Building and Environment 45 (2010) 1113–1122

Contents lists available at ScienceDirect

Building and Environment

journal homepage: www.elsevier.com/locate/buildenv

Multi-feature fusion based fast video flame detection

Juan Chen [a,c], Yaping He [b], Jian Wang [a,*]

[a] State Key Laboratory of Fire Science, University of Science and Technology of China, Hefei 230027, PR China
[b] School of Engineering, University of Western Sydney, Australia
[c] Zhejiang Key Laboratory of Safety Engineering and Technology Research, Zhejiang Institute of Safety Science and Technology, Hangzhou 310012, PR China

ARTICLE INFO

Article history:
Received 26 June 2009
Received in revised form
22 October 2009
Accepted 25 October 2009

Keywords:
Motion detection
Gaussian mixture model
Color mapping
Flickering analysis

ABSTRACT

A video flame detection method based on the multi-feature fusion is presented in this paper. The temporal and spatial characteristics of flames, such as ordinary flame movement and color clues, a flame flickering detection algorithm is incorporated into the scheme to detect fires in color video sequences. An improved Gaussian mixture model method is firstly adopted to extract moving foreground objects from the still background of detection scenes; secondly, detected moving objects are then categorized into candidate and non-candidate flame regions by using a flame color filtering algorithm; finally, a flame flicker identification algorithm based on statistical frequency counting is used to distinguish true flames from fire-like objects in video images. Testing results show that the proposed algorithms are effective, robust and efficient. The processing rate of the flame detection method can achieve 24 fps with image size of 320 × 240 pixels on a PC with an AMD 2.04 GHz processor.

1. Introduction

Fire detection is an important part of building fire safety systems. Early and reliable detection and warning are crucial for building occupants to evacuate to safety during fire emergencies [1,2], other fire safety measures, such as fire suppression system and smoke control unit, may also rely on fire detections systems to activate [3]. The conventional fire detection technology is usually seen in the forms of smoke or heat detectors. These detectors are mainly based on sampling of soot particles, product gas, temperature or radiant heat [4]. Their applications are, as a result, confined to enclosure fire detections. For large space and out-door scenes this kind of technology suffers a dramatic reduction in detection capability because the variations in detectable parameters become so weak when the detection distance becomes large. A new technology of video fire detection has attracted considerable attention with the development of CCTV (Closed Circuit Television) surveillance system in recent years [6–10,12–14,17–22]. This new technology can provide more reliable information and be more cost-effective. The video fire detection technology uses CCD cameras to capture images of the observed scene, which provides abundant and intuitive information for fire detection using image processing algorithms. This technology can be easily incorporated into the intelligent building systems [5]. Furthermore, this video based fire detect system can be incorporated in the existing surveillance systems with a minimal addition to total fire protection cost.

Early methods of visual fire detection rely extensively on spectra analysis using expensive spectroscopy equipments [6]. The cost of such equipments hinders the wide application of this technology. To by-pass the spectroscopy and utilize the ordinary black-and-white or color surveillance cameras, effective and efficient digital image processing algorithms are required. Healey et al. [7] presented a pioneering vision-based fire detection technique which makes use of the color information to distinguish fire or non-fire regions in the captured images. Cheng et al. [8] demonstrated the principle of the visual fire detection with the aid of liquid crystal light valve to adjust the valid integral range of CCD cameras, and used the high intensity of flame image to extract flame pixels.

Methods simply based on criteria of flame color or intensity have low reliability and may produce false alarms in case of the presence of non-fire objects which resemble the color of fires, e.g., a reflection of the Sun, a torch or a moving flag. To improve the robustness and reliability of the detection algorithm, the analysis of ordinary motion feature is incorporated to identify other fire aliases in addition to chromatic features. Yamagishi and Yamaguchi [9] introduced a neural network based flame detection method by adopting the spatiotemporal features of the flame contour as inputs. The contour of the flame is extracted in their method on the basis of statistical criteria from images in HSV color space. The contour is then mapped into a polar coordinate to convert the image sequences into a fluctuating time series. This time series signal is transferred into Fourier domain to obtain the frequency

* Corresponding author. Tel.: +86 5513606463; fax: +86 5513601669.
E-mail address: wangj@ustc.edu.cn (J. Wang).

0360-1323/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.buildenv.2009.10.017

spectrum which is at last passed to a trained neural network to identify fire flames. These procedures give good results but the computational complexity is too high to be used for real-time sequences. Phillips et al. [6] used the color predicate information and temporal variation to recognize flame in video sequences. A manually labeled training set is employed to create a look-up table in the preprocessing phase. By using the look-up table, fire colored regions are extracted in the video image first. Then, a temporal variation of pixels based on a small subset of images is also used to determine which parts of these pixels are actually fire pixels. The shape of a fire region is also represented in terms of the spatial frequency content of the region contour using its Fourier coefficients by Liu and Ahuja [10]. The temporal changes in these coefficients are treated as the temporal signatures of the fire region. However, the problem is that spatial quantization errors for small regions are likely to introduce considerable noise in the Fourier domain. To avoid this problem, a threshold is introduced to eliminate regions of small size and also exclude elongated narrow regions. Consequently, the algorithm described in ref. [10] is limited to detection of developed flames that occupy significant portions of video image frames. Phillips et al.'s [6] method and a motion detection method by Wren et al. [11] were merged and improved by Celik et al. [12]. The resulting statistical color model was reported to have high reliability and processing speed. However, the demonstrated applications were limited to images of low resolution (176×144).

An alternative method can be found in Chen et al. [13]. In this method, the existence of fire is detected through analyzing flame color characteristics first and then consolidating the results by examining the status of the flame spread. In the analysis of flame color, the RGB model is combined with the HSI model. The fire pixels in an image are simply deduced by examining the chromatic features, namely the intensity and saturation of the R component. The identified fire region in the images is then further verified by calculating the number of total fire-pixel changes from every two consecutive frames. This algorithm also has a poor reliability. To detect car fires in expressway tunnels, Ono et al. [14] used features of experimental fires in images which have high red intensity. In this method, the potential flame zones are identified by means of calculating red intensity differences between the background image and each new coming frame. A set of fire feature parameters is extracted from the potential fire zones and then fed into a predesigned artificial neural network (ANN) to confirm and complete the detection. Wider applications of this method have not been reported in the literature.

One of the characteristics of fire flames is the flicker, or the flickering motion of the flames around the edges of the flaming region [15,16]. To improve the reliability of detection and reduce the false alarms due to ordinary moving objects with fire-like colors, some periodicity analyses are employed in detection algorithm. Toreyin et al. [17] and Dedeoglu et al. [18] employed temporal wavelet analysis to extract the quasi-periodic behavior of flickers and use spatial wavelet analysis to capture spatial color variations of flames. Moreover, they proposed a method using three-state hidden Markov models to model flame flicker in visible video images [19]. The two properties of fires, flickering and maximal luminance, have been used in the method. Marbach et al. [20] proposed to realize automatic real-time video fire detection. A temporal accumulation of time derivative images was used to select a best candidate fire region. From this candidate region, characteristic fire features are extracted and combined to compute the fire indicator, whose pattern are analyzed to determine whether fires are present or not. Schultze et al. [21] used not only sonagram and spectrogram to investigate the flickering property, but also a simplified flow analysis to trace the characteristic upward movement of turbulent open flames. This method represents a fusion of flickering and flow movement analyses. It delivers interesting and promising results and may be applied with ordinary video cameras for video fire detection. An example of the application of video surveillance flame detection in conjunction with a fire suppression system can be found in Chen et al. [22], though the flame identification algorithm was not fully described in this particular publication.

In the current study, the focus is given to a multi-feature fusion based flame detection method to improve the speed and the reliability of video fire detection. Most of the algorithms discussed above face some typical problems in robustness and flexibility, as well as efficiency. Typically, the challenges are imposed by the changes in lighting conditions, image quality, scene complexity and processor performance. These challenges are addressed by a multi-feature fusion method in the current study. In addition to the ordinary flame motion and color analysis, flame flickering is also analyzed in this method to enhance the robustness and efficiency and to reduce both false alarm and failure rates.

Full descriptions of the method are presented in Section "Method" of this paper. The applications of the method to a number of recorded fire and non-fire scenes and the results are presented in Section "Testing and analysis" followed by "Concluding remarks".

2. Method

The proposed method consists of three major steps:

(1) Motion detection and elimination of static background;
(2) Color discrimination and elimination of non-fire colored moving objects;
(3) Flickering test and elimination of non-flickering fire colored objects.

The first step is achieved by applying an improved fast Gaussian mixture model (GMM) to analyze the RGB color space of image pixels. The second step is achieved by further filtering non-fire characteristic colors and the third by detecting the flickering motions in the flame image. Detailed discussions of these steps are presented in the following subsections.

2.1. Motion detection

Natural fire flames are seen as dynamic objects in the video images due to the reason that the majority of objects in images of a fixed surveillance camera are static in observed scenes. This feature can be utilized to distinguish a true fire flame from other fire-like objects or sources of fire-like images. In order to eliminate the disturbance of the fire-like background objects (stationary scene), the first step is to use a Gaussian mixture model (GMM) [23] to distinguish the foreground moving objects from the still background. The Gaussian mixture model is an effective and simple adaptive background modeling algorithm which tracks the history of each pixel with a mixture of K Gaussian distributions respectively in a video sequence. These K distributions are classified into background and foreground subsets. A pixel that fits its distributions of background subset is considered as background, otherwise it is considered as foreground. This method can solve the problem of multi-modal background and adapt the parameters of each mixture model automatically with gradual changes of illumination environment. However, the Gaussian mixture model suffers from a slow speed of calculation because it is based on pixel-level which contains certain degree of space-time redundancy. In order to achieve a real-time detection of video moving objects, an improved algorithm is introduced.

2.1.1. Gaussian mixture model

The color feature vector for a given pixel with coordinates (x, y) of an image at any time t in RGB color space can be expressed as:

$$\overrightarrow{X}_{xy,t} = \left[R_{xy,t}, G_{xy,t}, B_{xy,t}\right]^T \quad (1)$$

A series of observations of the color vector for a pixel (x, y) along a time sequence are recorded as $\{\overrightarrow{X}_{xy,1}, \ldots, \overrightarrow{X}_{xy,t}\}$, which can be seen as a stochastic process independent of other pixels, and has its own probability density function. A new observation from a new image is considered to be a background pixel if it is well described by this very density function. However, pixel values often have complex distributions, making the efficient estimation and adaptation of probability density functions a difficult problem. Hence, a pixel in time series is modeled by a mixture of K Gaussian distributions [23], where K is the number of Gaussian distributions. With a larger K, the system can model more complex scenes at the cost of substantially increased computational expenses. Taking the calculation complexity into account, K generally ranges from 3 to 5 [23]. The probability of observing a given color vector for a pixel is then expressed as

$$P(\overrightarrow{X}_{xy,t}) = \sum_{k=1}^{K} w_{xy,k,t}\, \eta\left(\overrightarrow{X}_{xy,t}, \overrightarrow{u}_{xy,k,t}, \Sigma_{xy,k,t}\right) \quad (2)$$

where $w_{xy,k,t}$, is the weight, which is regarded as the probability that an observation value belongs to the *kth* Gaussian component (the sum of these K weights is one), $\overrightarrow{u}_{xy,k,t}$ and $\Sigma_{xy,\,k,t}$ are mean vector and covariance matrix of the *kth* Gaussian distribution in the mixture at time t, respectively. η is the Gaussian probability density function, defined as:

$$\eta\left(\overrightarrow{X}_{xy,t}, \overrightarrow{u}_{xy,k,t}, \Sigma_{xy,k,t}\right) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_{xy,k,t}|^{\frac{1}{2}}} e^{-1/2\left(\overrightarrow{X}_{xy,t}-\overrightarrow{u}_{xy,k,t}\right)^T \sum_{xy,k,t}^{-1}\left(\overrightarrow{X}_{xy,t}-\overrightarrow{u}_{xy,k,t}\right)} \quad (3)$$

where n is the dimension of color feature vector, which as a result equals to 3.

To avoid time-consuming matrix calculation, the red, green, and blue color channels of the each pixel are assumed to be independent and have the same variance, which means

$$\Sigma_{xy,k,t} = \mathrm{diag}\left[\sigma_R^2, \sigma_G^2, \sigma_B^2\right] = \sigma_{xy,k,t}^2 I \quad (4)$$

where I is the identity matrix and σ is the standard deviation.

It is known that a larger weight of Gaussian distribution, w, means more observations matching the Gaussian distribution over a period of time; a smaller standard deviation σ stands for a more stable Gaussian distribution. Therefore, the K Gaussian distributions describing the history of a pixel are ordered by the descending value of w/σ first to determine its status, i.e., whether the pixel belongs to the background subset or the foreground subset. Typically, the moving objects will be represented by some distributions with small weights. Therefore, the first B distributions whose weighted sum is more than a threshold value T are chosen to approximate the background subset, while the remaining K–B distributions are the foreground subset, where B is estimated as:

$$B_{xy,t} = \arg\min_B \left[\sum_{k=1}^{B} w_{xy,k,t} > T\right] \quad (5)$$

The threshold T is a minimum portion which should be accounted for the background. If T is high, the background model is composed of a multi-modal distribution and could represent complex background information.

For newly observation value of a pixel, it may match one of the major components of the K distributions. The matching Gaussian distribution can be determined if the distance between $\overrightarrow{X}_{xy,t}$ and $\overrightarrow{u}_{xy,k,t}$ is less than a threshold of λ multiplied by the standard deviations. Every new observation of a given pixel is checked against the existing K Gaussian components according to the following equation to determine which of the previous distribution is satisfied firstly.

$$M_{xy,k,t} = \begin{cases} 1, & \text{if} \left|\overrightarrow{X}_{xy,t} - \overrightarrow{u}_{xy,k,t}\right| < \lambda\sigma_{xy,k,t} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $M_{xy,\,k,t}$ is a matching function of the *kth* Gaussian distribution at time t to update parameters in the following step; the moderation factor λ is a positive number and is usually set in the range of 2.5–3 [24] to achieve a confidence level higher than 0.95. If the observation value $\overrightarrow{X}_{xy,k,t}$ matches with one of these B background distributions, then the pixel (x, y) is marked as a background pixel for the present frame, otherwise it is considered to be a foreground pixel.

In practice, the sunlight change during daytime, weather conditions and switching light could change illuminations in the scene. Therefore, it is necessary to update parameters of GMM to adapt for the changed illumination or other natural effects. Thus, according to the current match result for each frame, the corresponding Gaussian mixture model for each pixel will be updated depending on the current observation value. The method for updating the first matched distribution is as follows: at time t, if there is a match in a mixture of K Gaussian distributions, the mean vector and covariance are adjusted according to the current observation $\overrightarrow{X}_{xy,t}$, but remain the same for non-match distributions. Meanwhile, the weights of the K distributions of time t are adjusted by the current matching function. A simple adaptation of corresponding pixel's model parameters can be formulated as in the following equation:

$$\begin{cases} w_{xy,k,t+1} = (1-\alpha)w_{xy,k,t} + \alpha M_{xy,k,t} \\ \overrightarrow{u}_{xy,k,t+1} = \left(1-\rho_{xy,k,t}\right)\overrightarrow{u}_{xy,k,t} + \rho_{xy,k,t}\overrightarrow{X}_{xy,t} \\ \sigma_{xy,k,t+1}^2 = \left(1-\rho_{xy,k,t}\right)\sigma_{xy,k,t}^2 + \rho_{xy,k,t}\left(\overrightarrow{X}_{xy,t} - \overrightarrow{u}_{xy,k,t+1}\right)^T\left(\overrightarrow{X}_{xy,t} - \overrightarrow{u}_{xy,k,t+1}\right) \\ \rho_{xy,k,t} = \alpha\eta\left(X_{xy,t}, u_{xy,k,t}, \sigma_{xy,k,t}^2 I\right) \end{cases} \quad (7)$$

where α refers to the learning rate of the background updating, with a values between 0 and 1. If there is no match in the K distributions for the current observation $\overrightarrow{X}_{xy,t}$, the Gaussian component with small w/σ is replaced by a new distribution with its mean vector set by $\overrightarrow{X}_{xy,t}$, and with an initially large variance and small prior weight.

2.1.2. Fast algorithm for Gaussian mixture model

The background subtraction method discussed above is based on establishing a GMM for each pixel, which has the advantages of flexibility and accuracy. However, as large portions of image frames are usually occupied by still background, the information obtained from applying GMM to the background pixels is highly redundant. The pixels which are associated with potential flame images are well correlated in spatial and temporal domains and these correlations can be utilized to reduce the amount of data processing task. In other words, it is unnecessary to apply GMM to every pixel in a frame.

Based on the correlation of pixels in temporal and spatial domains, the pixels in a frame can be classified into active pixels and inactive pixels and only the active pixels instead of all the pixels in the image are processed [25,26]. For example, Fig. 1 represents an image with a resolution of 6 × 6, where the grey cells represent active pixels and the white inactive ones. If only the history of each active pixel is modeled by a GMM respectively, then the total number of GMM is reduced by one half.

In the current study, the active and inactive pixels are prescribed in alternate fashion along the rows and columns of the pixel matrix as shown in Fig. 1. The background or foreground statuses of all active pixels are judged according to the method described in Section "Gaussian Mixture Model". The statuses of all inactive pixels are determined according to the statuses of their four neighboring active pixels. For instance, the pixel m is an inactive pixel in Fig. 1, which will be classified as a background or foreground pixel in accordance with its four adjacent pixels (A, B, C and D). If there are more than two pixels judged as background in A, B, C and D, the inactive pixel m is then judged as background, and vice versa. If there are equal number of foreground and background neighboring pixels, the status of m is judged to be background if the condition as expressed by Eq. (8) is satisfied or otherwise foreground.

$$\left|\vec{X}_{m,t} - \vec{u}_{m,t}\right| < T_m \tag{8}$$

where T_m is a threshold, $\vec{X}_{m,t}$ is the observation value of pixel m at time t and $\vec{u}_{m,t}$ is the average of the mean vectors of the first Gaussian distribution descended by w/σ for the four neighboring active pixels A, B, C and D, i.e.,

$$\vec{u}_{m,t} = (\vec{u}_{\mathrm{A},t} + \vec{u}_{\mathrm{B},t} + \vec{u}_{\mathrm{C},t} + \vec{u}_{\mathrm{D},t})/4 \tag{9}$$

Redundant information also exists in the temporal domain and this redundancy is carried forward during the updating of GMM. The following algorithm is introduced to minimize this redundancy. Assume the observation $\vec{X}_{A,t}$ of active pixel A matches its Kth Gaussian distribution at time t; if the distance between $\vec{X}_{A,t}$ and its mean $\vec{u}_{A,k,t}$ satisfies the following condition:

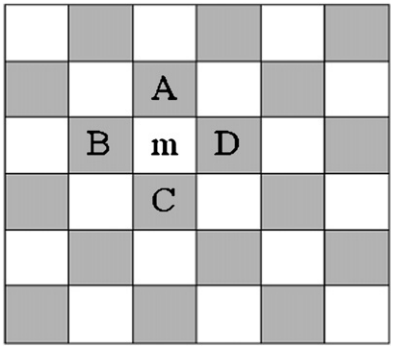

Fig. 1. Active and inactive pixels in an image.

$$\left|\vec{X}_{\mathrm{A},t} - \vec{u}_{\mathrm{A},k,t}\right| < \lambda_1 \sigma_{\mathrm{A},k,t} \tag{10}$$

where the moderation factor λ_1 is a positive number smaller than λ, then the mean and the standard deviation of the Gaussian distribution are updated using the following equations instead of Eq. (7):

$$\vec{u}_{\mathrm{A},k,t+1} = \vec{u}_{\mathrm{A},k,t} \tag{11}$$

and

$$\sigma^2_{\mathrm{A},k,t+1} = (1 - \rho_{\mathrm{A},k,t})\sigma^2_{\mathrm{A},k,t} \tag{12}$$

The above discussed algorithm removed redundant information in both temporal and spatial domains, and greatly improves the efficiency of motion detection over the existing method employed by Phillips et al. [6] and Celik et al. [12].

2.1.3. Motion detection

In summary, the improved Gaussian mixture model is established for each active pixel in the RGB color space, whereby each active pixel is classified as part of the foreground moving object or the background static object by calculating the corresponding model match, while inactive pixels are classified as background or foreground according to the classification of their neighboring active pixels. The result of background subtraction is a binary map (BM) corresponding to the following formula:

$$\mathrm{BM}(x,y,t) = \begin{cases} 0, & \text{if } \vec{X}_{xy,t} \subseteq \text{background} \\ 1, & \text{if } \vec{X}_{xy,t} \subseteq \text{foreground} \end{cases} \tag{13}$$

where $\vec{X}_{xy,t}$ is the color feature vector in coordinates (x, y) at time t. See Eq. (1).

Fig. 2 shows the results of moving object detection using the improved Gaussian mixture model. Snapshot Fig. 2(a) was taken with an out-door flame and a walking man. Snapshot Fig. 2(c) is a scene of a camp fire beside a road with street lights and moving vehicles. The moving objects, including the walking man, the flames and vehicle headlights, are segmented accurately, and the static background objects that have fire-like colors, such as street lamps, are eliminated as shown in Fig. 2(b) and Fig. 2(d) respectively.

2.2. Fire color feature extraction

Flames have distinctive visual features. Except for combustions involving very high temperature or certain types of fuel materials, where flame color can be blue, most diffusion fire flames display bright white color in the core region. The color changes from white to yellow then to red as the temperature decreases from the core region towards the flame edges [10]. Further flame image analysis was conducted in the current study to quantify the description of flame color features. Over 3000 images were collected from various sites under 50 different illumination backgrounds. Nearly 117 k flame pixels were identified manually from the images collected. The RGB values of these pixels formed a flame color database. Presented in Fig. 3 is a two-dimensional representation, or mapping, of the flame color RGB distribution. The abscissa in this figure is the G component value of a flame pixel and the ordinate are the corresponding R, G and B component values of the same pixel [27]. The mapping of G component naturally collapses on to a straight line, whilst the R and B components are scattered in two regions separated by the G component line. For most flame images, the R values are greater than the G values which in turn are greater than the B values. In addition, it can be seen from Fig. 3 that the majority of the red dots are above a threshold value, say 110. Based

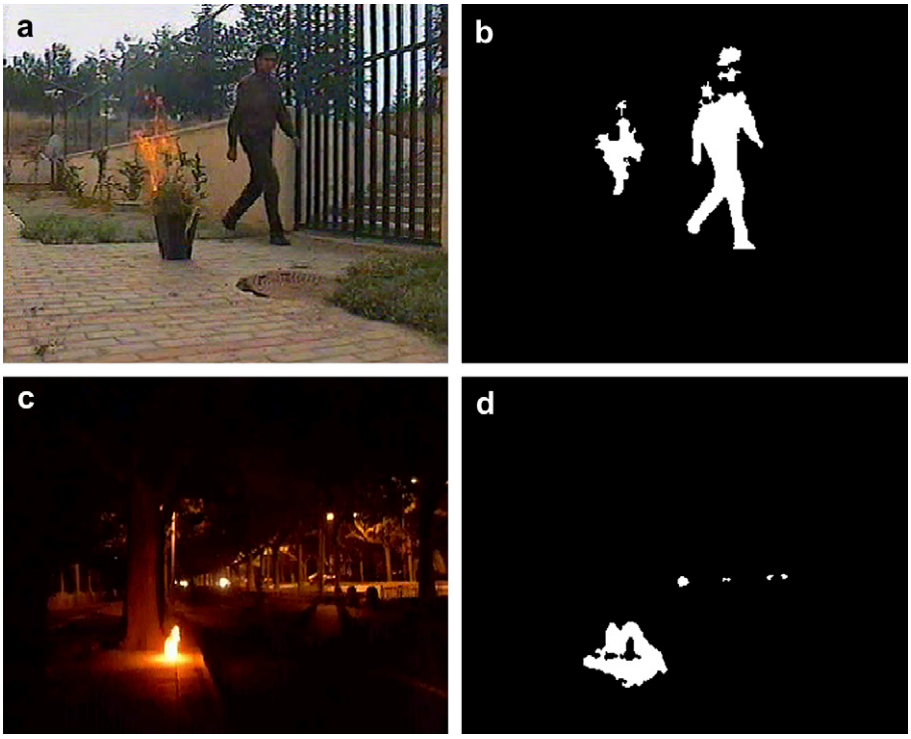

Fig. 2. Results of motion detection. (a) and (c) are original images, (b) and (d) are the binary maps BM of (a) and (c) respectively, which are extracted using background subtraction.

on these observations, the following two rules [13] are adopted for flame color identification in the current work:

$$\text{Rule1:} \quad R(x,y,t) \geq R_T \tag{14}$$

$$\text{Rule 2}: \quad R(x,y,t) \geq G(x,y,t) \geq B(x,y,t) \tag{15}$$

where $R(x, y, t)$, $G(x, y, t)$, $B(x, y, t)$, denotes the R, G, B components of a given pixel coordinates (x, y) at time t, and R_T is the global threshold value of R component It has been found through experiment that the optimum threshold values of R component is in a range from 115 to 135, which is in agreement with reference [13].

Furthermore, the background illumination may adversely affect the video images, resulting in spurious fire-liker regions which must be eliminated by the detection scheme. This elimination is achieved by introducing a color saturation constraint. Presented in Fig. 4 is a plot of the color saturation *vs* blue color scale of true flame images captured by video cameras. As can be seen from this figure, there is a correlation between the blue color component and the saturation. The saturation of the fire pixels are almost all in a bounded region enclosed by curves $Y1$ and $Y2$. As a result, the third rule is introduced to further consolidate the determination of fire colored pixels:

$$\text{Rule 3}: \quad Y2 \leq S(x,y,t) \leq Y1 \tag{16}$$

where $S(x, y, t)$ represents the value of saturation [28], $Y1$ and $Y2$ are the upper and lower boundaries obtained by curve fitting as shown in Fig. 4. The expressions for $Y1$ and $Y2$ are as follows,

$$Y1 = \begin{cases} 100 - 0.48B(x,y,t), & \text{if } B(x,y,t) \geq 117 \\ 79.27 - 0.311B(x,y,t), & \text{if } B(x,y,t) < 117 \end{cases} \tag{17}$$

$$Y2 = -2.0147 + 90.59435\, e^{-B(x,y,t)/77.6027} \tag{18}$$

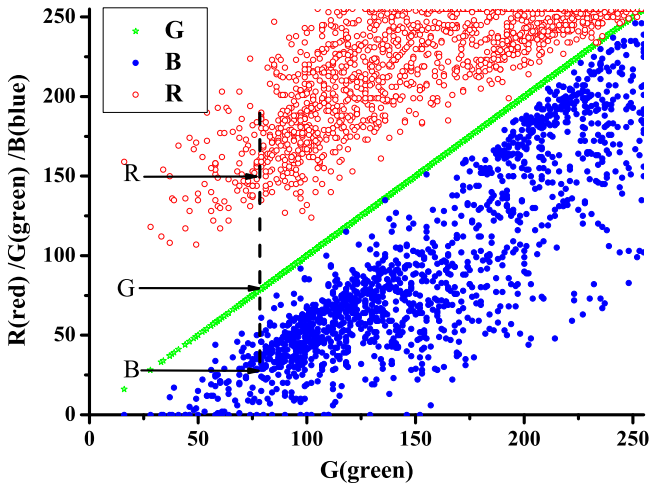

Fig. 3. Two-dimensional RGB color component mapping of flame pixels.

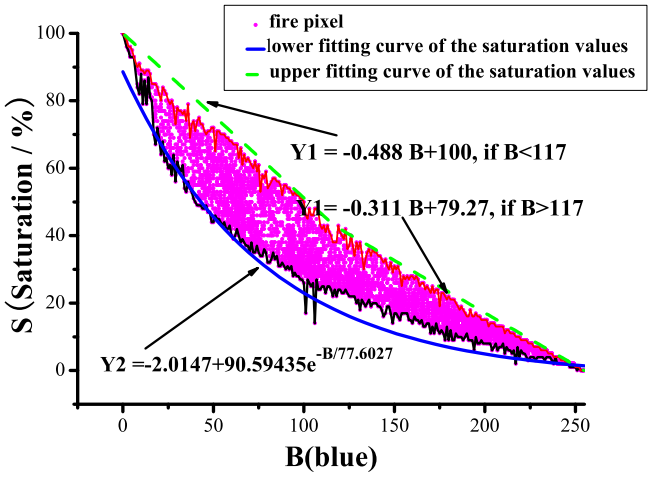

Fig. 4. The saturation against the blue component of the fire pixels.

To save computation time, a look-up table for the discrete B value domain [0, 255] is generated from Eqs. (17) and (18) and is used in the processing scheme to perform Rule 3 check.

In order to eliminate foreground moving objects which do not correspond to flames, the above described rules are implemented for pixels which are judged as pixels of moving objects based on background subtraction in the binary map BM. The pixels which have ordinary motion and fire-like color characteristics are considered as potential flame pixels. Through color discrimination, a binary Color Map (CM) is generated. CM(x, y, t) indicates whether a pixel at a given location and time (x, y, t) is classified as potential fire pixel (binary value of 1) or non-potential fire pixel (binary value of 0). The decision-making formula is as follows.

$$\mathrm{CM}(x,y,t) = \begin{cases} 1, & \text{if } (\mathrm{BM}(x,y,t) = 1)\&(\vec{X}_{xy,t} \subseteq \\ & (\text{Rule 1 \& Rule 2 \& Rule 3})) \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

Signal noises existing in the RGB values may sometimes cause incorrect registrations in CM. In order to remove these noises, morphological operations of erosion and dilation are applied [29]. To merge the same object together as far as possible, the adjacent pixels of *CM* are linked appropriately. The foreground pixels can then be labeled into connected regions by an eight-connectivity component labeling algorithm [29] while any connected component with less than 10 pixels are removed.

Fig. 5 displays the results of the combined motion detection and color mapping for the examples presented in Fig. 2. The potential fire candidates are encircled with rectangular frames.

In Fig. 5(b), the moving pedestrian image in Fig. 5(a) is filtered out from the corresponding binary image presented in Fig. 2(b). The effectiveness of color mapping is demonstrated. However, both the moving flame and the moving vehicle headlights in Fig. 5(c) are retained in Fig. 5(d) after the combined motion detection and color filtering process. Further processing is required to eliminate the images of fire-like objects.

2.3. Flame flicker feature extraction

It has been observed that burning fires exhibit flickering, or pulsation, movement with apparent random oscillations at flame edge, forming intermittent flame regions [15,16,30]. This flickering motion is caused by turbulent entrainment flows of the surrounding air. According to Hamins et al. [30], the dynamic frequency range of flame flicker is around 10 Hz no matter what the fuels are. More observations by Yang and Wang [31] indicate that the contours and chrominance or brightness of flames generally oscillate with a frequency range of 0.5–20 Hz.

The characteristic oscillation frequency of the flickering motion offers an opportunity to differentiate fire flames from other moving light sources by performing a frequency analysis. However, according to the Shannon sampling theorem, the sampling frequency of a signal should be greater than twice the signal frequency. Unfortunately, the video capture rate is generally at 25 Hz (25 frames per second) in most surveillance systems. This sampling frequency can only reveal any parameter change rate of less than 12.5 Hz, and cannot meet the sampling requirement. In addition, transforming signals from the time domain to the frequency domain is time-consuming and will affect the efficiency of the detection algorithm.

A different approach is employed in the current study to resolve this problem. The approach is described below.

With the passage of time, the brightness and color of a pixel at (x, y) that registers a flickering motion of a flame vary significantly so that the corresponding CM will undergo transformations, disappearing into background and reappearing in the foreground. This transformation or oscillation is repeated several times per second. Let SUM be a matrix that register the oscillation frequency of pixels.

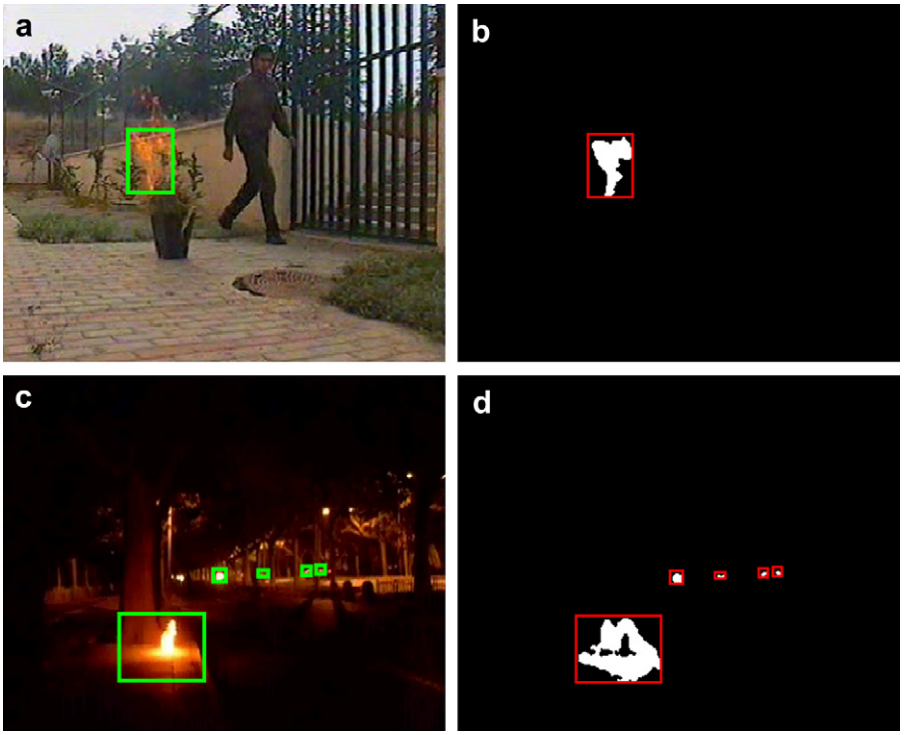

Fig. 5. Results of fire recognition with motion detection and fire color criteria. (a) and (c) are original images, (b) and (d) are the binary maps CM of (a) and (c) respectively, which are extracted by using motion and color filtering.

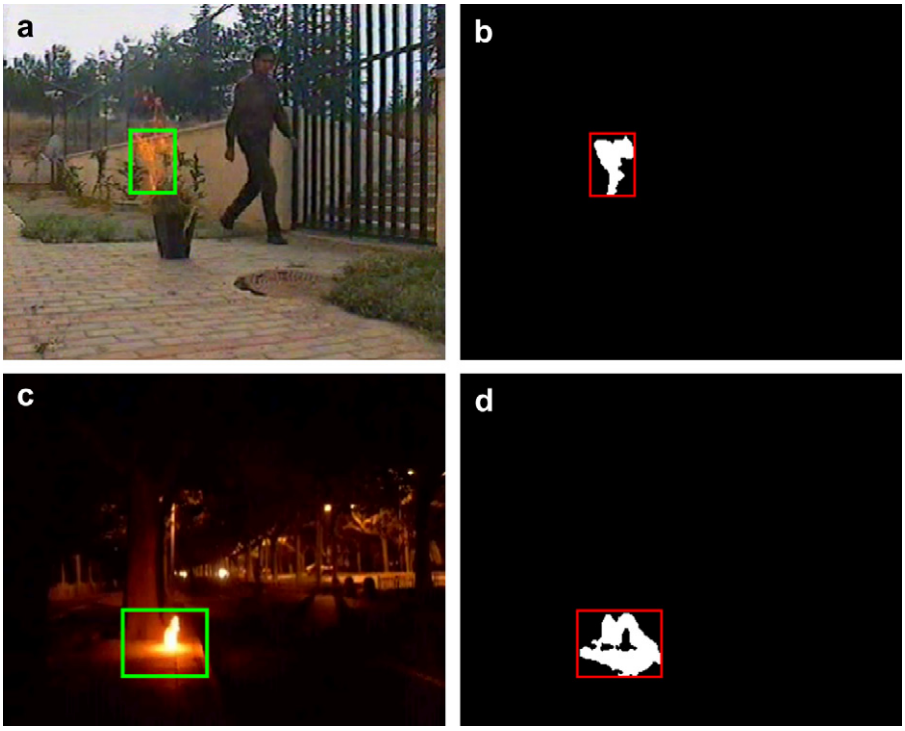

Fig. 6. Final flame recognitions based on flame flicker analysis. (a) and (c) are original images, (b) and (d) are the corresponding binary maps of (a) and (c) after motion detection, color filtering and flickering analysis.

If the brightness value of a pixel at (x, y) changes between two consecutive image frames, $\mathrm{SUM}(x, y, t)$ is added by 1, or 0 otherwise. To eliminate the impact of system noise, a global threshold T_I is introduced for registering the change of brightness, $\Delta I(x, y, t)$. The above scheme is formulated in Eqs. (20)–(22) below.

$$\mathrm{SUM}(x,y,t) = \begin{cases} \mathrm{SUM}(x,y,t-1)+1, & \text{if}(\Delta I \geq T_\mathrm{I}) \\ \mathrm{SUM}(x,y,t-1)+0, & \text{if}(\Delta I < T_\mathrm{I}) \end{cases} \tag{20}$$

where

$$\Delta I(x,y,t) = |I(x,y,t) - I(x,y,t)| \tag{21}$$

and

$$I(x,y,t) = \frac{1}{3}[\mathrm{R}(x,y,t) + \mathrm{G}(x,y,t) + \mathrm{B}(x,y,t)] \tag{22}$$

The brightness I can also be equal to $0.30\mathrm{R} + 0.59\mathrm{G} + 0.11\mathrm{B}$, according to the transformation from the RGB color space into the YUV color space. But for computation performance, equally weighted average of the three colors is used to reduce floating point computation.

The value of the threshold T_I in Eq. (20) was determined by averaging the changes in brightness of the image pixels in recorded video sequences. The threshold T_I was set to a value slightly greater than the average change by a fraction of 10% to minimize of the effect of noise.

A pixel is regarded as part of the flickering flame picture if its oscillation registration counter exceeds a threshold SUM_0 in a given period:

$$(\mathrm{SUM}(x,y,t) - \mathrm{SUM}(x,y,t-N)) > \mathrm{SUM}_0 \tag{23}$$

where N is the counting period. The threshold SUM_0 is also related to the counting period. Its value was determined in image laboratory experiments and by comparing the oscillation registration counters for flickering motion pixels with that for non-flickering pixels in 14 different video clips. It was found that the

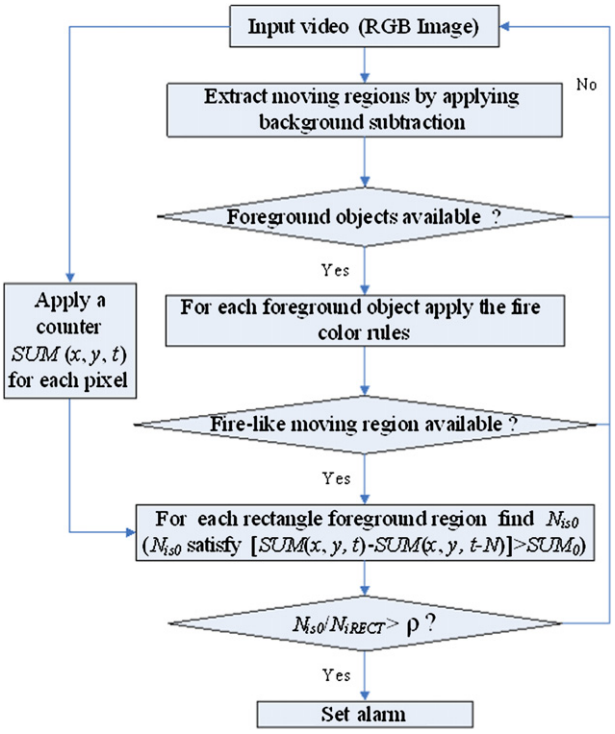

Fig. 7. Flow chart of the proposed fire detection algorithm.

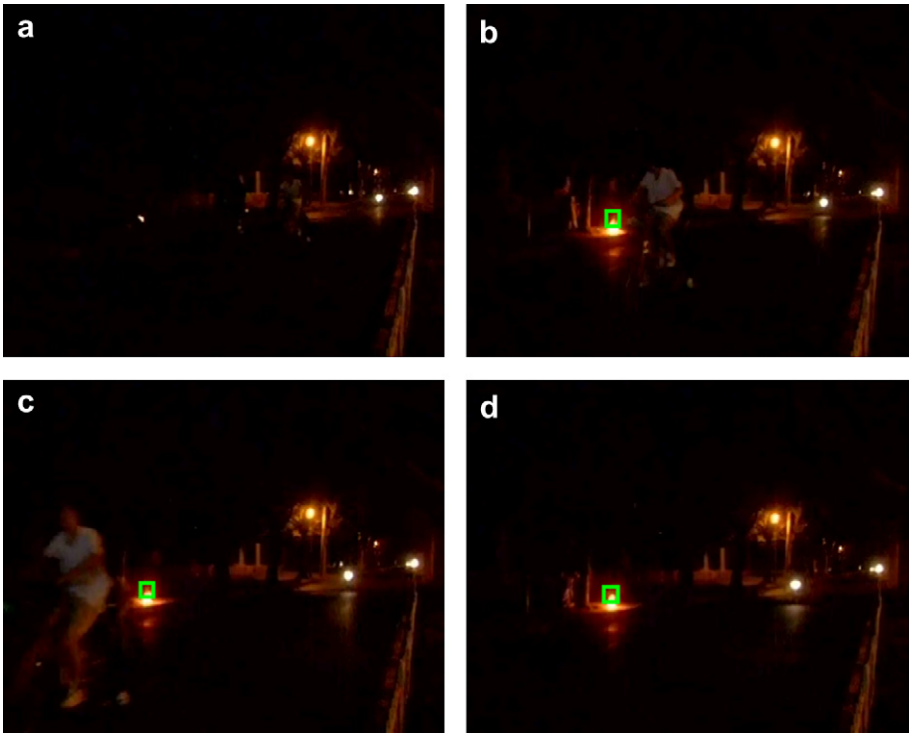

Fig. 8. Detection result of highway scene.

oscillation counters for flickering pixels registered much larger values than those for non-flickering pixels. The value of SUM_0 was then set in between the two group average values.

The flicker analysis is performed to the candidate flame regions in the binary map *CM* [see the rectangles in Fig. 5(b and d)] after the color filtering described in Section "Fire Color Feature Extraction". The number, N_{is0}, of the pixels that satisfy Eq. (23), as well as the total number, N_{iRECT}, of the object pixels within the candidature flame region, is counted. If the ratio of the two numbers for a candidate flame region exceeds a threshold ρ, which is determined by real video fire clips, i.e.,

$$N_{is0}/N_{iRECT} > \rho \tag{24}$$

then the region is regarded as a flame region; otherwise, it is regarded as a pseudo-flame region.

Fig. 6 depicts the results of the combined motion detection, color filtering and flickering analysis to two original video sequences. It can be seen by comparing Fig. 6(d) with Fig. 5(d) that disturbances of vehicle headlights were removed after flickering analysis.

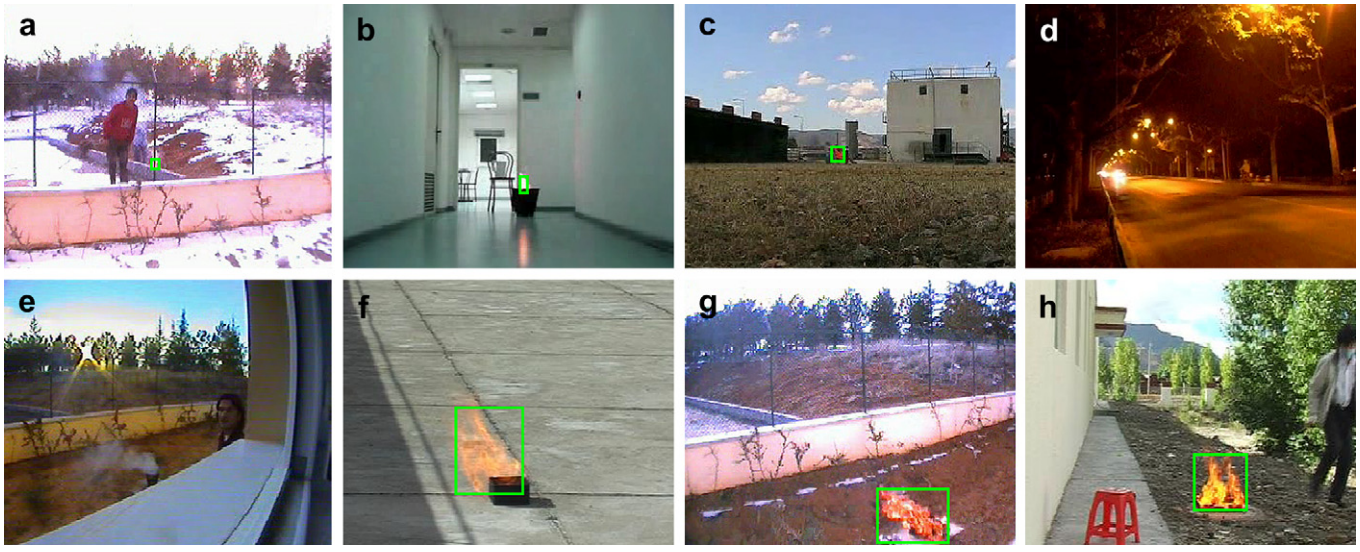

Fig. 9. Testing images of eight different scenes. Successful detections of flames are marked by green frames around the identified objects and successful omissions of non-fire moving light sources are shown in (d) and (e).

2.4. Flow chart of the algorithm

The flow chart of the proposed video flame detection algorithm is presented in Fig. 7.

3. Testing and analysis

The proposed algorithm has been implemented using C++ and tested on a PC with an AMD 2.04 GHz processor and 1 GB memory. The algorithm was applied to a set of video clips from a variety of scenes including different environmental background and illumination conditions. The free available video clips were collected from http://202.117.58.58/files/videoForFlasm.rar and http://signal.ee.bilkent.edu.tr/VisiFire/Demo/. The resolution of the video images are 320 × 240.

Fig. 8 contains multiple images of a fire beside a highway at nighttime. In the original video clip, the ignition occurs at frame No. 635, and the flame can be identified by visual inspection of frame No. 643. Fig. 8 presents frames Nos. 645, 680, 694 and 704. The detection of the flame by the proposed method is highlighted by the green rectangle inside the pictures. It can be seen from these results that the proposed algorithm was able to detect the fire in a very short time, less than 2 s from the start of the fire. The detection was not disturbed by the cyclist, street lamps and vehicle headlights.

Presented in Fig. 9 are test images of eight different scenes. The fire in Fig. 9(a) is against the high-brightness of snow and there are some pedestrians and brown grassland in the background. Fig. 9(b) depicts a fire in a corridor with a strong floor reflection. The fire in Fig. 9(c) is located on a lawn. In all of these scenes the fires appear at long ranges which result in small observable flame images. Nevertheless, the fires were successfully detected. Fig. 9(f–h) demonstrate the successful detection of close range fires against not much dissimilar background colors. The steady and moving light sources in Fig. 9(d and e) were successfully eliminated by the detection algorithm.

The test results have shown the adaptability and robustness of the proposed detection algorithm under different scenes and illumination conditions. The processing rate of the flame detection method achieved 24 fps in the sample applications.

4. Concluding remarks

In this study, a video flame detection method has been developed based on the multi-feature fusion. The method incorporates three features, namely, motion detection, statistical color filtering and statistical flickering analysis, into one scheme. The method is able to achieve high detection reliability and fast processing speed for reasonably high-resolution images. Firstly, a real-time adaptive background subtraction method based on a Gaussian mixture model is applied to aid the segmentation of moving objects which may be potential fire flames. Secondly, the images are further subjected to color mapping and flickering counting to distinguish true fire flames from fire-like objects. The improved Gaussian mixture model and the simple flickering identification method significantly enhanced the efficiency of the data processing. In particular flickering identification method is based on the statistics of the change frequencies in pixel brightness in video sequences. This method overcomes the difficulty and complexity associated with the traditional transformation and analysis in the frequency domain.

The experimental results, which were obtained under a variety of experimental environments, demonstrate that the multi-feature fusion video flame detection algorithm has good adaptability, robustness and efficiency. Moreover, this method can be sued for real-time surveillance and flame detection. It should be pointed out that the testing of the algorithm was conducted on video clips obtained from independent sources, which were taken under limited environmental conditions. Other environmental conditions, such as wind or ambient air movement and heat release rate of the fires, were not known in those incidences. Further laboratory real-time experiments with wider a range of recoded conditions are required to achieve more quantifiable performance assessment.

The video flame detection technology relies on the flame visual characteristics of fires. This limits its applications to flaming fire detection only. It can be used to enhance fire detection, but not to replace other type of detectors with which non-flaming and/or non-visible fires can be detected. Further work is needed to extend the video flame detection technology to detect flames with unusual color characteristics, such as those produced by hydrogen or metal fuels.

The detection algorithm presented in this paper can be combined with a location determination algorithm to generate information for automatic fire suppression and fire fighting operations. It is possible to extend the technology such that not only the location but also the size of the detected fires can be determined.

Acknowledgements

The authors are indebted to Mr Ma Jian and Dr Jeffrey Zou for their helpful discussions. The support from the Ministry of Science and Technology of the People's Republic of China (2006BAK06B07) is acknowledged.

References

[1] Proulx G. Evacuation time, the SFPE handbook of fire protection engineering. 4th ed. Quincy, MA: National Fire Protection Association; 2008. p. 3.355–3.372.
[2] Tavares RM, Galea ER. Evacuation modelling analysis within the operational research context: a combined approach for improving enclosure designs. Building and Environment 2009;44(5):1005–16.
[3] Custer RLP, Meacham BJ, Schifiliti RP. Design of detection systems, the SFPE handbook of fire protection engineering. 4th ed. Quincy, MA: National Fire Protection Association; 2008. p. 4.1–4.44.
[4] Cote AE, editor. Fire protection handbook. 19th ed. Quincy, Mass: National Fire Protection Association; 2003.
[5] Wong J, LI H. Development of a conceptual model for the selection of intelligent building systems. Building and Environment 2006;41(8):1106–23.
[6] Phillips III W, Shah M, da Vitoria Lobo N. Flame recognition in video. Pattern Recognition Letters 2000;23(1–3):319–27.
[7] Healey G, Slater D, Lin T, Drda B, Goedeke AD. A system for real-time fire detection. In: Proceedings of conference on computer vision and pattern recognition (CVPR '93); 1993. p. 605–6.
[8] Cheng XF, Wu JH, Yuan X, Zhou H. Principles for a video fire detection system. Fire Safety Journal 1999;33(1):57–69.
[9] Yamagishi H, Yamaguchi J. Fire flame detection algorithm using a color camera. In: Proceedings of 1999 International symposium on micromechatronics and human science; 1999. p. 255–60.
[10] Liu CB, Ahuja N. Vision based fire detection. In: Proceedings of the 17th International conference on pattern recognition (ICPR '04); 2004. p. 134–7.
[11] Wren C, Azarbayejani A, Darrell T, Pentland A. Pfinder: real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 1997;19(7):780–5.
[12] Celik T, Demirel H, Ozkaramanli H, Uyguroglu M. Fire detection using statistical color model in video sequences. Journal of Visual Communication and Image Representation 2007;18(2):176–85.
[13] Chen TH, Wu PH, Chiou YC. An early fire-detection method based on image processing. In: Proceedings of IEEE International Conference on image processing (ICIP '04); 2004. p. 1707–10.
[14] Ono T, Ishii H, Kawamura K, Miura H, Momma E, Fujisawa T, et al. Application of neural network to analyses of CCD colour TV-camera image for the detection of car fires in expressway tunnels. Fire Safety Journal 2006;41(4):279–84.
[15] McCaffrey BJ. Fire plume dynamics – a review. In: Proceedings of large-scale fire phenomenology conference; 1984. p. 1–10.
[16] Detriche P, Lanore JC. An acoustic study of pulsation characteristics of fires. Fire Technology 1980;16(3):204–11.
[17] Toreyin BU, Dedeoglu Y, Gudukbay U, Cetin AE. Computer vision based method for real-time fire and flame detection. Pattern Recognition Letters 2006;27(1):49–58.

[18] Dedeoglu Y, Toreyin BU, Gudukbay U, Cetin AE. Real-time fire and flame detection in video. In: Proc. IEEE International Conference on acoustics, speech, and signal processing (ICASSP '05); 2005. p. 669–72.
[19] Toreyin BU, Dedeoglu Y, Cetin AE. Flame detection in video using hidden markov models. In: Proc. 2005 International conference on image processing (ICIP 2005); 2005. p. 2457–60.
[20] Marbach G, Loepfe M, Brupbacher T. An image processing technique for fire detection in video images. Fire Safety Journal 2006;41(4):285–9.
[21] Schultze T, Kempka T, Willms I. Audio-video fire-detection of open fires. Fire Safety Journal 2006;41(4):311–4.
[22] Chen T, Yuan HY, Su GF, Fan WC. Automatic fire searching and suppression system for large spaces. Fire Safety Journal 2004;39(4):297–307.
[23] Stauffer C, Grimson WEL. Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000;22(8):747–57.
[24] Huang WQ, Wang YM. Tracking and detecting multi-objects based on adaptive mixture models and features of region. Computer Measurement & Control 2003;11(9):648–50. 654 [in Chinese].
[25] Gianluca B, Massimo B. Background estimation with Gaussian distribution for image segmentation, a fast approach. In: Proceedings of the 2005 IEEE International workshop on measurement systems for homeland security, contraband detection and personal safety workshop (IMS 2005); 2005. p. 29–30.
[26] Meng F, Wang CR. Research of fast moving object detection in multimode background. Electronic Measurement Technology 2007;30(6):34–5. 79 [in Chinese].
[27] Cheng X, Wang DC, Yin DL. Image type fire flame detecting principle. Fire Safety Science 2005;14(4):239–45 [in Chinese].
[28] Gonzalez RC, Woods RE. Digital image processing. Beijing: Publishing House of Electronics Industry; 2003 [in Chinese].
[29] Gonzalez RC, Woods RE, Eddins SL. Digital image processing using MATLAB. Beijing: Publishing House of Electronics Industry; 2005 [in Chinese].
[30] Hamins A, Yang JC, Kashiwagi T. An experimental investigation of the pulsation frequency of flames. In: Proceedings of the 24th (International) symposium on combustion; 1992. p. 1695–702.
[31] Yang J, Wang RS. A survey on fire detection and application based on video image analysis. Video Engineering 2006;8:92–6 [in Chinese].

Nomenclature

$B_{xy,t}$: background subset of the K Gaussian distributions for a pixel (x, y) at time t
$B(x, y, t)$: blue component of a given pixel with coordinates (x, y) at time t
$BM(x, y, t)$: Background Map, a binary map corresponding background subtraction
$CM(x, y, t)$: Color Map, a binary map corresponding color filtering
$G(x, y, t)$: green component of a given pixel with coordinates (x, y) at time t
I: identity matrix
$I(x, y, t)$: brightness of pixel (x, y) at time t
K: the number of Gaussian distributions in a mixture model
$M_{xy,k,t}$: matching function of the kth Gaussian distribution of a pixel with coordinates (x, y)at time t
N: a given integer in order to track for N consecutive frames of the changes in brightness
N_{is0}: the number of the pixels that satisfy $S(x, y, t) > S_0$ within the identified candidate rectangular region i
N_{iRECT}: the number of the object pixels within the identified candidate rectangular region i
$P(\vec{X}_{xy,t})$: the probability of observing a given color vector value for a pixel (x, y)
$R(x, y, t)$: the red component of a given pixel with coordinates (x, y) at time t
R_T: a threshold of R component
$SUM(x, y, t)$: counter that registers the oscillation frequency of a pixel with coordinates (x, y) at time t
SUM_0: threshold of the oscillation frequency of flames after tracking for N consecutive frames
$S(x, y, t)$: the saturation of a given pixel with coordinates (x, y) at time t
T: the minimum portion accounted for the background
T_I: a global threshold for registering the change of brightness
T_m: threshold that determines the match of inactive pixels between their observations and mean vector
$\vec{u}_{xy,k,t}$: mean vector of the kth Gaussian for a given pixel with coordinates (x, y) in the mixture at time t
$\vec{u}_{A,k,t}$: mean vector with the highest priority of active pixel A at time t
$w_{xy,k,t}$: weight of the kth Gaussian in a mixture model for a given pixel with coordinates (x, y) at time
$\vec{X}_{xy,t}$: color feature vector for a given pixel with coordinates (x, y) at time t
x: x coordinate
$\vec{u}_{m,t}$: the average mean of a inactive pixel m at time t
$\vec{X}_{m,t}$: the observation of a inactive pixel m at time t
$\vec{X}_{A,t}$: the observation of a active pixel A at time t
y: y coordinate
α: learning rate of background updating
$\Delta I(x,y,t)$: the change in brightness of a pixel at (x, y) between consecutive frame at time t
λ: the moderation factor that determines the matching distance between the observation values and corresponding mean for the updating scheme in the GMM. See Eq. (6)
λ_1: the moderation factor that determines the matching distance between the observation values and corresponding mean for simplified updating scheme. See Eq. (9)
η: Gaussian probability density distribution function
ρ: the threshold of area ratio for a candidate flame region
$\Sigma_{xy,k,t}$: covariance matrix of the kth Gaussian for a given pixel with coordinates (x, y) in the mixture at time t
$\sigma_{xy,k,t}$: standard deviation for a given pixel with coordinates (x, y) at time
$\sigma_{A,k,t}$: standard deviation of a active pixel A at time t