

# Video Based Wild Fire Detection at Night

Osman Günay, Kasım Taşdemir, B. Uğur Töreyn, A. Enis Çetin

*Bilkent University, Department of Electrical and Electronics Eng., 06800, Bilkent, Ankara, Turkey,  
Telephone: +90-312-290-2596, Fax : +90-312-266-4192*

---

## Abstract

There has been an increasing interest in the study of video based fire detection algorithms as video based surveillance systems become widely available for indoor and outdoor monitoring applications. A novel method explicitly developed for video based detection of wildfires at night (in the dark) is presented in this paper. The method comprises four sub-algorithms: (i) slow moving video object detection, (ii) bright region detection, (iii) detection of objects exhibiting periodic motion, and (iv) a sub-algorithm interpreting the motion of moving regions in video. Each of these sub-algorithms characterizes an aspect of fire captured at night by a visible range PTZ camera. Individual decisions of the sub-algorithms are combined together using a least-mean-square(LMS) based decision fusion approach, and fire/nofire decision is reached by an active learning method.

*Key words:* Fire detection, least-mean-square methods, active learning, decision fusion, on-line learning, computer vision.

---

*Email addresses:* osman@ee.bilkent.edu.tr (Osman Günay), tasdemir@ee.bilkent.edu.tr (Kasım Taşdemir), bugur@ee.bilkent.edu.tr (B. Uğur Töreyn), cetin@bilkent.edu.tr (A. Enis Çetin)

## 1. Introduction

Forest watch towers are used all around the world to detect wild fires. Average reported fire detection time is five minutes in manned lookout towers in Turkey. Guards have to work 24 hours in remote locations under difficult circumstances. They may get tired, leave the lookout tower, or may fall asleep at night. Therefore, computer vision based video analysis systems capable of producing automatic fire alarms are necessary to reduce the average forest fire detection time at night, as well as during the daytime. Surveillance cameras can be placed on to the watch towers to detect the surrounding forestal area for possible wild fires. They can be also used for monitoring the progress of the fire from remote centers [1].

We recently developed wildfire detection methods using ordinary visible range cameras [2]. In this paper, a computer vision based method for wildfire detection at night is presented. 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.1, a day-time wildfire at an initial stage is shown. This fire was detected by our system in the summer of 2008 [2]. On the other hand smoke is not visible at night but an unusual bright object appears. 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. 2. Even the flame flicker is not visible from long distances. Therefore, one cannot use the flame flicker information in [3] for long distance night-fire detection.

Recently, there has been an increase in the number of publications on computer vision based fire detection [4]-[15]. 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 Fig. 1 and 2 and 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, (iii) detection of objects exhibiting periodic motion, and (iv) a sub-algorithm interpreting the motion of moving regions in video. 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 [16]. 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 participate in the learning process.

The paper is organized as follows: Section 2 describes each one of the four sub-algorithms which make up the compound (main) wildfire detection algorithm. Adaptive active fusion method is described in Section 3. In Section 4, experimental results based on test fires are presented.

## 2. 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 four main sub-algorithms: (i) slow moving object detection in video, (ii) bright region detection, (iii) detection of objects exhibiting periodic motion, and (iv) a sub-algorithm interpreting the motion of moving regions in video, with decision functions,  $D_1(x, n)$ ,  $D_2(x, n)$ ,  $D_3(x, n)$  and  $D_4(x, n)$ , respectively, for each pixel at location  $x$  of every incoming image frame at time step  $n$ .

The decision functions  $D_i$ ,  $i = 1, \dots, M$  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.

### 2.1. Detection of Slow Moving Objects

Video objects at far distances to the camera seem to move slower (px/sec) in comparison to the nearby objects moving at the same speed. Let  $I(x, n)$  represent the intensity value of the pixel at location  $x$  in the  $n^{\text{th}}$  video frame. Assuming the camera is fixed, two background images,  $B^{\text{fast}}(x, n)$  and  $B^{\text{slow}}(x, n)$  corresponding to the scene with different update rates are estimated [17, 18], from the video images  $I(x, n)$ . Initially,  $B^{\text{fast}}(x, 0)$  and  $B^{\text{slow}}(x, 0)$  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:

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

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. Rapidly moving objects can be detected by subtracting the current image from the estimated background image and thresholding the result when  $a$  is close to 1 [19]. 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.

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. (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 [17, 18, 20] 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.

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:

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

where  $0 < T_{low} < T_{high}$  are experimentally determined threshold values. Adaptive thresholding methods that do not require any constants are developed in [19]. However, the threshold update equation increases the computational cost. Our aim is to realize a real-time fire detection system working in an ordinary PC. Furthermore we do not require a binary decision using a threshold as in [19]. Therefore, we developed the scheme described above to reduce the computational cost. 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.

Forest fires at much longer distances ( $> 5$  Km) to the camera seem to move even slower.

Therefore, fire regions at these distances appear neither in  $B^{fast}$  nor  $B^{slow}$  images. This results in lower difference values between background images  $B^{fast}$  and  $B^{slow}$ . In order to have substantial difference values and detect fire at distances further than 5 Km to the camera,  $B^{fast}$  terms in Eq. 2 are replaced by the current image  $I(x, n)$ , because temporary light sources are not significantly visible in the current image  $I(x, n)$ .

## 2.2. 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. 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:

$$D_2(x, n) = \begin{cases} 1 - \frac{255-Y(x,n)}{128}, & \text{if } Y(x, n) > T_I \\ -1, & \text{otherwise} \end{cases} \quad (3)$$

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 [1]. Our test videos are captured in a clear day with low humidity level.

### 2.3. 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:

$$A(x, n) = \begin{cases} 1, & \text{if } D_1(x, n) > 0.8 \text{ and } D_2(x, n) > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where the thresholds 0.8 and 0.5 are determined experimentally 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 [21]. The movement of the labeled regions between frames is also observed using an object tracking algorithm [20]. 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. Two different methods are used for detection of objects exhibiting periodic motion, namely, average magnitude difference function (AMDF) and similarity matrix.

### 2.3.1. Average Magnitude Difference Function Method

AMDF is generally used to detect pitch period of voiced speech signals [22]. For a given sequence of numbers  $s[n]$ , AMDF is calculated as follows:

$$P(l) = \sum_{n=1}^{N-l+1} |s[n+l-1] - s[n]|, \quad l = 1, 2, \dots, N \quad (5)$$

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

In this sub-algorithm  $s[n]$  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.7. If the AMDF of  $s[n]$  is periodic we define  $P_{AMDF}=1$ , otherwise we set  $P_{AMDF}=-1$ .



### 2.3.2. Similarity Matrix

Mean value of each region  $s[n]$ ,  $n = 1, 2, \dots, 50$  can be also used in a similarity matrix based method to check for periodicity. The simplest way to convert the mean sequence into a similarity matrix domain is to use absolute correlation [23]. We calculate the similarity matrix  $M$  as follows:

$$M(k, l) = |s(k) - s(l)|, \quad k = 1, 2, \dots, N \quad l = 1, 2, \dots, N \quad (6)$$

where  $M(k, l)$  is the  $(k, l)$ th component of the similarity matrix. To check for the periodicity of the original mean value sequence  $s[n]$ , the Discrete Fourier Transform (DFT) of each row of the  $M$  matrix is calculated and the results are added together. The resulting sum of DFTs have different characteristics for periodic and non-periodic mean sequences. The plots of DFTs for an actual fire region and a candidate object that belongs to a periodic flashing light source are shown in Fig.4 (a) and (b) respectively.

To determine the periodicity of a sequence given its sum of DFTs obtained from similarity matrix the following method is used:

$$P_{SM} = \begin{cases} 1, & 3\sigma + \mu < \max_{n=1:N}(\text{abs}(F(n))) \\ -1, & \text{otherwise} \end{cases} \quad (7)$$

where  $\sigma$  is the standard deviation and  $\mu$  is the mean of the absolute values of the DFT sequence  $F$ . The decision function for the third sub-algorithm is determined by combining the results of both periodicity detection methods in the following manner:

$$D_3(x, n) = \begin{cases} 1, & P_{AMDF} = 1 \text{ and } P_{SM} = 1 \\ -1, & \text{otherwise} \end{cases} \quad (8)$$

#### 2.4. Interpreting the Motion of Moving Regions in Video

The candidate fire regions should not move outside some predefined bounds to be correctly identified as fire regions in a fixed camera at the initial stages of fire. This sub-algorithm is mainly aimed at reducing false alarms issued by the lights of slow moving cars at night. The decision function will be designed by analyzing the movements of the previously labeled and tracked objects between frames. The objects are tracked for 5 consecutive frames and the resulting object motion is analyzed. The experimental results show that the center of mass of the bounding rectangle of the candidate object should not move more than the length of the diagonal of its bounding box. For this sub-algorithm the decision function is calculated as follows:

$$D_4(x, n) = \begin{cases} 1 - 2\frac{e}{d}, & d \geq e \\ -1, & \text{otherwise} \end{cases} \quad (9)$$

where  $d$  is the diagonal of the bounding rectangle of the candidate object and  $e$  is its displacement after 5 frames, as shown in Fig.5.

### 3. Adaptation of Sub-algorithm Weights

Cameras, once installed, operate at forest watch towers throughout the fire season for about six months. There is usually a security guard in charge of cameras, as well. The guard can supply feed-back to the detection algorithm after the installation of the system. Whenever an alarm is issued, she/he can verify it or reject it. In this way, she/he can participate the learning process of the adaptive algorithm.

As described in the previous section, the main wildfire detection algorithm is composed of four sub-algorithms. Each algorithm has its own decision function. Decision values from sub-algorithms are linearly combined and weights of sub-algorithms are adaptively updated in our

approach. Sub-algorithm weights are updated according to the Least-Mean-Square (LMS) algorithm which is the most widely used adaptive filtering method [24, 25].

Another innovation that we introduced in this paper is that some of 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. Higher the absolute value, the more confident the sub-algorithm.

Let the compound algorithm be composed of  $M$ -many detection algorithms:  $D_1, \dots, D_M$ . 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. In the wildfire detection problem the number of sub-algorithms,  $M=4$  and each pixel at the location  $x$  of incoming image frame is considered as a sample input for every detection algorithm.

Let  $\mathbf{D}(x, n) = [D_1(x, n) \dots D_M(x, n)]^T$ , 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  $\mathbf{w}(n) = [w_1(n) \dots w_M(n)]^T$  be the current weight vector.

We define

$$\hat{y}(x, n) = \mathbf{D}^T(x, n)\mathbf{w}(n) = \sum_i w_i(n)D_i(x, n) \quad (10)$$

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):

$$\min_{w_i} E[(y(x, n) - \hat{y}(x, n))^2], \quad i = 1, \dots, M \quad (11)$$

where  $E$  represents the expectation operator. Taking the derivative with respect to weights:

$$\frac{\partial E}{\partial w_i} = -2E[(y(x, n) - \hat{y}(x, n))D_i(x, n)] = -2E[e(x, n)D_i(x, n)], \quad i = 1, \dots, M \quad (12)$$

and setting the result to zero:

$$-2E[e(x, n)D_i(x, n)] = 0, \quad i = 1, \dots, M \quad (13)$$

a set of  $M$  equations is obtained. The solution of this set of equations is called the Wiener solution [24, 25]. Unfortunately, the solution requires the computation of cross-correlation terms in Eq. (13). The gradient in Eq. (12) can be used in a steepest descent algorithm to obtain an iterative solution to the minimization problem in Eq. (11) as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda E[e(x, n)\mathbf{D}(x, n)] \quad (14)$$

where  $\lambda$  is a step size. In the well-known LMS algorithm, the ensemble average  $E[e(x, n)\mathbf{D}(x, n)]$  is estimated using the instantaneous value  $e(x, n)\mathbf{D}(x, n)$  or it can be estimated from previously processed pixels as follows:

$$\hat{e}(x, n)\hat{\mathbf{D}}(x, n) = \frac{1}{L} \sum_{x,n} e(x, n)\mathbf{D}(x, n) \quad (15)$$

where  $L$  is the number of previously processed pixels. The LMS algorithm is derived by noting that the quantity in Eq. (14) is not available but its instantaneous value is easily computable, and hence the expectation is simply replaced by its instantaneous value [26]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda e(x, n)\mathbf{D}(x, n) \quad (16)$$

Eq. (16) 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. (16). Note that, oracle does not assign her/his decision to each and every pixel one by one. She/he actually selects a

window on the image frame and assigns a “1” or “−1” to the selected window.

Convergence of the LMS algorithm can be analyzed based on the MSE surface:

$$E[e^2(x, n)] = P_y(x, n) - 2\mathbf{w}^T \mathbf{p} - \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (17)$$

where  $P_y = E[y^2(x, n)]$ ,  $\mathbf{p} = E[y(x, n)\mathbf{D}(x, n)]$ , and  $\mathbf{R} = E[\mathbf{D}(x, n)\mathbf{D}^T(x, n)]$ , with the assumption that  $y(x, n)$  and  $\mathbf{D}(x, n)$  are wide-sense-stationary random processes. The MSE surface is a function of the weight vector  $\mathbf{w}$ . Since  $E[e^2(x, n)]$  is a quadratic function of  $\mathbf{w}$ , it has a single global minimum and no local minima. Therefore, the steepest descent algorithm of Eq. (14) and (16) is guaranteed to converge to the Wiener solution,  $w^*$  [26] with the following condition on the step size  $\lambda$  [25]:

$$0 < \lambda < \frac{1}{\alpha_{max}} \quad (18)$$

where  $\alpha_{max}$  is the largest eigenvalue of  $\mathbf{R}$ .

In Eq. (16), the step size  $\lambda$  can be replaced by

$$\frac{\mu}{\|\mathbf{D}(x, n)\|^2} \quad (19)$$

as in the normalized LMS algorithm, which leads to:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(x, n)}{\|\mathbf{D}(x, n)\|^2} \mathbf{D}(x, n) \quad (20)$$

where the  $\mu$  is an update parameter and the normalized LMS algorithm converges for  $0 < \mu < 2$  to the Wiener solution,  $w^*$  with the wide-sense-stationarity assumption. 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  $\mu$  lies between 0 and 2 [27].

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, in all of the wild fire video recordings that we have. 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. (20) and negative weights are reset to zero complying with the non-negative weight constraint.

Unfortunately, the wide-sense-stationarity assumption is not a valid assumption in natural images, as in many signal processing applications. Nevertheless, the LMS algorithm is successfully used in many telecommunication and signal processing problems. Wide-sense-stationarity assumption may be valid in some parts of a sequence in which there are no spatial edges and temporal changes.

The main advantage of the LMS algorithm compared to other related methods, such as the weighted majority algorithm [28], is the controlled feedback mechanism based on the error term. Weights of the algorithms producing incorrect (correct) decision is reduced (increased) according to Eq. (20) in a controlled and fast manner. In weighted majority algorithm, conflicting weights with the oracle are simply reduced by a factor of two [28, 29]. Another advantage of the LMS algorithm is that it does not assume any specific probability distribution about the data.

#### **4. 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. 6.

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 three other methods: one uses only slow moving objects to detect fire, one uses only intensity information, the other uses both slow moving objects and intensity information. The results are summarized in Table 1. Fig.7 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. Fig.8 and 9 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.

A set of video clips containing various artificial light sources is used to generate Table 2. The snapshots from four of the videos are shown in Fig. 10. 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 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, other methods produce 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.

## **5. Conclusion**

An automatic wildfire detection algorithm that operates at night conditions using an LMS based active learning capability is developed. The compound algorithm comprises four sub-algorithms which produce their individual decision values for night fires. Each algorithm is designed to characterize an aspect of night fires. 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 wildfire detection system working on a standard PC. The LMS based adaptive decision fusion strategy takes into account the feedback from guards of forest watch towers. Experimental results show that the learning duration is decreased with the proposed active learning scheme. It is also observed that false alarm rate of the proposed LMS based method is the lowest in our data set, compared to the methods using only intensity information and slow object detection.

## **Acknowledgment**

This work was supported in part by the Scientific and Technical Research Council of Turkey, TUBITAK, with grant no. 106G126 and 105E191, and in part by European Commission 6th Framework Program with grant number FP6-507752 (MUSCLE Network of Excellence Project).



## References

- [1] N. Dogan, *Orman Yangin Yonetimi ve Yangin Sivilkulturu (In Turkish)*, pp. 143–155, Orman Genel Mudurlugu, (General Directorate of Forestry), 2008.
- [2] B.U. Töreyn 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)*, 2009.
- [3] B.U. Töreyn, Y. Dedeoglu, 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.
- [4] W. Phillips, M. Shah, and N.V. Lobo, “Flame recognition in video,” *Pattern Recognition Letters*, vol. 23, pp. 319–327, 2002.
- [5] 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)*, 2004, pp. 1707–1710.
- [6] C.B. Liu and N. Ahuja, “Vision based fire detection,” in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2004, vol. 4.
- [7] W. Straumann, D. Rizzotti, and N. Schibli, *Method and Device for Detecting Fires Based on Image Analysis*, European Patent EP 1,364,351, 2002.
- [8] G. Healey, D. Slater, T. Lin, B. Drda, and A.D. Goedeke, “A system for real-time fire detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1993, pp. 15–17.
- [9] M. Thuillard, “A new flame detector using the latest research on flames and fuzzy-wavelet algorithms,” *Fire Safety Journal*, vol. 37, pp. 371–380, 2002.
- [10] Y. Takatoshi, *Fire Detecting Device*, Japanese Patent 11,144,167, 1999.
- [11] Z. Xiong, R. Caballero, H. Wang, A.M. Finn, M.A. Lelic, and P.-Y. Peng, “Video based Smoke Detection: Possibilities, Techniques, and Challenges, available at [http://vision.ai.uiuc.edu/~wanghc/papers/smoke\\_detection.pdf](http://vision.ai.uiuc.edu/~wanghc/papers/smoke_detection.pdf),” Accessed at December 2008.
- [12] Y. Dedeoglu, B.U. Töreyn, 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)*, 2005, pp. 669–672.
- [13] B.U. Töreyn, 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)*, 2005, pp. 1230–1233.

- [14] B. U. Töreyn and A. E. Çetin, "Online detection of fire in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–5.
- [15] T. Çelik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Safety Journal*, vol. 44, pp. 147–158, 2009.
- [16] B. Widrow and M.E. Hoff, "Adaptive switching circuits," in *Proceedings of the IRE WESCON (New York Convention Record)*, 1960, vol. 4, pp. 96–104.
- [17] A.E. Cetin, M.B. Akhan, B.U. Toreyin, and A.Aksay, "Characterization of motion of moving objects in video," *US Patent No. 20040223652, pending*, 2004.
- [18] 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.
- [19] R.T. Collins, A.J. Lipton, and T. Kanade, "A system for video surveillance and monitoring," in *Proceedings of the 8-th International Topical Meeting on Robotics and Remote Systems*. April 1999, American Nuclear Society.
- [20] B.U. Töreyn, "Moving object detection and tracking in wavelet compressed video," M.S. thesis, Bilkent University, Department of Electrical and Electronics Engineering, Ankara, Turkey, 2003.
- [21] F. Heijden, *Image Based Measurement Systems: Object Recognition and Parameter Estimation*, Wiley, 1996.
- [22] M J Ross et al., "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust., Speech Signal Processing*, vol. 22, no. 5, pp. 353–362, 1974.
- [23] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Trans. Pattern An. and Machine Intelligence*, vol. 22, no. 8, pp. 781–796, 2000.
- [24] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 2002.
- [25] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Prentice Hall, 1985.
- [26] B.A. Schnaufer and W.K. Jenkins, "New data-reusing LMS algorithms for improved convergence," in *Proceedings of the Asilomar Conference, Pacific Groves, CA*, 1993, pp. 1584–1588.
- [27] B. Widrow, J.M. McCool, M.G. Larimore, and C.R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proceedings of the IEEE*, vol. 64(8), pp. 1151–1162, 1976.
- [28] N. Littlestone and M.K. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212261, 1994.

- [29] N.C. Oza, *Online Ensemble Learning*, Ph.D. thesis, Electrical Engineering and Computer Sciences, University of California, September 2001.

## Figure Captions

**Fig. 1:** A Snapshot of a typical forest fire smoke at the initial stages captured by a forest watch tower which is 3 km away from the fire (fire region is marked with an arrow).

**Fig. 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).

**Fig. 3:** AMDF graphs for (a) periodic flashing light and (b) non-periodic bright region in video.

**Fig. 4:** Sum of DFT graphs for (a) periodic flashing light source and (b) non-periodic bright region in video.

**Fig. 5:** The displacement of candidate fire region after 5 frames.

**Fig. 6:** Samsung analog camera mounted at the watch tower.

**Fig. 7:** Correct alarm for a fire at night and elimination of fire-truck headlights.

**Fig. 8:** Detection results on an actual forest fire at night.

**Fig. 9:** Detection results on an actual forest fire at night.

**Fig. 10:** Snapshots from videos that we 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.

| Video Seq. | Range<br>(km) | Frame Rate<br>(fps) | Frame number of first alarm |          |                |                 |
|------------|---------------|---------------------|-----------------------------|----------|----------------|-----------------|
|            |               |                     | LMS Based                   | SMO Only | Intensity Only | SMO + Intensity |
| V1         | 5             | 25                  | 221=10 sec                  | 276      | 64             | 241             |
| V2         | 6             | 25                  | 100=4 sec                   | 121      | 12             | 115             |
| V3         | 6             | 25                  | 216=8 sec                   | 726      | 8              | 730             |
| V4         | 7             | 25                  | 151=6 sec                   | 751      | 15             | 724             |
| V5         | 1             | 25                  | 83=4 sec                    | 153      | 12             | 184             |
| V6         | 0.5           | 25                  | 214=8 sec                   | 140      | 8              | 204             |
| V7         | 0.1           | 30                  | 59=2 sec                    | 229      | 5              | 241             |
| V8         | 0.1           | 30                  | 74=3 sec                    | 181      | 6              | 194             |
| V9         | 0.1           | 30                  | 56=2 sec                    | 209      | 7              | 211             |

**Table 1:** Three different methods (LMS Based, Only Slow Moving Objects (SMO) based, Only intensity based, SMO and intensity 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. | Frame Rate<br>(fps) | Duration<br>(frames) | Number of false alarms |          |           |                |
|------------|---------------------|----------------------|------------------------|----------|-----------|----------------|
|            |                     |                      | LMS Based              | SMO Only | Intensity | SMO+ Intensity |
| V10        | 15                  | 3000                 | 1                      | 11       | 24        | 4              |
| V11        | 15                  | 1000                 | 0                      | 8        | 17        | 2              |
| V12        | 15                  | 2000                 | 0                      | 12       | 16        | 3              |
| V13        | 15                  | 1000                 | 0                      | 2        | 14        | 2              |
| V14        | 10                  | 1900                 | 0                      | 2        | 12        | 1              |
| V15        | 10                  | 1200                 | 0                      | 8        | 10        | 5              |

**Table 2:** Three different methods (LMS Based, Only Slow Moving Objects (SMO) based, Only intensity based, SMO and intensity based) are compared in terms of the number of false alarms issued to video sequences that do not contain fire.

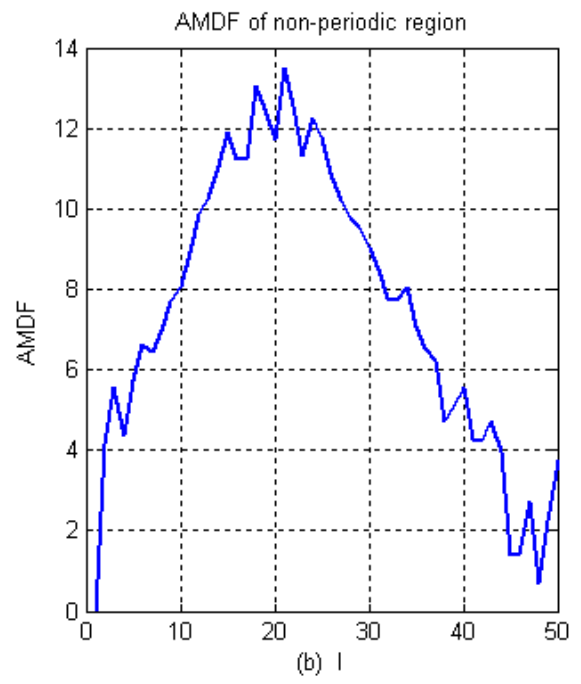
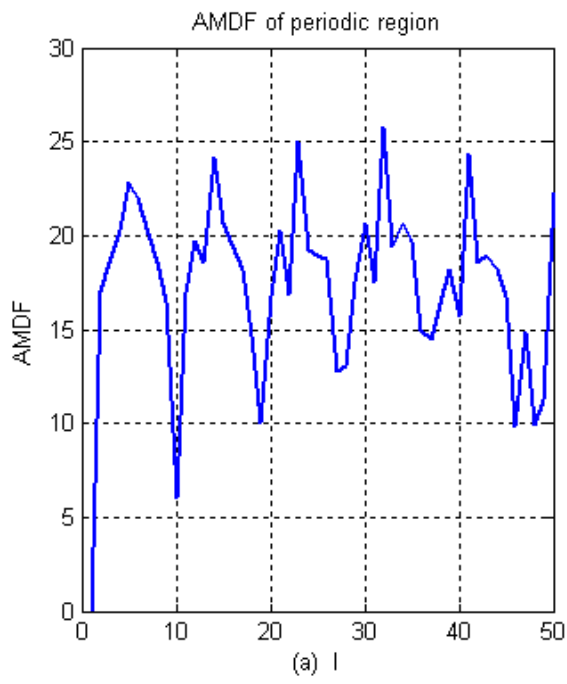


**Figure 1**

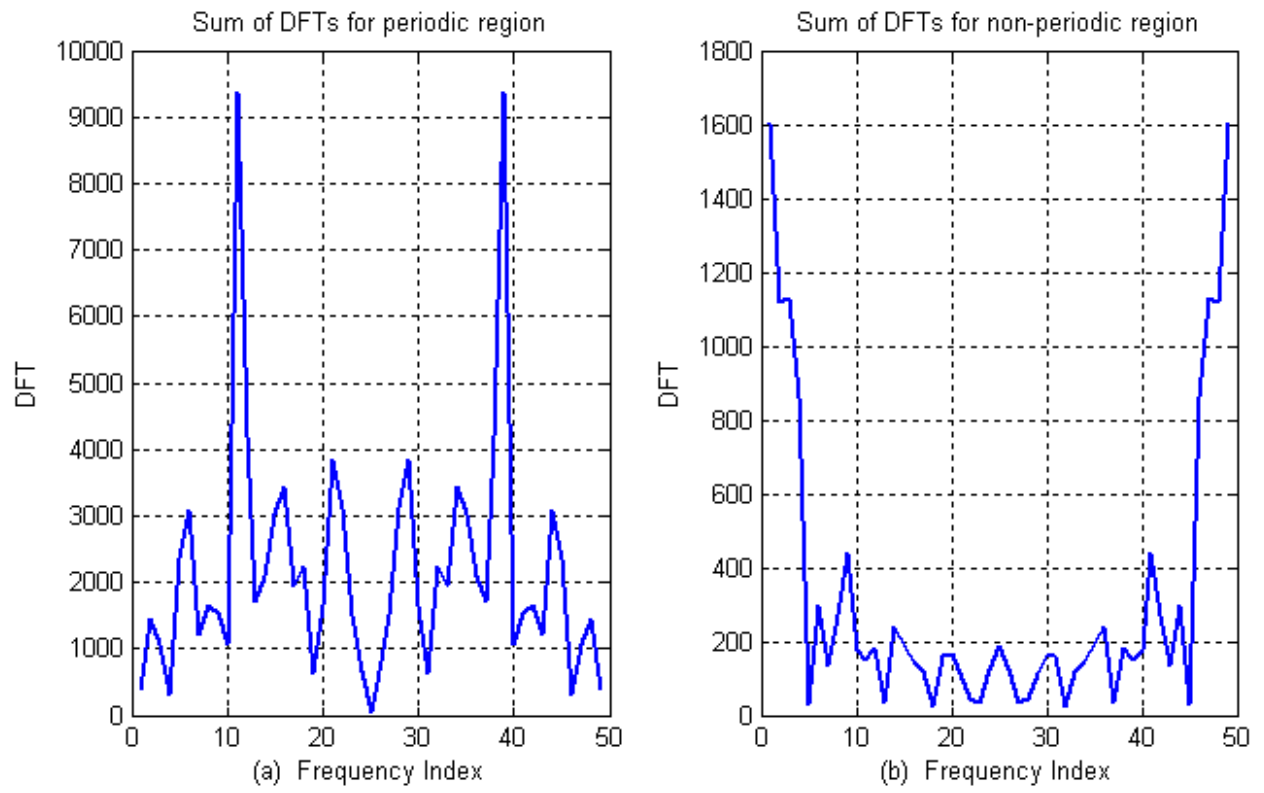


**Figure 2**





**Figure 3**



**Figure 4**

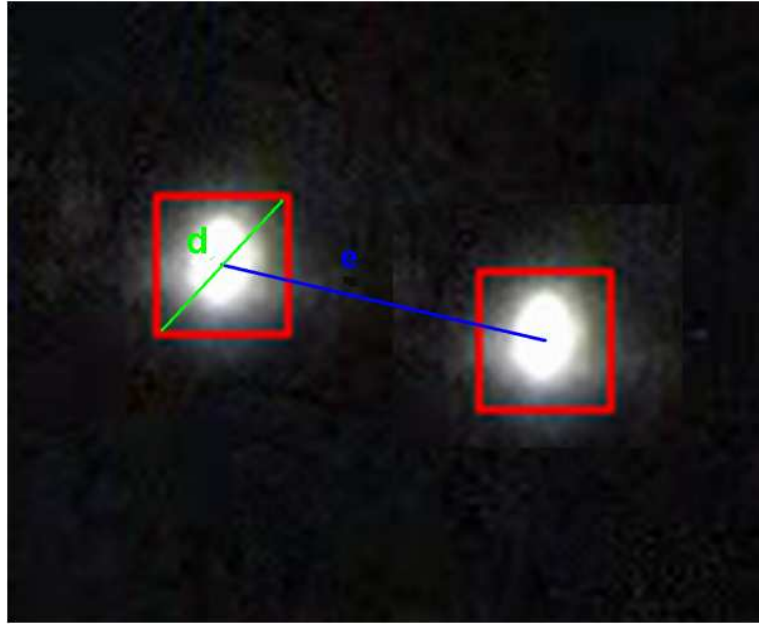


Figure 5



**Figure 6**



Figure 7



**Figure 8**



Figure 9



(a)



(b)



(c)



(d)

**Figure 10**