = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ and $g[l] = \{-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\}$, respectively [50, 51].

Since regular objects have relatively smooth boundaries compared to flames, the high-frequency wavelet coefficients of flame boundary feature signals have more energy than regular objects. Therefore, the ratio of the wavelet domain energy to the energy of the low-band signal is a good indicator of a fire region. This ratio is defined as

$$\rho = \frac{\sum_l |w[l]|}{\sum_l |c[l]|}. \tag{3.17}$$

The likelihood of the moving region to be a fire region is highly correlated with the parameter ρ. The Higher the value of ρ, higher the probability of the region belonging to flame regions [40]. The decision function for this sub-algorithm is defined as follows:

$$D_4(x,n) = \begin{cases} 2\frac{\rho}{\rho_{max}} - 1, & \text{if } \rho \geq \rho_T \\ -1, & \text{otherwise} \end{cases}, \tag{3.18}$$

where ρ_{max} is the maximum value of ρ and ρ_T is an experimentally determined threshold. The threshold determines the definite non-fire cases. The decision function is not sensitive to this threshold. One can also use $D_4(x,n) = 2\frac{\rho}{\rho_{max}} - 1$ as the decision without the dependence on the threshold.

3.5.2 Experimental Results

Three approaches are compared with each other in the experiments: (a) LMS based method, (b) weighted majority algorithm (WMA) based method and (c) a non-adaptive method. The method with no adaptive learning simply issues an alarm if all of the decision functions are 1 for the case of binary decision functions producing outputs 1 and -1 for fire and non-fire regions. Comparative tests are carried out with recordings containing actual fire and test sequences with no fires. Fire alarms are issued by all three methods at about the same time after fire becomes visible. However, there are some performance differences among the schemes in terms of false alarm rates.

The WMA [52] is summarized in Fig. 3.11. In WMA, as opposed to our method, individual decision values from sub-algorithms are binary, i.e., $d_i(x,n) \in \{-1,1\}$, which are simply the quantized version of real valued $D_i(x,n)$ defined in Section 3.4.1. In the WMA, the weights of sub-algorithms yielding contradictory decisions with that of the oracle are reduced by a factor of two in an un-controlled

Table 3.3: Three different methods (Non-adaptive, LMS based, WMA based) are compared in terms of frame numbers at which an alarm is issued for fire captured at various ranges and fps. It is assumed that the fire starts at frame 0.

Video Seq.	Range (m)	Frame Rate (fps)	Frame number/Time(sec) of first alarm					
			LMS Based		WMA Based		Non-Adaptive	
V1	2	10	44	4.4s	39	3.9s	51	5.1s
V2	50	30	48	1.6s	43	1.4s	62	2.0s
V3	50	30	49	1.6s	35	1.2s	37	1.2s
V4	30	30	106	3.5s	44	1.5s	No Alarm	
V5	1	30	64	2.1s	42	1.4s	18	0.6s
V6	5	10	43	4.3s	140	14s	38	3.8s
V7	60	30	334	11.1s	320	10.7s	349	11.6s
V8	80	30	73	2.4s	78	2.6s	No Alarm	
V9	100	30	41	1.4s	36	1.2s	12	0.4s
V10	10	15	48	3.2s	44	2.9s	56	3.7s
V11	20	15	49	3.3s	33	2.2s	48	3.2s
V12	50	15	46	3.0s	41	2.7s	80	5.3s
V13	40	15	47	3.1s	43	2.9s	18	1.2s
V14	30	15	72	4.8s	68	4.5s	20	1.3s
V15	70	30	212	7.0s	216	7.2s	156	5.2s
V16	15	30	259	8.6s	249	8.3s	163	5.4s
V17	20	30	68	2.3s	47	1.6s	No Alarm	
Average			94.3	4.0s	89.3	4.1s	79.1	3.6s

manner, unlike the proposed LMS based algorithm [53]. Initial weights for WMA are taken as $\frac{1}{M}$, as in the proposed LMS based scheme.

The LMS based scheme, the WMA based scheme, and the non-adaptive approach are compared with each other in the following experiments. In Tables 3.3 and 3.4, video recordings containing actual fires and video sequences with no fires are used.

LMS and WMA based decision fusion methods detect fires within 12 seconds but the method with no learning capability failed to produce alarms for 3 of the video sequences, as shown in Table 3.3. The LMS based method issues a correct alarm within 4 seconds on the average. The detection rates of the methods are comparable to each other. On the other hand, the proposed adaptive fusion strategy reduces the false alarm rate of the system by integrating the feedback

Weighted Majority(x,n)
for $i = 1$ to M **do**
  $w_i(0) = \frac{1}{M}$, Initialization
end for
if $\sum_{i:d_i(x,n)=1} w_i(n) \geq \sum_{i:d_i(x,n)=-1} w_i(n)$ **then**
  return 1
else
  return -1
end if
for $i = 1$ to M **do**
  if $d_i(x,n) \neq y$ **then**
    $w_i(n+1) \leftarrow \frac{w_i(n)}{2}$
  end if
end for

Figure 3.11: The pseudo-code for the Weighted Majority Algorithm

Table 3.4: Three different methods (Non-adaptive, LMS based, WMA based) are compared in terms of the number of false alarms issued for fire video sequences that do not contain fire.

Video Seq.	Frame Rate (fps)	Duration (frames)	Number of false alarms		
			LMS Based	WMA Based	Non-Adaptive
V18	25	1500	0	0	1
V19	25	2000	2	1	6
V20	25	2000	0	0	2
V21	25	150	0	2	1
V22	25	500	3	5	7
V23	25	1000	0	5	2
V24	25	150	0	0	2

from the guard (oracle) into the decision mechanism within the active learning framework described in Section 3.3.

A set of video clips that do not contain fire is used to generate the results in Table 3.4. These video clips are especially selected from recordings that contain fire colored moving objects. Number of false alarms issued by different methods are presented. The adaptive algorithms produce lower number of false alarms and LMS based scheme is better than WMA except for one video sequence. Total number of false alarms for the clips in Table 3.4 issued by the methods (a) the

LMS based scheme, (b) the WMA based scheme, (c) the non-adaptive approach are 5, 13 and 21, respectively.

Detection results for some of the test sequences in Table 3.3 are given in Fig. 3.12. Sample frames from the test videos in Table 3.4 are shown in Fig. 3.13.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.12: Examples from detected fires.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.13: Examples from the test set.

3.6 Summary

In this chapter, an active learning method for dynamic texture recognition is described. The algorithm combines the decision results of sub-algorithms that characterize a different aspect of the analyzed texture. Individual decisions of the sub-algorithms are combined together using a least-mean-square (LMS) based decision fusion approach, and texture/no-texture decision is reached by an active learning method. The algorithm is applied to wildfire detection at night and close range flame detection. The compound algorithm for night fire detection is composed of three sub-algorithms which produce their individual decision values. Each algorithm is designed to characterize an aspect of night fires. The main algorithm for flame detection comprises four sub-algorithms. Each algorithm is designed to characterize an aspect of flames. The decision functions of sub-algorithms yield their own decisions as confidence values in the range $[-1, 1] \in \mathbb{R}$. Computationally efficient sub-algorithms are selected in order to realize a real-time fire detection system working on a standard PC. The LMS based adaptive decision fusion strategy takes into account the feedback from the user of the application. Experimental results show that the learning duration is decreased with the proposed active learning scheme.

Chapter 4

Fire and Smoke Detection with a Moving Camera

In this chapter, an optical flow based algorithm is developed to segment moving objects from a continuously moving camera. The algorithm is applied to flame detection and wildfire smoke detection from a panning camera. Two-dimensional image motion can be obtained by projecting the three-dimensional motion of objects, relative to a camera, onto the image plane of the camera. Two-dimensional image motion can be observed as either instantaneous image velocities or discrete image displacements using time-ordered image sequences. These instantaneous image velocities are usually called the optical flow field or the image velocity field. If it is assumed that optical flow is a reliable approximation to two-dimensional image motion, it may then be used to recover the three-dimensional motion of the camera. Optical flow may be also used to perform motion detection and object segmentation [54, 55]. The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t + \delta$t using partial derivatives with respect to the spatial and temporal coordinates at every pixel position. There are different algorithms for optical flow computation, in our

method we use Lucas-Kanade optical flow calculation algorithm. The Lucas-Kanade method is a two-frame differential method for optical flow estimation. It introduces an additional term to the optical flow by assuming the flow to be constant in a local neighborhood around the central pixel under consideration at any given time [56]. We developed two methods for object segmentation from a moving camera. The first method uses three consecutive frames to register camera motion between the reference middle frame and the other frames. After motion estimation the previous and next frames are warped into the reference frame using the estimated affine transformation. This approach can be used for segmenting moving objects like walking people, cars, etc. but it cannot be used to segment smoke because the smoke plume does not have well defined boundaries as ordinary objects do and its movement is slow when viewed from long distances. To segment smoke from a moving camera we use a background subtraction method similar to panoramic backgrounds usually employed for using background subtraction with moving cameras.

4.1 Related Work

Motion segmentation is a fundamental component of many surveillance applications. The simplest method used to segment motion is background subtraction which involves constructing a reference frame to represent the background of the scene and subtracting each incoming image from the reference frame [19, 25]. Most background subtraction algorithms assume that the camera is stationary and additional steps are required to compensate for camera motion when the camera is non-stationary. Several algorithms for camera motion compensation are developed that use optical flow and change detection [57]. In [58], a new global motion estimation algorithm for change detection with moving camera is proposed. The algorithm assigns confidence measures to the motion vectors for

each block. The confidence measure is determined by combining the "cornerness" and "distinctness" measure of blocks.

In [59], an image registration and moving object detection system is designed to work in real-time on a standard PC using probabilistic methods. The displacement of each pixel is described as probability distribution over a matrix of possible displacements. The registration parameters are computed using a small set of randomly selected points. The consistency of the probabilistic displacement of image points with the global image motion is used to detect moving objects.

Qi and Ghazal describe a global motion estimation algorithm oriented to video object segmentation. They propose a hierarchical differential GME. The method combines three-step search and motion parameters prediction as initial estimation step to increase efficiency. The outliers introduced by local motion are rejected using a robust estimator that uses object information. For the first frame, when the object information is unavailable, a robust estimator is proposed which rejects outliers by examining their distribution in local neighborhoods of the error between the current and the motion-compensated previous frame [60].

Arnell and Petersson propose a segmentation algorithm based on a special representation of optical flow, on which u-disparity is applied. The background flow is approximated by a quadratic function and u-disparity is used to indirectly find and mask it out. Contrast filtering is used to increase robustness in the optical flow calculation [61]. A global motion estimation method for aerial imagery is proposed by fitting the optical flow field [62]. An optical flow evaluation scheme is introduced to choose a small set of reliable flows for fitting, as opposed to the traditional least-squares regression technique which is sensitive to outliers. The improvement of the GME algorithm is achieved by removing almost all of the flow outliers that are unfit for fitting.

In [57], optical flow and change detection methods are combined with a boundary extraction algorithm to segment moving objects. For change detection they use three consecutive video frames, a backward frame, the frame of interest and a forward frame instead of two adjacent time frames. Backward and forward frames are compensated relative to the reference frame using optical flow based simultaneous iterative camera motion compensation and background estimation. Compensated backward and forward frames are subtracted from the frame of interest and then compared with the estimated background models for intensity change detection. Then the approximate shape of the objects are acquired using these change detection results.

Another method for moving object segmentation from moving cameras is to use panoramic backgrounds. In this method a background model is generated by stitching together the individual images representing the stationary parts of the scene into a spherical or cylindrical mosaic. The moving regions are then detected by registering the incoming images into the panoramic background [63, 64, 65]. However, this approach is computationally costly to implement in real-time because both the mosaic generation and trigonometric registration required for mapping the image to the background are time consuming procedures. In, Dellaert and Collins developed a simpler method for tracking the pan and tilt of a PTZ camera [66]. The initial background model is generated methodically from a set of images with known pan and tilt settings. Instead of a background mosaic they use the individual images directly and choose the one closest to the current images in terms of their distance in the pan-tilt space. Then they use a planar projective transformation between the current image and the reference background.

In our method we use modified versions of the methods in [57] and [66]. For moving object detection we find the optical flow between the reference frame and backward and forward frames. For outliers rejection we use RANSAC method

and then find an affine transformation between the frames. This method is applicable to both motion detection and fire detection. For smoke detection with a moving camera we use a background subtraction method in which we first gather a collection of images from the non-stationary part of the background assuming that camera is continuously panning with constant tilt setting. For the first frame we search all of the background images and find the one closest to the current image. We then warp the current image to the background using an affine transformation. For the next frames we only look for the background images closer to the previously selected one to reduce the search time. The difference between the warped image and the background are used to find the moving objects.

4.2 Detection Algorithm

4.2.1 Optical Flow Background

The starting point for most optical flow estimation algorithms is the brightness constancy assumption which states that the pixel intensities are preserved from one frame to the next,

$$I(x,t) = I(x+u, t+1), \tag{4.1}$$

where $I(x,t)$ is image intensity as a function of pixel coordinate x and time t, and u is the $2D$ velocity. The brightness constancy assumption does not exactly hold for most practical estimation problems. Using Eq. 4.1 generally leads to satisfactory results as stated in [67].

If we assume that the displaced image can be approximated by a first-order Taylor series:

$$I(x+u, t+1) \approx I(x,t) + u \cdot \nabla I(x,t) + I_t(x,t), \tag{4.2}$$

where $\nabla I \equiv (I_x, I_y)$ and I_t are the spatial and temporal partial derivatives of the image I. Ignoring higher-order terms in the Taylor series and then substituting Eq. 4.2 into Eq. 4.1, the following equation is obtained:

$$\nabla I(x,t) \cdot u + I_t(x,t) = 0 \tag{4.3}$$

which is called the gradient constraint equation and gives the relation between the velocity and the image derivatives at one image location. If we have only two frames, and cannot estimate I_t, we can replace $I_t(x,t)$ in Eq. 4.3 by $\delta I(x,t) \equiv I(x,t+1) - I(x,t)$ and obtain a new constraint equation [56].

The Lucas-Kanade optical flow method solution assumes a locally constant flow. The additional constraint needed for the estimation of the flow field is introduced in this method by assuming that the flow u is constant in a small window of size $m \times m$ with $m > 1$, which is centered at position x. This method constrains u using the gradient constraints from nearby pixels using the assumption that they share the same velocity. The velocity that minimizes the constraint errors is found using the least-squares (LS) estimation [56]:

$$E(u) = \sum_x g(x)[u \cdot \nabla I(x,t) + I_t(x,t)]^2, \tag{4.4}$$

where $g(x)$ is a weighting function which is usually chosen to be Gaussian that determines the support of the estimator. The least squares optical flow estimate is the $2D$ velocity u that minimizes $E(u)$. The minimum of $E(u)$ can be found by equating its derivatives with respect to u to zero:

$$\partial E(u_1,u_2)u_1 = \sum_x g(x)[u_1 I_x^2 + u_2 I_x I_y + I_x I_t] = 0, \tag{4.5}$$

$$\partial E(u_1,u_2)u_2 = \sum_x g(x)[u_1 I_x^2 + u_2 I_x I_y + I_x I_t] = 0, \tag{4.6}$$

where u_1 and u_2 are the elements of the flow.

In matrix form these equations can be written as:

$$\mathbf{M}u = \vec{b}, \tag{4.7}$$

where $\mathbf{M}$ and $\vec{b}$ are defined as:

$$\mathbf{M} = \begin{bmatrix} \sum gI_x^2 & \sum gI_xI_y \\ \sum gI_xI_y & \sum gI_y^2 \end{bmatrix}, \vec{b} = - \begin{bmatrix} \sum gI_xI_t \\ \sum gI_yI_y \end{bmatrix} \tag{4.8}$$

When $\mathbf{M}$ has full rank, then the least squares estimate of the flow can be found as $u = \mathbf{M}^{-1}\vec{b}$ [67].

In our application, we use the pyramidal implementation of Lucas-Kanade optical flow method described in [68]. In Fig. 4.1, the optical flow between two consecutive frames of a video sequence is shown as directed arrows that describe the motion of evenly distributed 480 points from the first frame.

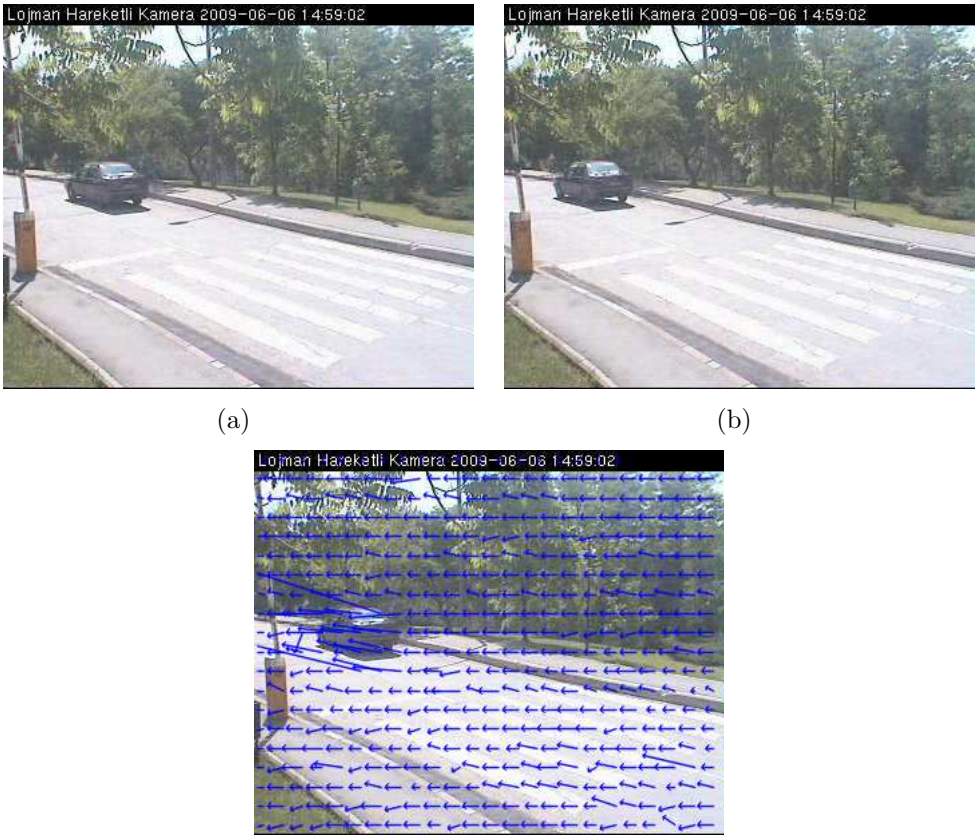


Figure 4.1: Optical flow between 2 consecutive frames: (a) frame 1; (b) frame 2; (c) optical flow.

4.3 Motion Detection Algorithm

The flowchart of the motion detection algorithm is shown in Fig. 4.2. Global motion compensation step is carried out using optical flow and RANSAC (Ransom Sample Consensus). For registering the backward and forward frames an affine transformation is found that describes the camera motion between these frames and the reference frame. A six parameter affine model as in Eq. 4.9 is used:

$$\begin{aligned} x' &= a_0 x + a_1 y + a_2 \\ y' &= a_3 x + a_4 y + a_5 \end{aligned}, \tag{4.9}$$

where (x, y) is the location of the pixel in the n^{th} image frame I_n, (x', y') is location of the pixel in the previous, I_{n-1} or the next frame, I_{n+1} , and $\vec{a} = (a_0, a_1, a_2, a_3, a_4, a_5)$ is the vector of affine transformation parameters. The affine model is an approximation to projective motion model and the projected 2D motion of most camera motions can be described by an affine transformation [69].

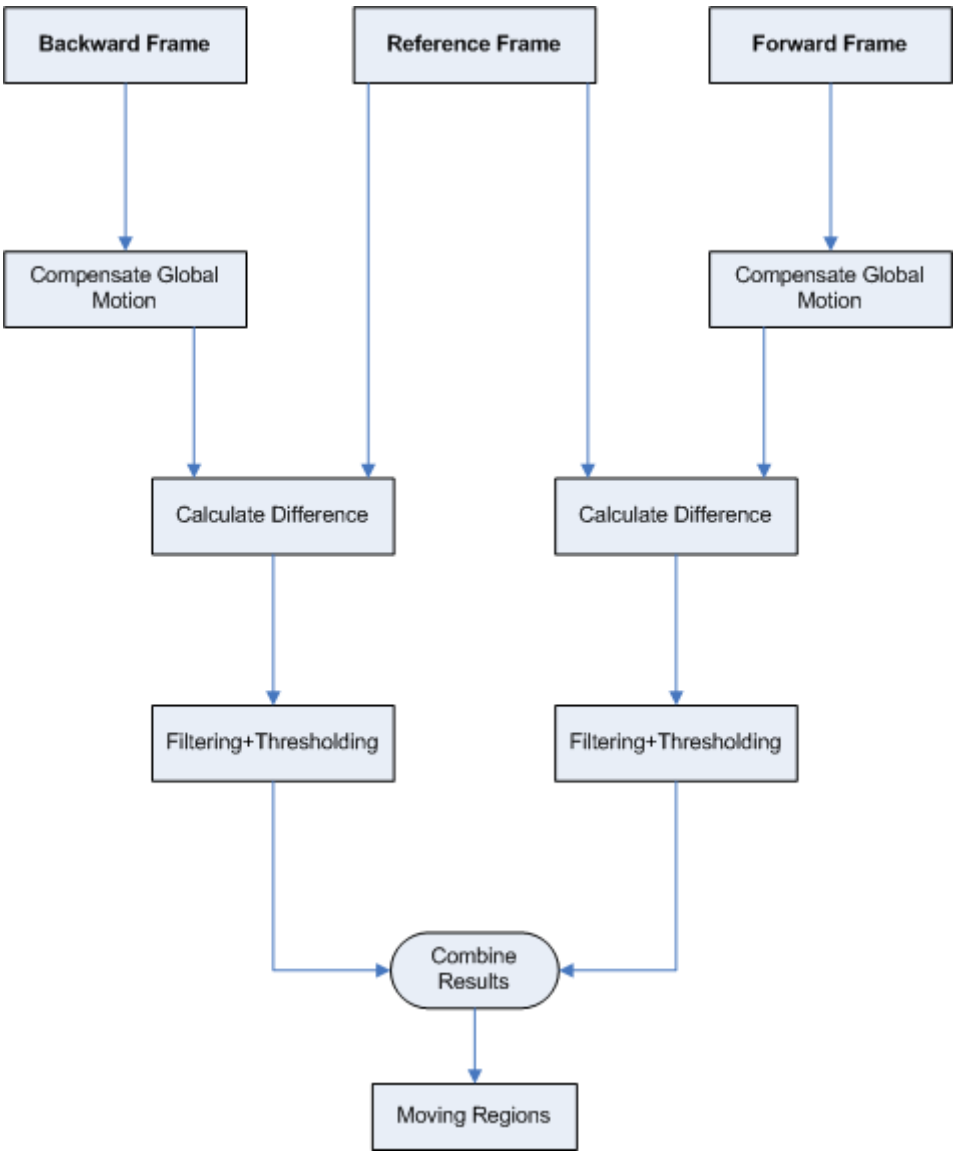


Figure 4.2: The flowchart of the moving region detection algorithm.

For optical flow estimation, evenly distributed points from the currently processed image are chosen and then the corresponding points in the reference image

are found. RANSAC is used to find the affine model parameters that can transform the point coordinates between the two frames. RANSAC is a method to estimate the parameters of a mathematical model from a set of observations. RANSAC differs from classical parameter estimation techniques such as least-squares method since it tries to detect and reject outliers which are the "gross errors" that do not fit the model [70]. For the global motion compensation problem, the inliers are the motion vectors that are caused by movement of the camera and the outliers are the local motion of moving objects.

The procedure for applying RANSAC to the results of optical flow is as follows; random three points are chosen from the first image and then an affine transformation is found between the corresponding points in the reference frame. All points selected in the first frame and the reference frame are checked with the estimated transform and if a point from features set can be transformed within some error threshold with the estimated model parameters it is added to the consensus set. If the size of the consensus set reaches a threshold value or the maximum number of iterations for RANSAC is reached the algorithm stops and the final model parameters are used to form the affine transformation vector.

In Figs. 4.3 and 4.4 the motion detection algorithm is used to find moving objects from a panning camera. As can be seen from the uncompensated difference frames in Figs. 4.3(d) and 4.3(e) it is difficult to identify moving objects when camera motion is present. The affine model parameters for backward and forward frames are:
$\vec{a_b} = (1.002690, -0.002519, -5.332665, -0.000968, 0.999622, 0.176728)$ and
$\vec{a_f} = (0.997627, 0.002666, 3.451948, 0.000571, 1.000087, -0.101921)$
respectively. Since the only camera motion is caused by panning the effective parameter of the model is a_2 which corresponds the translation in the x-direction.

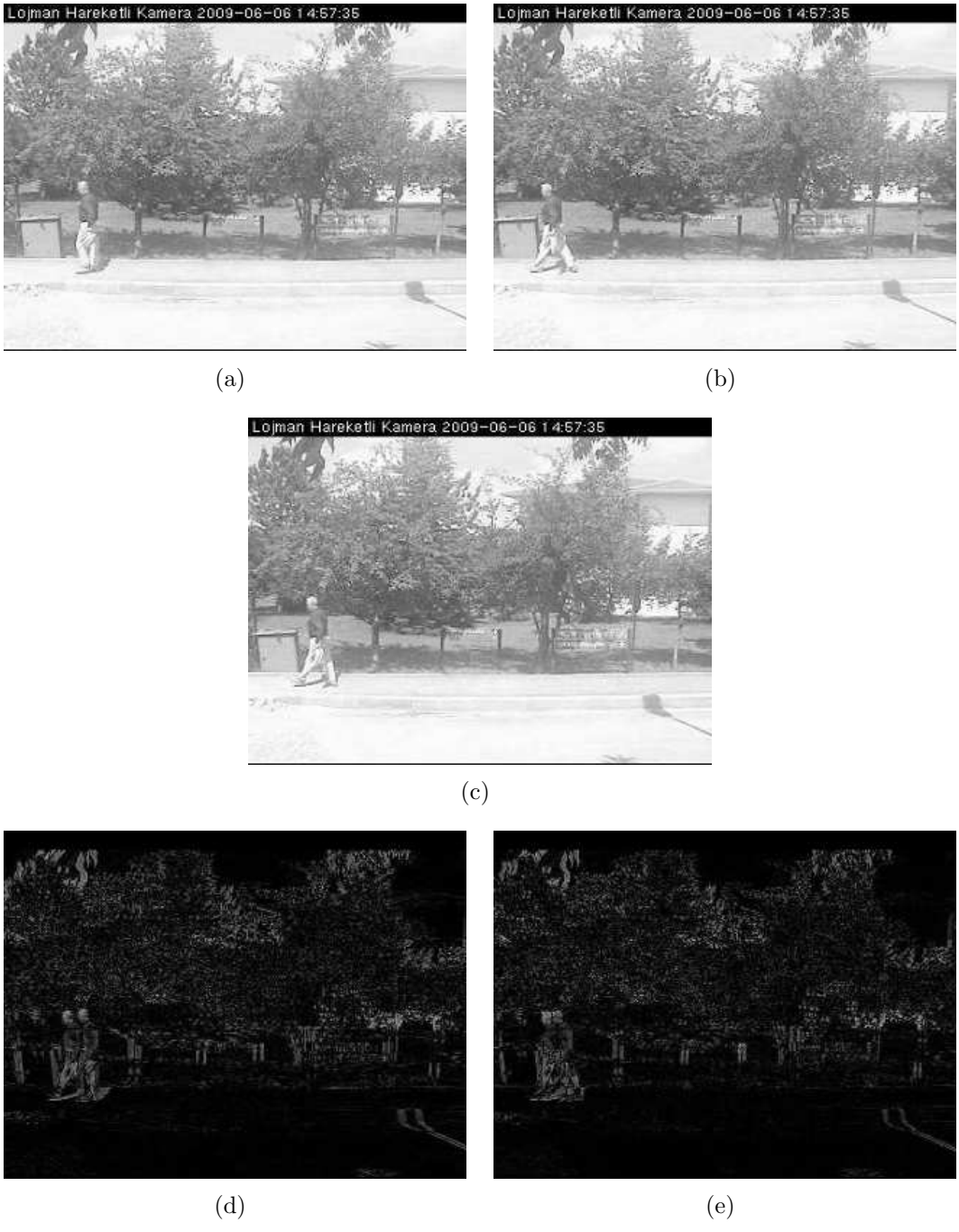


Figure 4.3: Three consecutive frames from a panning camera and their differences: (a) backward frame; (b) forward frame; (c) reference frame; (d) reference and backward difference; (e) reference and forward difference.

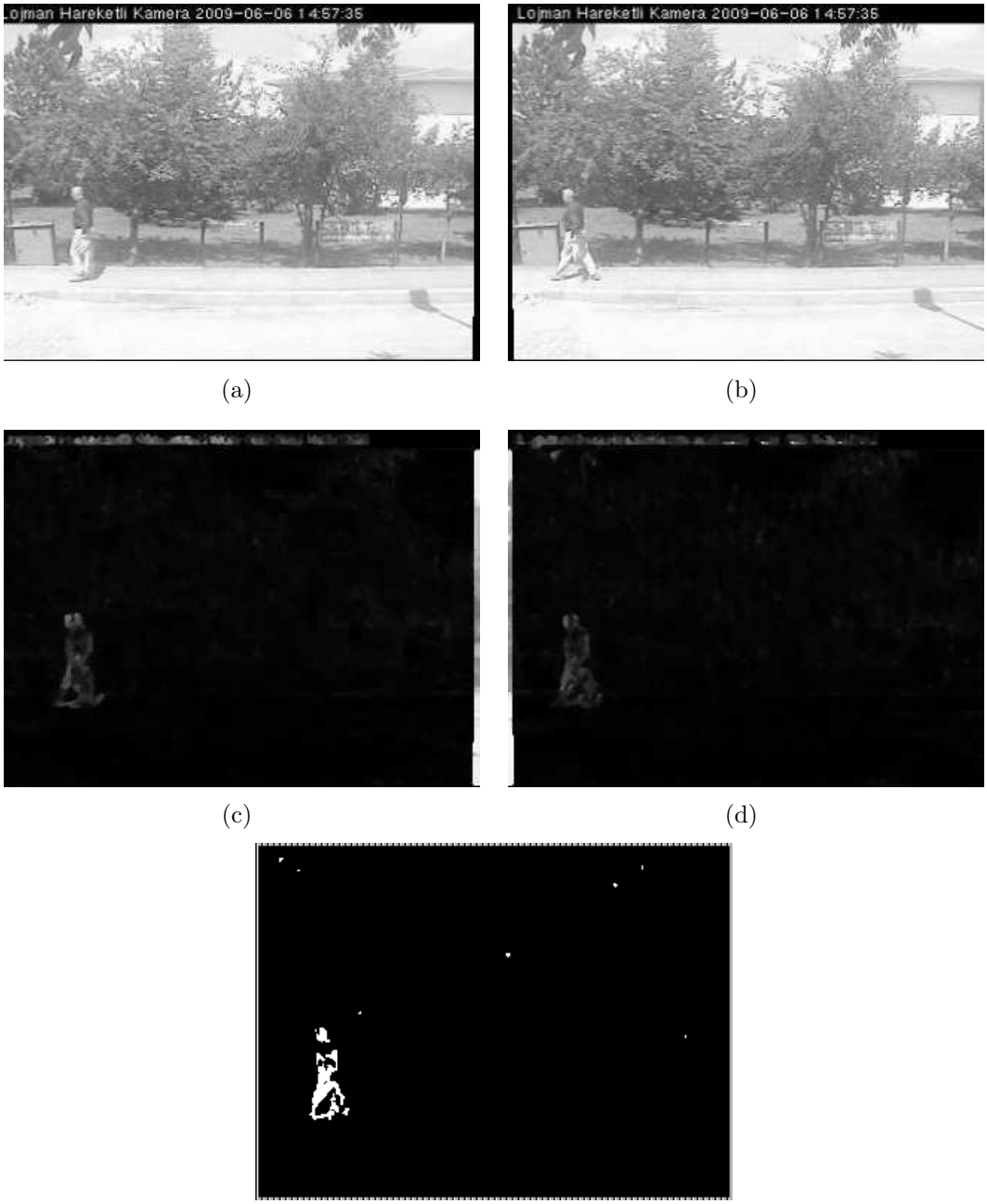


Figure 4.4: Sample application of the algorithm for detecting moving objects with a panning camera: (a) compensated backward frame; (b) compensated forward frame; (c) compensated forward difference; (d) compensated backward difference; (e) after thresholding and smoothing.

4.4 Background Subtraction with Motion Compensation

The three frame change detection method can be used to find moving objects from a moving camera moving sequence, but to detect slow moving or stationary objects we need an algorithm like background subtraction that can segment objects of interest. For background subtraction we use a method similar to the one described in [66]. We first gather images that form the stationary part of the scene to be monitored. Instead of forming a panoramic background by stitching together these images, we use them directly as a collection of background images. At the start of the analysis all images are searched and the background image that is closest to the current image is chosen as the background for that image. Then an affine transformation is found between the background and current images using optical flow and RANSAC using the method in Section 4.3. The difference between the compensated image and the background image is filtered and thresholded to find the objects that do not belong to the background. After the first frame only a small number of background images that are close to the previously selected background image in the pan-tilt space, are searched for a possible match to the incoming frame assuming that camera does not make sudden movements.

For matching the current image to a background image we use a simple normalized correlation based template matching algorithm. The number of background images that are sufficient for a correct representation of the background depends on the speed of the camera and the FPS (frames per second) of the video stream. Collecting too many images may slow down the background matching process and occupy more memory space, on the other hand if the number of background images is not enough the whole background cannot be represented and registration of the incoming frame to the background may not be possible.

In Fig. 4.6, the results of applying the background subtraction algorithm to detect moving objects from a panning camera are shown. The camera moves with a constant speed of $4.6°/sec$ and FPS of the stream is 10. 39 background images are selected from the stationary part of the scene as shown in Fig. 4.5. Since the camera completes a 360° turn in around 79 seconds we are representing each 9.2° turn by a single background image.

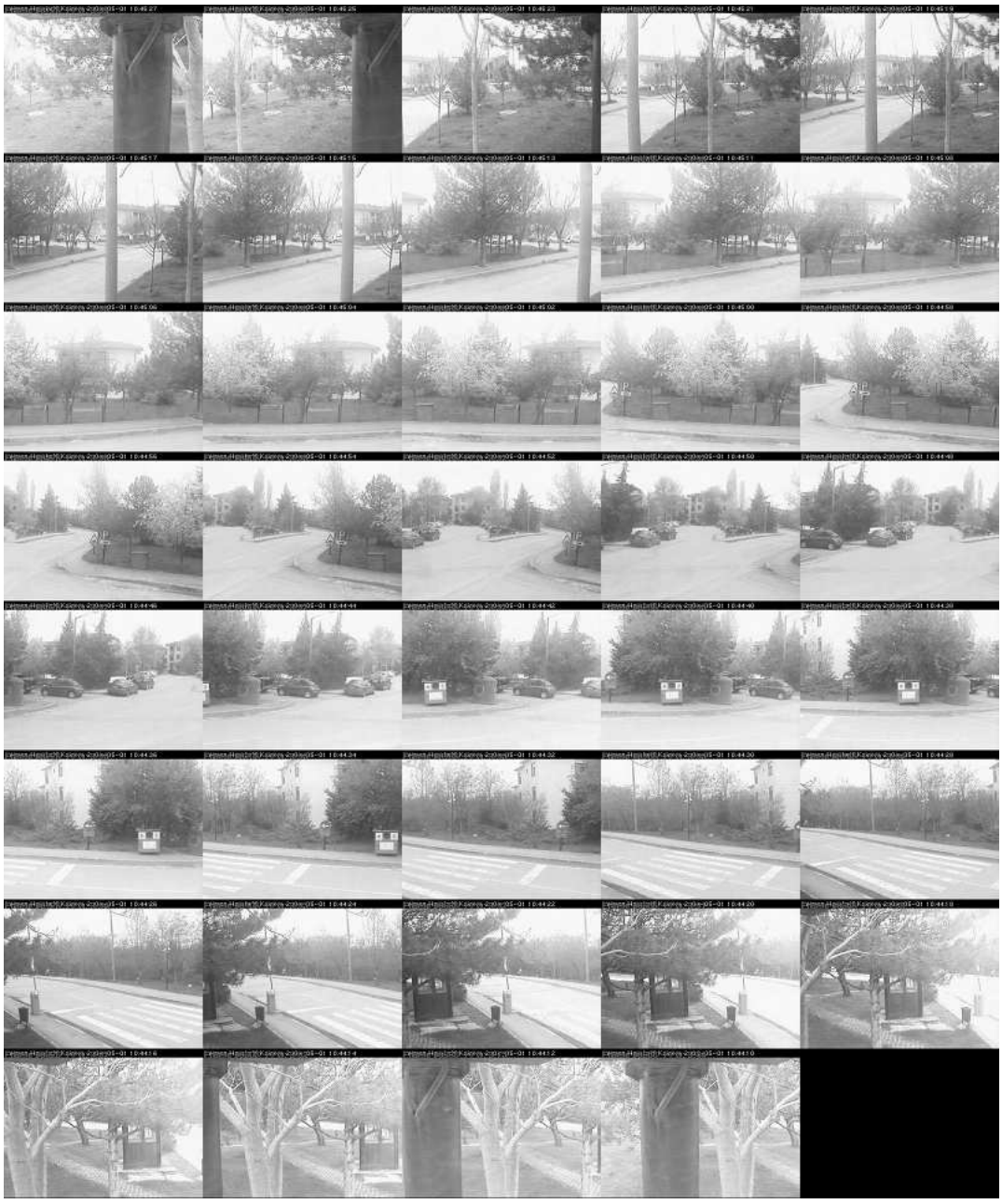


Figure 4.5: The background images used for the test in Fig. 4.6.

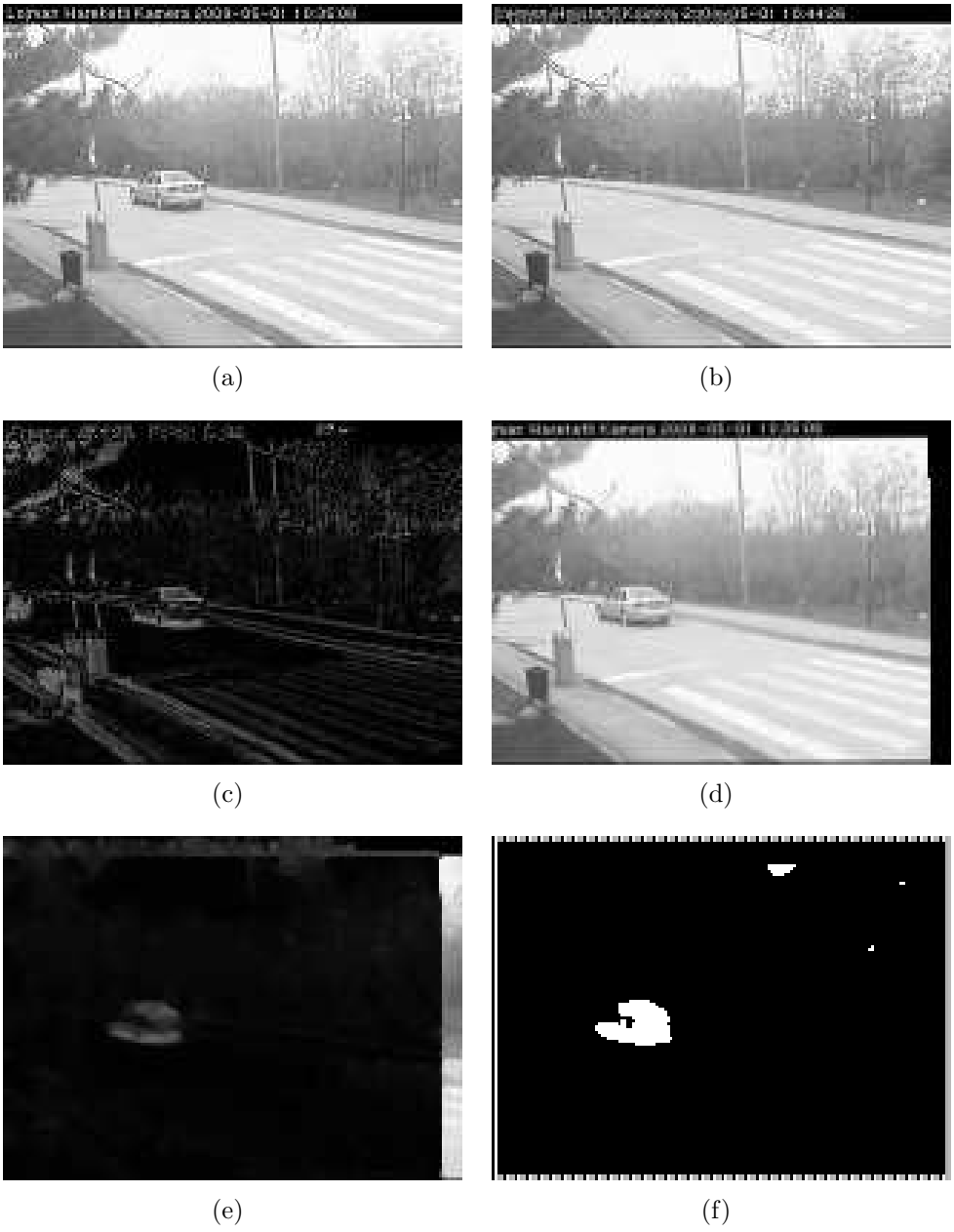


Figure 4.6: Background subtraction applied to a panning camera. The camera speed is $4.6°/sec$. The background images are taken at 2 second intervals; there is a total of 39 background frames representing the 360° range of the panning camera: (a) current frame; (b) selected background frame; (c) current frame and background difference; (d) compensated frame; (e) compensated difference; (f) after thresholding and smoothing.

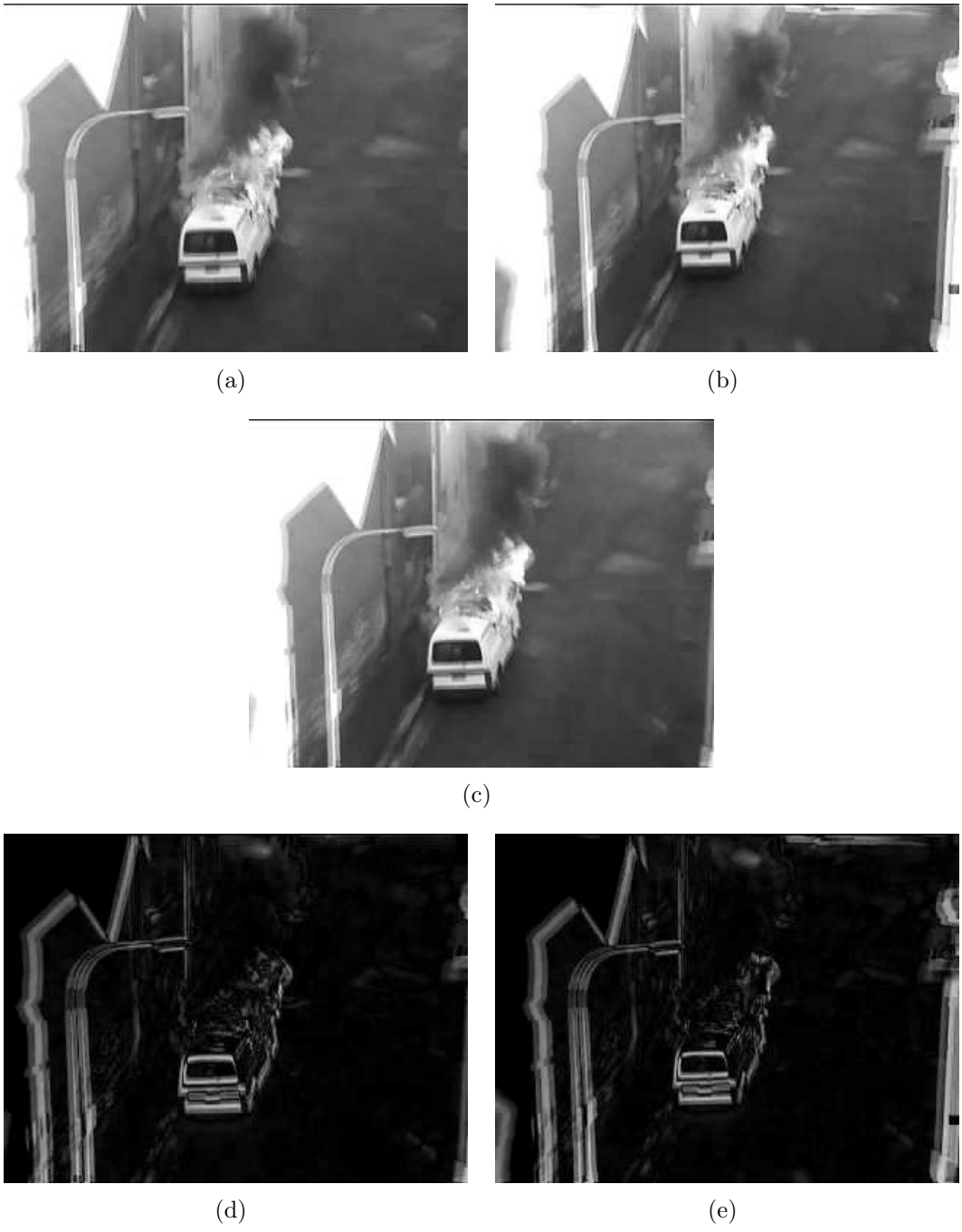
For a system working with live video streams we cannot use the same background images for a long time, because the lighting conditions or the stationary part background might change over time. Therefore, the background images

should be updated periodically to overcome this problem. Background update is performed by replacing the background image by the closest image that is mapped to that background. Since the background images should be stationary we use three frame change detection technique described in Section 4.3 to check if there are moving objects in the frame. If all frames that are mapped to the same background have moving objects that background frame is updated in the next round.

4.5 Detection Experiments

4.5.1 Fire Detection Experiments

In Figs. 4.7 and 4.8 the three frame change detection with motion compensation algorithm is applied to fire detection with a moving camera. First moving regions are found using the method in Section 4.3 and then fire colored pixels connected together to cover the whole flame regions because motion detection algorithm may only find the edges of flame regions. After finding fire colored moving regions the methods in Section 3.5.1 of Chapter 3 are applied to these regions to check if they are actual fire regions. We use three of the four sub-algorithms for flame detection (i) detection of flame colored moving objects, (ii) spatial wavelet analysis for flicker detection and (iii) contour analysis of fire colored region boundaries. Keeping temporal history of flame regions is difficult with a moving camera so we do not use this algorithm. We mark the pixels as belonging to a fire region when they meet the requirements of all three algorithms. Fig. 4.8(e) shows the moving regions that are found by the motion compensation algorithm. Fig. 4.8(f) shows the fire detection result after fire colored pixels are connected and fire detection algorithms are applied. The final segmentation result is shown in Fig. 4.9.



(a) (b)

(c)

(d) (e)

Figure 4.7: Three consecutive frames from camera that is zooming out of a burning van: (a) backward frame; (b) forward frame; (c) reference frame; (d) reference and backward difference; (e) reference and forward difference.

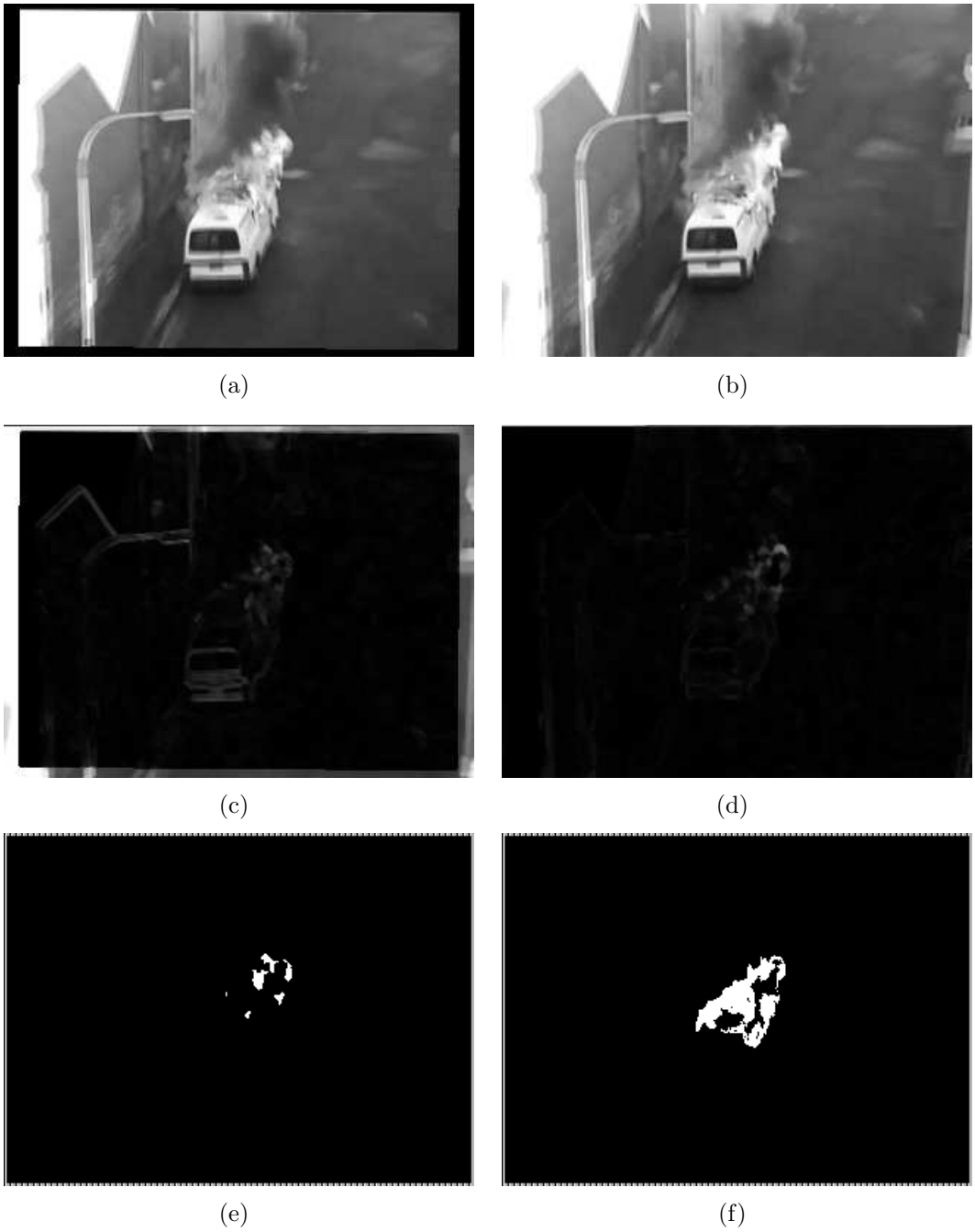


Figure 4.8: Sample application of the motion compensation algorithm for detecting fire with a moving camera: (a) compensated backward frame; (b) compensated forward frame; (c) compensated backward difference; (d) compensated forward difference; (e) after thresholding and smoothing; (f) after connecting fire colored pixels and applying fire detection algorithm.

Figure 4.9: Final fire detection result.

In Table 4.1, video sequences that contain close range flames are tested with the developed method to check the performance of the algorithm. There are 14 videos with 14,803 frames in total that are mostly recorded with hand-held moving cameras. The number of frames with flames, the number of correctly classified frames, incorrectly registered moving frames, and frames for which false alarms are issued are summarized in the table. The method was able to detect flames in most of the frames. Since videos are recorded with hand-held cameras the motion of the camera is not smooth which caused some of the frames to be incorrectly aligned. The majority of the false alarms are issued when the registration of the frames was unsuccessful.

Table 4.1: Fire detection results for 14 different video sequences mostly recorded with hand-held moving cameras. The videos are compared in terms of total number of frames with flame, the detected number of frames, incorrectly registered frames and frames with false positive alarms.

Video Seq.	# of Frames	# of Frames with flame	Correctly classified frames	Incorrectly registered frames	False positive frames
V1	268	260	234	0	0
V2	210	210	204	0	1
V3	600	590	535	3	4
V4	571	565	560	2	3
V5	1000	960	794	2	0
V6	451	450	410	0	0
V7	500	500	405	0	5
V8	1081	1070	1001	30	18
V9	2518	2518	2331	23	12
V10	2608	2600	2294	10	8
V11	1200	1200	1093	3	4
V12	2248	2248	2169	6	0
V13	1219	1219	1095	7	5
V14	330	330	325	0	0

4.5.2 Smoke Detection Experiments

The application of the background subtraction algorithm for smoke detection is shown in Fig. 4.10. The current image is mapped to the selected background image using the affine transformation that is found using optical flow. The difference between the warped current image and the selected background image is used to find the new objects in the scene. After that smoke colored object detection and shadow elimination steps from the wildfire smoke detection algorithm in [13] are applied. If possible smoke regions are detected using the background subtraction algorithm the camera stops panning and runs the complete smoke detection algorithm in [13] for some time to verify the existence of smoke.

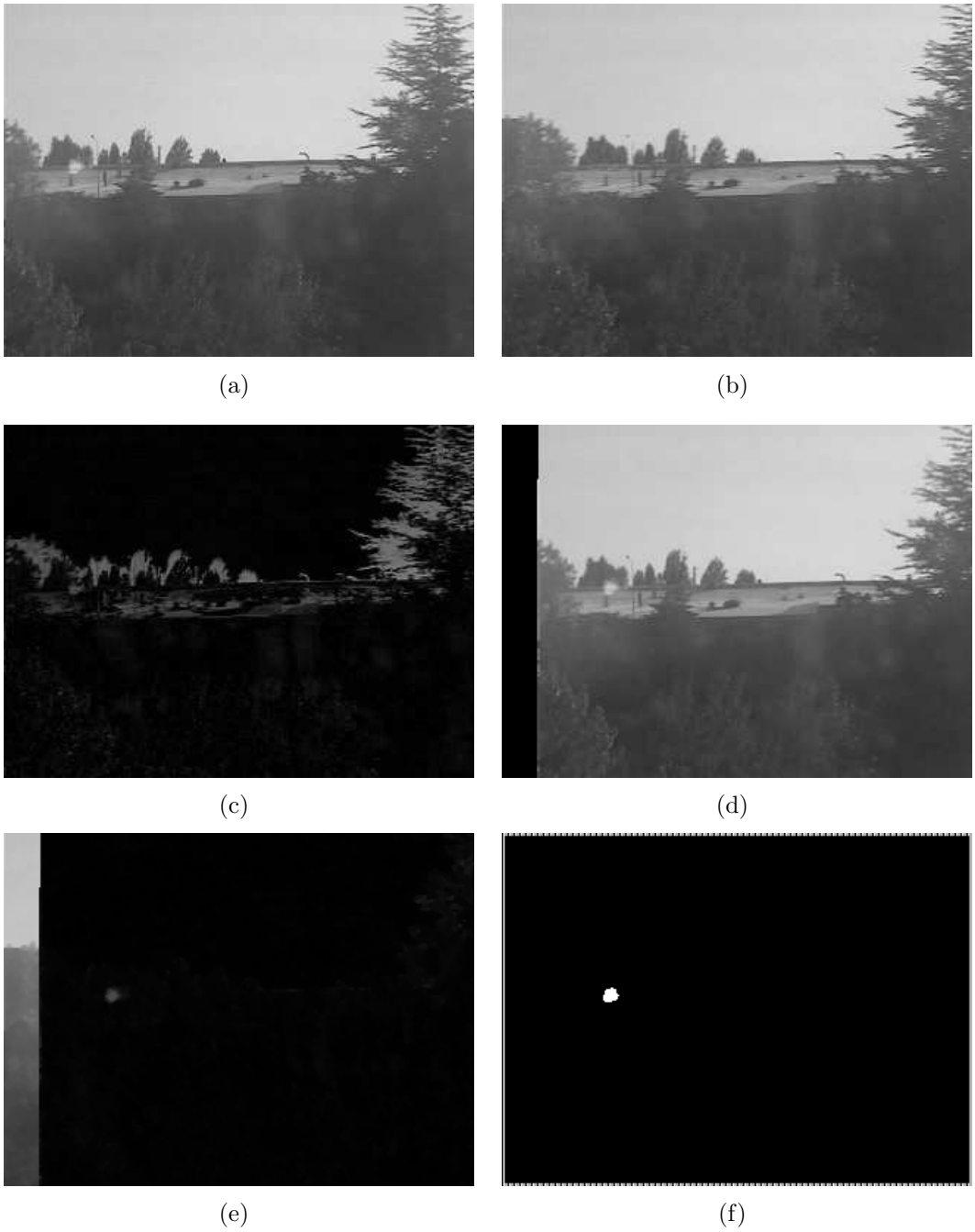


Figure 4.10: Sample application of the background subtraction algorithm for detecting smoke with a moving camera: (a) current image; (b) selected background image; (c) uncompensated difference; (d) compensated current frame; (e) compensated difference; (f) after thresholding the difference and applying the forest smoke detection algorithm.

In Table 4.2 the results of applying the background subtraction algorithm to four different video sequences are shown.

Table 4.2: Smoke detection results for four different video sequences. The videos are compared in terms of total number of frames with smoke, the detected number of frames, incorrectly registered frames and frames with false positive alarms.

Video Seq.	# of Frames	# of Frames with smoke	Correctly classified frames	Incorrectly registered frames	False positive frames
V1	186	100	93	5	6
V2	420	420	417	10	8
V3	897	840	840	40	15
V4	572	532	523	9	3

4.6 Summary

In this chapter global motion compensation methods are used to segment moving objects from a moving camera. The methods are actually intended to detect fire and smoke from moving camera sequences. For fire detection we use three frame change detection methods for registering the backward and forward frames to the reference frame. For registering the global motion of the camera optical flow between the two frames is calculated by RANSAC and an affine transformation is found that maps the camera motion from frame to frame. Since the flicker of flames has a detectable frequency we use this method to detect close range fires. For detecting long range wild fire smokes we use a background subtraction algorithm because they usually move slower when viewed from long distances. A collection of images are used to represent the stationary part of the background, and when a new frame arrives a background image that is closest to the frame is chosen as the background for this frame. The global motion compensation method is then used to register the current frame to the background and the difference between the registered image and the background is used to find the new objects in the background. Experimental results show that these can be used to detect fire smoke from continuously moving cameras.

Chapter 5

Conclusion and Future Work

Automatic recognition of fire, smoke, VOC plumes, etc. is important for video based surveillance systems used in industrial plants, chemical factories and outdoor systems for monitoring forestal areas. In this thesis, some signal and image processing techniques for automatic recognition of some classes of dynamic textures are developed. Specifically, algorithms are developed for VOC leak detection in IR videos, wild fire detection at night, and fire and smoke detection with a moving camera.

The algorithm developed for VOC leak detection uses the observation that during the initial stages of emission the leak causes a gray scale value change around the leaking equipment and these regions become darker in IR camera images. Background subtraction and change detection algorithms are used to detect this temporal variation and VOC plume. The method is tested with videos recorded at TÜPRAŞ (Türkiye Petrol Rafinerileri A.Ş.) factory in Kırıkkale and showed promising results.

The LMS based algorithm developed and applied to wildfire smoke detection in [13] is used to detect wild fires at night. The smoke visible in the early stages of wild fires is not visible at night and the fire becomes visible only when the

flames reach a certain height. The wild fire flames appear as bright spots in a visible range PTZ camera. The method is integrated into the surveillance system used by Directorate of Forestry of Turkey for monitoring forestal areas for early detection of forest fires. During day time the system uses the wild fire smoke detection and at night switches to our method for detecting night fires. The proposed fire detection scheme with LMS based active learning method is tested with forest surveillance recordings captured from cameras mounted on top of forest watch towers near Antalya and Muğla regions in Turkey. The LMS based algorithm is also used to detect close range flames by using the algorithms developed in ([38]-[40]) in an active learning framework.

The cameras used in the early wild fire detection system are in continuous scan mode between predefined preset (park) positions. At each preset position the camera waits for a certain period of time and runs the detection algorithm. If the area to be monitored is large the number of park positions should also be large and this could introduce additional delays in the detection. To overcome this problem we also developed methods for segmenting fire and smoke from continuously moving cameras.

For fire detection we used three frame change detection method for registering the backward and forward frames to the reference frame. For registering the global motion of the camera optical flow between the two frames is calculated and then using RANSAC an affine transformation is found that maps the camera motion from one frame to the next. Since the flicker of flames has a detectable frequency we use this method to detect close range fires.

For detecting long range wild fire smokes we used a background subtraction algorithm since they usually move slower. A collection of images are used to represent the stationary part of the background, and when a new frame arrives a background image that is closest to the frame is chosen as the background for this frame. The global motion compensation method is then used to register

the current frame to the background and the difference between the registered image and the background is used to find the new objects in the background. Experimental results show that these can be used to detect fire smoke from continuously moving cameras.

5.1 Future Work

The LMS algorithm used for wildfire detection uses a supervised learning approach. The oracle, the security guard watching the surveillance camera in this case, decides whether the detected regions are actual fire regions. This approach is time consuming since the oracle needs to mark every frame. An improvement over this approach would be to use an unsupervised learning method to classify frame pixels without the help of an oracle. We are planning to implement the unsupervised version of the active learning algorithm.

The motion compensation method developed for fire and smoke detection works well when the background of the scene monitored by the camera consists of rigid objects; like buildings, cars, etc. For the case of non-rigid dynamic backgrounds the algorithm sometimes fails to register the image to the background. As a future work we will implement global motion compensation algorithms that deal with dynamic backgrounds.

Bibliography

[1] Coblentz Society, "Evaluated infrared reference spectra," in *NIST Chemistry webbook, NIST Standard Reference Database* (P. J. Linstrom and W. G. Mallard, eds.), vol. 69, Gaithersburg MD, 20899, USA: National Institute of Standards and Technology.

[2] A. Ghoreyshi and R. Vidal, "Segmenting Dynamic Textures with Ising Descriptors, ARX Models and Level Sets," in *Proceedings of the European Conference on Computer Vision / Dynamical Vision Workshop (ECCV/DV)*, pp. 127–141, 2006.

[3] S. Fazekas and D. Chetverikov, "Analysis and performance evaluation of optical flow features for dynamic texture recognition," *Signal Processing:Image Communication*, vol. 22(7-8), pp. 680–691, 2007.

[4] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 58–63, December 2001.

[5] M. Szummer and R. W. Picard, "Temporal texture modeling," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 823–826, 1996.

[6] Z. Lu, W. Xie, J. Pei, and J. Huang, "Dynamic texture recognition by spatio-temporal multiresolution histogram," in *Proceedings of the IEEE Workshop*

on Motion and Video Computing (WACV/MOTION), vol. 2, pp. 241–246, 2005.

[7] K. Fujita and S. Nayar, "Recognition of dynamic textures using impulse responses of state variables," in *Proceedings of the Third International Workshop on Texture Analysis and Synthesis (Texture)*, pp. 31–36, 2003.

[8] R. Vidal and A. Ravichandran, "Optical Flow Estimation and Segmentation of Multiple Moving Dynamic Textures," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 516–521, 2005.

[9] C.B. Liu and R.S. Lin and N. Ahuja, "Modeling dynamic textures using subspace mixtures," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1378–1381, 2005.

[10] K. Otsuka, T. Horikoshi, S. Suzuki, and M. Fujii, "Feature extraction of temporal texture based on spatiotemporal motion trajectory," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 1047–1051, 1998.

[11] C. Peh and L.F. Cheong, "Synergizing spatial and temporal texture," *IEEE Transactions on Image Processing*, vol. 11, pp. 1179–1191, 2002.

[12] A. Rahman and M. Murshed, "Real-time temporal texture characterisation using block-based motion co-occurrence statistics," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1593–1596, 2004.

[13] B.U. Toreyin and A.E. Cetin, "Wildfire Detection Using LMS Based Active Learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1461–1464, 2009.

[14] U. S. Environmental Protection Agency, *Protocol for Equipment Leak Emission Estimates*. EPA-453/R-95-017, 1995.

[15] L. Zhou and Y. Zeng, "Automatic alignment of infrared video frames for equipment leak detection," *Analytica Chimica Acta*, vol. 584/1, pp. 223–227, 2007.

[16] M. Lev-On, H. Taback, D. Epperson, J. Siegell, L. Gilmer, and K. Ritterf, "Methods for quantification of mass emissions from leaking process equipment when using optical imaging for leak detection," *Environmental Progress*, vol. 25, no. 1, pp. 49–55, 2007.

[17] U. S. Environmental Protection Agency, *Development of Emissions Factors and/or Correlation Equations for Gas Leak Detection, and the Development of an EPA Protocol for the Use of a Gas-imaging Device as an Alternative or Supplement to Current Leak Detection and Evaluation Methods.* Final Report to Texas Council on Environmental Technology and the Texas Commission on Environmental Quality, 2004.

[18] B. U. Toreyin and A. E. Cetin, "Volatile Organic Compound Plume Detection Using Wavelet Analysis of Video," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1836–1839, 2008.

[19] R. Collins, A. Lipton, and T. Kanade, "A system for video surveillance and monitoring," in *Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, April 1999.

[20] R. Polikar, R. Shinar, V. Honavar, L. Udpa, and M. Porter, "Detection and identification of odorants using an electronic nose ," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, pp. 3137–3140, 2001.

[21] M. Maute and S. Raible and F. E. Prins and D. P. Kern and H. Ulmer and U. Weimar and W. Göpel, "Detection of volatile organic compounds (VOCs) with polymer-coated cantilevers," *Sensors and Actuators B: Chemical*, vol. 58, no. 1-3, pp. 505 – 511, 1999.

[22] A. K. Srivastava, "Detection of volatile organic compounds (VOCs) using SnO2 gas-sensor array and artificial neural network," *Sensors and Actuators B: Chemical*, vol. 96, no. 1-2, pp. 24 – 37, 2003.

[23] A.E. Cetin and M.B. Akhan and B.U. Toreyin and A. Aksay, "Characterization of motion of moving objects in video," *US Patent No.20040223652, pending*, 2004.

[24] F. Porikli, Y. Ivanov, and T. Haga, "Robust abandoned object detection using dual foregrounds," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[25] B.U. Töreyin, "Moving object detection and tracking in wavelet compressed video," Master's thesis, Bilkent University, Department of Electrical and Electronics Engineering, Ankara, Turkey, 2003.

[26] B.U. Toreyin and Y. Dedeoglu and U. Gudukbay and A.E. Cetin, "Computer Vision Based System for Real-time Fire and Flame Detection," *Pattern Recognition Letters*, vol. 27, pp. 49–58, 2006.

[27] B.V. Dasarathy and B.V. Sheela, "A composite classifier system design: Concepts and methodology," *Proceedings of the IEEE*, vol. 67, pp. 708–713, May 1979.

[28] T. K. Ho, J. Hull, and S. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66–75, Jan 1994.

[29] V. I. Gorodetskiy and S. V. Serebryakov, "Methods and algorithms of collective recognition," *Automation and Remote Control*, vol. 69, no. 11, pp. 1821–1851, 2008.

[30] A. Kumar and D. Zhang, "Personal authentication using multiple palmprint representation," *Pattern Recognition*, vol. 38, no. 10, pp. 1695 – 1704, 2005.

[31] X. Tang and Z. Li, "Video based face recognition using multiple classifiers," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 345–349, May 2004.

[32] M. A. García and D. Puig, "Supervised texture classification by integration of multiple texture methods and evaluation windows," *Image and Vision Computing*, vol. 25, no. 7, pp. 1091 – 1106, 2007.

[33] W. Phillips and M. Shah and N.V. Lobo, "Flame Recognition in Video," *Pattern Recognition Letters*, vol. 23, pp. 319–327, 2002.

[34] T. Chen, P. Wu, and Y. Chiou, "An early fire-detection method based on image processing," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1707–1710, 2004.

[35] C.B. Liu and N. Ahuja, " Vision Based Fire Detection," in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, vol. 4, pp. 134– 137, 2004.

[36] W. Straumann, D. Rizzotti, and N. Schibli, *Method and Device for Detecting Fires Based on Image Analysis.* Boulevard de Grancy 19A, CH-1006 Lausanne, Switzerland: PCTPubn.No:WO02/069292, 2002.

[37] G. Healey, D. Slater, T. Lin, B. Drda, and A. Goedeke, "A system for real-time fire detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15–17, 1993.

[38] Y. Dedeoglu and B.U. Töreyin and U. Gudukbay and A.E. Cetin, "Real-time Fire and Flame Detection in Video," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 669–672, 2005.

[39] B.U. Töreyin and Y. Dedeoglu and A.E. Cetin, "Flame Detection in Video Using Hidden Markov Models," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1230–1233, 2005.

[40] B. U. Töreyin and A. E. Çetin, "Online Detection of Fire in Video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–5, 2007.

[41] T. Çelik and H. Demirel, "Fire Detection in video sequences using a generic color model," *Fire Safety Journal*, vol. 44, pp. 147–158, 2009.

[42] B. Widrow and M.E. Hoff, "Adaptive switching circuits," in *Proceedings of the IRE WESCON (New York Convention Record)*, vol. 4, pp. 96–104, 1960.

[43] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2002.

[44] B.Widrow and S.D.Stearns, *Adaptive Signal Processing*. Prentice Hall, 1985.

[45] B. Widrow and J.M. McCool and M.G. Larimore and C.R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.

[46] B.A. Schnaufer and W.K. Jenkins, " New data-reusing LMS algorithms for improved convergence," in *Proceedings of the Asilomar Conference, Pacific Groves, CA*, vol. 2, pp. 1584–1588, 1993.

[47] N. Dogan, *Orman Yangın Yönetimi ve Yangın Sivilkultürü (In Turkish)*, pp. 143–155. Türkiye Orman Genel Müdürlüğü, (General Directorate of Forestry of Turkey), 2008.

[48] F. Heijden, *Image Based Measurement Systems: Object Recognition and Parameter Estimation*. Wiley, 1996.

[49] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 22, no. 5, pp. 353–362, 1974.

[50] Ö. Gerek and A. Cetin, "Adaptive polyphase subband decomposition structures for image compression," *IEEE Transactions on Image Processing*, vol. 9, pp. 1649–1659, October 2000.

[51] A. Cetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Transactions on Signal Processing*, vol. 42, pp. 194–196, 1994.

[52] N. C. Oza, *Online Ensemble Learning*. PhD thesis, The University of California, Berkeley, CA, Sep 2001.

[53] N. Littlestone and M.K. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 1994.

[54] K. R. T. Aires, A. M. Santana, and A. A. D. Medeiros, "Optical flow using color information: preliminary results," in *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pp. 1607–1611, 2008.

[55] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–466, 1995.

[56] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *International Joint Conferences on Artificial Intelligence*, pp. 674–679, 1981.

[57] N. Thakoor and J. Gao, "Automatic Object Detection in Video Sequences with Camera in Motion," in *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pp. 325–332, September 2004.

[58] J. Lee, K. Rhee, and S. Kim, "Moving target tracking algorithm based on the confidence measure of motion vectors," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 369–372, 2001.

[59] Y. Rosenberg and M. Werman, "Real-Time Object Tracking from a Moving Video Camera: A Software Approach on a PC," in *IEEE Workshop on Applications of Computer Vision*, pp. 238–239, 1998.

[60] B. Qi, M. Ghazal, and A. Amer, "Robust Global Motion Estimation Oriented to Video Object Segmentation," *IEEE Transactions on Image Processing*, vol. 17, pp. 958–967, June 2008.

[61] F. Arnell and L. Petersson, "Fast object segmentation from a moving camera," in *Proceedings of the IEEE on Intelligent Vehicles Symposium*, pp. 136–141, June 2005.

[62] D. Xu and J. An, "Robust global motion estimation method for aerial imagery," *Optical Engineering*, vol. 44, no. 9, p. 090501, 2005.

[63] L. Marcenaro, F. Oberti, and C. Regazzoni, "Change detection methods for automatic scene analysis by using mobile surveillance cameras," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 244–247, 2000.

[64] A. Mittal and D. Huttenlocher, "Scene Modeling for Wide Area Surveillance and Image Synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 160–167, 2000.

[65] R. K. Anandan, R. Kumar, P. An, M. Irani, J. Bergen, and K. Hanna, "Representation of Scenes from Collections of Images," in *Proceedings of the IEEE Workshop on Representations of Visual Scenes*, pp. 10–17, 1995.

[66] F. Dellaert and R. Collins, "Fast Image-Based Tracking by Selective Pixel Integration," in *Proceedings of the ICCV Workshop on Frame-Rate Vision*, pp. 1–22, 1999.

[67] D. Fleet and Y. Weiss, "Optical Flow Estimation," in *Handbook of Mathematical Models in Computer Vision*, ch. 15, pp. 239–259, Springer, 2005.

[68] J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm," tech. rep., Intel Corporation, Microprocessor Research Labs, 2000. KLT implementation in OpenCV.

[69] Y. Wang, Y. Zhang, and J. Ostermann, *Video Processing and Communications*. Prentice Hall, 2001.

[70] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.