IET Image Processing

Research Article

Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis

ISSN 1751-9659
Received on 24th September 2014
Revised on 14th April 2015
Accepted on 23rd April 2015
doi: 10.1049/iet-ipr.2014.1032
www.ietdl.org

Feiniu Yuan[1] ✉, *Zhijun Fang*[2], *Shiqian Wu*[3], *Yong Yang*[1], *Yuming Fang*[1]

[1]*School of Information Technology, Jiangxi University of Finance & Economics, Nanchang, Jiangxi 330032, People's Republic of China*
[2]*College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Longteng Road No. 333, Songjiang District, Shanghai 201620, People's Republic of China*
[3]*College of Machinery and Automation, Wuhan University of Science and Technology, Wuhan, Hubei 430081, People's Republic of China*
✉ *E-mail: yfn@ustc.edu*

Abstract: It is very challenging to accurately detect smoke from images because of large variances of smoke colour, textures, shapes and occlusions. To improve performance, the authors combine dual threshold AdaBoost with staircase searching technique to propose and implement an image smoke detection method. First, extended Haar-like features and statistical features are efficiently extracted from integral images from both intensity and saturation components of RGB images. Then, a dual threshold AdaBoost algorithm with a staircase searching technique is proposed to classify the features of smoke for smoke detection. The staircase searching technique aims at keeping consistency of training and classifying as far as possible. Finally, dynamic analysis is proposed to further validate the existence of smoke. Experimental results demonstrate that the proposed system has a good robustness in terms of early smoke detection and low false alarm rate, and it can detect smoke from videos with size of 320×240 in real time.

1 Introduction

Traditional paradigms for fire detection generally use very precise sensors, which are based on particle sampling, temperature sampling, relative humidity sampling and smoke analysis [1]. Those sensors are widely applied because of the low cost and simplicity. Since those sensors require physical contact with combustion products or environmental gas, the sensors must be installed in close proximity of fires, otherwise fires cannot be detected at all. Accordingly, traditional sensor-based fire detection can only be applied for indoor environments. Fire detection in video eliminates distance limitation and provides abundant visual information about fire. It's very promising that fire detection in video is used in applications. Fire detection in video can be classified into two categories that are video flame detection and video smoke detection according to detected objects.

Yamagishi and Yamaguchi [2] presented a flame detection algorithm to extract flame contours using colour information in spatio-temporal domain. A fire detection system based on grey-scale images applied in tunnels was implemented in [3]. A flame detection technique based on video using a Gaussian-smoothed colour histogram was proposed in [4]. Toreyin *et al.* [5] utilised multiple features, such as motion, flicker, edge blurring and colour for video flame detection. Temporal and spatial wavelet transforms are performed to extract characteristics of flicker and edge blurring. A video flame detection using mixture Gaussian model to extract temporal features was proposed by Yuan *et al.* [6]. Generally speaking, video flame detection methods provide faster response and wider monitored scenes than traditional sensor-based fire detection. However, robustness of these methods is a great issue because of large variances of fire texture, colour and shape, and disturbances of fired-colour objects.

Smoke detection provides earlier fire alarm than flame detection because smoke usually appears before flame does. So far, techniques on video smoke detection can be classified into three categories [7, 8]. The first category is a histogram-based smoke detection. In these techniques, histogram of smoke is constructed to detect presence of smoke. Then, statistical measures, such as mean and standard deviation, are calculated to determine smoke probability. The second one is temporal techniques. Differences of successive frames and wavelet transform of temporal pixel intensities are used to extract time-varying features [9]. The third one is rule-based techniques [10], that is, knowledge of fire is encoded as rules to infer the presence of smoke.

Toreyin *et al.* [9] proposed a method to discriminate smoke and non-smoke objects by fusing features of motion, flicker, edge blurring and colour. A smoke video detection algorithm using wavelets and support vector machine (SVM) was presented in [11]. They calculated arithmetic mean, geometric mean, standard deviation, skewness, kurtosis and entropy in each sub-band of three-level wavelet transformed images. A smoke detection method for forest fire detection was proposed in [12]. The method includes four successive steps: temporal embedding of grey-levels, fractal indexing of points, chaining points into a linked list, and motion extraction from point sequences of the linked list. An SVM-based method to detect steam in video was proposed in [13]. In their method, a statistical hidden Markov tree (HMT) model, which is derived from coefficients of the dual-tree complex wavelet transform (DT-CWT) in small local regions of image sequences, is used to characterise steam texture pattern. In [14], the effectiveness of commercial video fire detection systems for small, cluttered spaces on navy ships was evaluated. Experiments show that video fire detection systems can detect fires more accurately and efficiently than traditional fire detection systems. Yuan [15] proposed an accumulative motion model for video smoke detection, in which integral image was used to fast estimate the motion orientation of smoke. A video smoke detection using both colour and motion features was proposed in [16]. Yuan [17] utilised variants of LBPs to propose a video based smoke detection with histogram sequences of LBP and LBPV pyramids. Yuan [18] extracted shape-invariant features on multi-scale partitions with AdaBoost for video smoke detection. Recently, Saponara *et al.* [19] proposed to integrate standard smoke sensors with the results of their smoke detection method, which was used on passenger trains.

However, existing methods for video smoke detection still suffer from high false alarm rates or low detection rates because of large variances of smoke. It is still challenging to accurately detect smoke in videos because of the following reasons: (i) smoke colour varies largely and many non-smoke objects share the same colour with smoke; (ii) smoke textures and shapes often change irregularly; (iii) smoke often blurs images, resulting in unreliable features extracted from these blurred images; and (iv) foreground objects occlude smoke.

In this paper, we propose a robust real-time video smoke detection system. This paper has three contributions to video smoke detection. First, we combine extended Haar-like features and statistical measures together to propose a dual threshold AdaBoost algorithm, and these features are extracted from both the intensity and saturation components of original RGB images. The proposed method can detect smoke in videos at real-time speed because all the features are very simple and are able to be efficiently computed by integral images. Second, a staircase searching technique is proposed to keep consistency of training and detecting as far as possible, thus it improves the performance of the proposed method. Last, we use dynamic analysis to further validate the existence of smoke.

This paper is organised as follows. Section 2 describes fast feature extraction by integral images. In Section 3, we describe a dual threshold AdaBoost algorithm with a staircase searching technique in details. Section 4 proposes dynamic analysis of possible smoke regions detected by our method. Experimental results are demonstrated in Section 5, followed by conclusion drawn in Section 6.

2 Extraction of smoke features

2.1 Haar-like features

Theoretical analysis has proved that any number of weak classifiers with an error rate less than 50% can be combined to form an ensemble classifier whose final error rate approaches that of an optimal classifier. In [20], an AdaBoost algorithm was proposed to train one ensemble classifier by iteratively changing weight distribution of positive and negative samples. This method has been demonstrated to be an effective machine learning method to simultaneously select and combine relevant features from a large feature set, for instance, face detection [21], fingerprint classification [22] and so on. It is shown in [21] that the powerful feature selection by the AdaBoost method makes it possible to use a very simple feature extractor. In [21], Haar-like features are extracted and then cascade AdaBoost method is employed for face detection. This is the first paradigm to achieve real-time face detection with high performance. Inspired by this idea, we decide to use the Haar-like features to recognise smoke from images in real time.

A two-rectangle Haar-like feature is defined as difference of pixel sums of two rectangular regions, which can be at any position and scale within an original image. Haar-like features describe certain characteristics in the underlying image, such as edges or changes in texture. For instance, a 2-rectangle feature indicates the border between a light region and a dark region. In [21], Viola and Jones defined 3-rectangle and 4-rectangle features. However, our experiments demonstrated that the Haar-like features defined in [21] are not sufficient to represent detailed characteristics of smoke because smoke regions have large variations in colour, shapes and textures. In this work, some of extended Haar-like features proposed by Lienhart *et al.* [23, 24] are employed for smoke feature extraction.

As shown in Fig. 1, the features used in our method are classified into four categories: edge features, line features, diagonal line features and centre-surround features. Each feature is assigned a unique type code. For example, **code 11** represents a 2-rectangle feature that indicates the horizontal edge between a light region and a dark region. Each rectangular Haar-like feature can be totally determined by five parameters (c, x, y, w, h), that is, type code c, x and y coordinates of the start point, width w and height h of the rectangle.

2.2 Feature computation by integral images

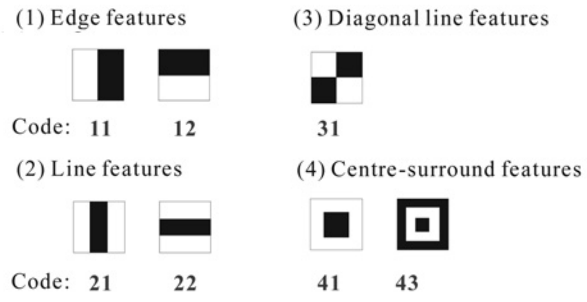


Fig. 1 *Extended Haar-like features used for image smoke detection*

Viola and Jones [21] originally used a summed-area table in the form of a matrix with the same size of the original image, which is called an integral image [23]. Actually, an integral image $ii(x, y)$ is an intermediate representation for an image $i(x, y)$ and contains sum of intensities located in up-left region of the original image, that is

$$ii(x, y) = \sum_{x'=0}^{x}\sum_{y'=0}^{y} i(x', y') \qquad (1)$$

This allows us to compute sum of rectangular areas in the image $i(x, y)$, at any position or scale, using only four lookups

$$S(x_1, y_1, x_2, y_2) = \sum_{y=y_1}^{y_2}\sum_{x=x_1}^{x_2} i(x, y) = ii(x_1, y_1) + ii(x_2, y_2) - ii(x_1, y_2) - ii(x_2, y_1) \qquad (2)$$

According to (2), the sum of a rectangle with bottom left point (x_1, y_1) and top right point (x_2, y_2) is computed using only four lookups. Each Haar-like feature may need more than four lookups, depending on how it was defined. For example, a 2-rectangle feature requires only six lookups, 3-rectangle features need eight lookups and 4-rectangle features need nine lookups.

It should be highlighted that several experiments show that intensity images computed from RGB videos are insufficient for smoke detection because of the extreme diversity of non-uniform visual appearance of smoke. Generally, smoke behaves in totally unpredictable ways because of its fluid characteristics. In some cases, smoke flows rapidly and appears dense, so that most regions are with high frequency components. In other cases, it may flow slowly and appears sparsely, which results in most regions with low frequency components. The experiments in subsequent sections validate that extended set of Haar-like features over intensity images have not sufficient discriminating capability. In most dangerous fires, smoke emerges often in white or black colours, which are high intensity and low intensity values both with low saturation. Therefore low saturation of colour is an important feature for smoke detection. To improve detection performance, saturation of colour is extracted from RGB images of videos to generate a saturation image. Then, the extended Haar-like features are computed from both intensity and saturation images. To unify feature codes, the code of a feature computed from a saturation image is 1000 larger than the one from the corresponding intensity image. For example, **code 11** represents a 2-rectangle feature computed from an intensity image while **code 1011** is a 2-rectangle feature from saturation images.

In addition, it is found that statistical measures of intensity and saturation images are also very important features for smoke detection. Means of intensity and saturation images within an arbitrary rectangle, which are coded as **01** and **1001** respectively, are efficiently computed by

$$\mathrm{E}(i) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} S(x_1, y_1, x_2, y_2) \qquad (3)$$

where $S(x_1, y_1, x_2, y_2)$ is the sum of a rectangle described as (2).

According to statistics theory, variance can be computed by the following equation

$$\sigma^2 = \mathrm{E}(i^2) - \mathrm{E}(i) \cdot \mathrm{E}(i) \tag{4}$$

where $\mathrm{E}(i)$ is the mean of image $i(x, y)$ in a rectangle with bottom left point (x_1, y_1) and top right point (x_2, y_2). In order to efficiently compute variance, another integral image $ii_2(x, y)$ is constructed by computing the sum of squared values of pixel intensity with height x and width y, defined as the following equation

$$ii_2(x, y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} i^2(x', y') \tag{5}$$

Then, $\mathrm{E}(i^2)$ can be quickly computed by the following equation

$$\begin{aligned}\mathrm{E}(i^2) = {} & \frac{1}{(x_2 - x_1)(y_2 - y_1)}[ii_2(x_1, y_1) \\ & + ii_2(x_2, y_2) - ii_2(x_1, y_2) - ii_2(x_2, y_1)]\end{aligned} \tag{6}$$

Accordingly, variances of intensity and saturation images within an arbitrary rectangle are quickly obtained via (4), which are coded as **02** and **1002**, respectively. It is seen that the variance of a rectangle is computed using only eight lookups, that is, four lookups in the integral image $ii(x, y)$ and four lookups in the squared integral image $ii_2(x, y)$. Now, 18 types of features are extracted using extended Haar features and statistical measures for smoke detection. Codes of the 18 feature types are set to **01**, **02**, **11**, **12**, **21**, **22**, **31**, **41**, **43**, **1001**, **1002**, **1011**, **1012**, **1021**, **1022**, **1031**, **1041** and **1043**, respectively.

3 Dual threshold AdaBoost with staircase searching

3.1 Single threshold AdaBoost

An AdaBoost classifier is normally used to solve the following three issues in one boosting procedure: (i) selecting discriminative features from a large redundant feature set, (ii) constructing a weak classifier for each selected feature and (iii) boosting all weak classifiers into a stronger classifier. For each feature, a weak classifier finds an optimal threshold classification function, such that the minimum number of samples is mis-classified. A weak classifier $h(\boldsymbol{x}, f, p, \theta)$ consists of a sample $\boldsymbol{x}$, a feature f, a threshold θ and a polarity p that indicates the direction of an inequality, defined as the following equation

$$h(\boldsymbol{x}, f, p, \theta) = \begin{cases} 1 & \text{if } pf < p\theta \\ 0 & \text{else} \end{cases} \tag{7}$$

For a binary problem of smoke and non-smoke, a set of N labelled training samples is given as $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$, where $y_i = 1$ for smoke and $y_i = 0$ for non-smoke. In fact, each sample $\boldsymbol{x}_i$ is normalised to a 24 × 24 image in our implementation. A feature f is defined as one of 14 types of extended Haar features as shown in Fig. 1, and 4 types of means and variances of a rectangle. Weights $\{w_1, \ldots, w_N\}$ assigned to each of training samples are used to determine training errors. The details of the standard AdaBoost algorithm can be found in [21]. To distinguish between the dual threshold AdaBoost algorithm described in Section 3.2 and the standard AdaBoost algorithm, the standard AdaBoost algorithm is called a single threshold AdaBoost algorithm because a single threshold is used.

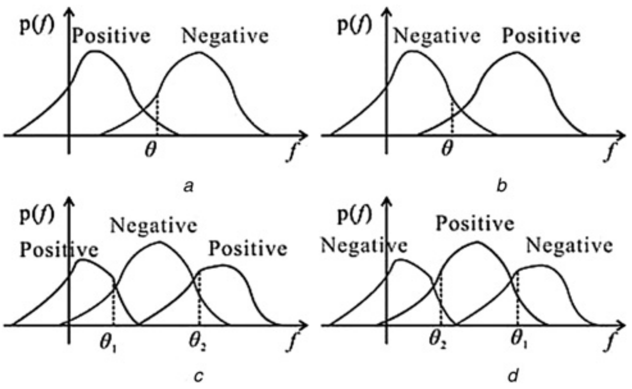


Fig. 2 *Single-peak and two-peak feature distributions*

a $p = 1$
b $p = -1$
c $\theta_1 < \theta_2$
d $\theta_1 \geq \theta_2$

3.2 Dual threshold AdaBoost

The single threshold AdaBoost is suitable for positive and negative samples with single-peak distributions, as shown in Figs. 2*a* and *b*. However, there is usually black smoke or white smoke, resulting in multi-peak distributions as shown in Figs. 2*c* and *d*, and accordingly, a dual threshold AdaBoost algorithm, similar to [25], is adopted to improve training performance.

Following (7), the polarity p is replaced with another threshold θ_2, and its original threshold θ is retained and denoted by θ_1. So the dual threshold AdaBoost algorithm has two thresholds, θ_1 and θ_2, as shown in Figs. 2*c* and *d*. The number of parameters for the dual threshold AdaBoost is the same as that of the single threshold AdaBoost algorithm. First, consider the situation, where $\theta_1 < \theta_2$, as shown in Fig. 3*a*. The method in [21] is adopted to search the two thresholds. For each feature, samples are sorted in ascending order based on the feature values. Then, four sums are maintained and evaluated for each element in the sorted list, that is, total sum of all positive sample weights T^+, total sum of all negative sample weights T^-, sum of positive weights with feature values less than ones of current sample S^+ and sum of negative weights with feature values less than ones of current sample S^-. As shown in Fig. 3*a*, the misclassified rate of positive samples is the sum of positive weights between θ_1 and θ_2, and the misclassified rate of

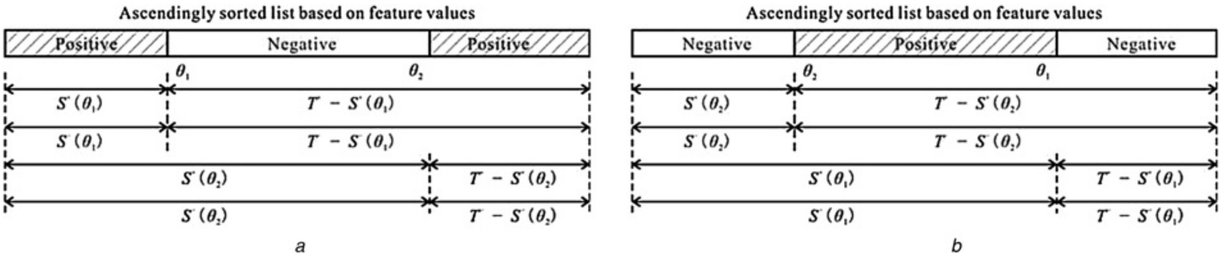
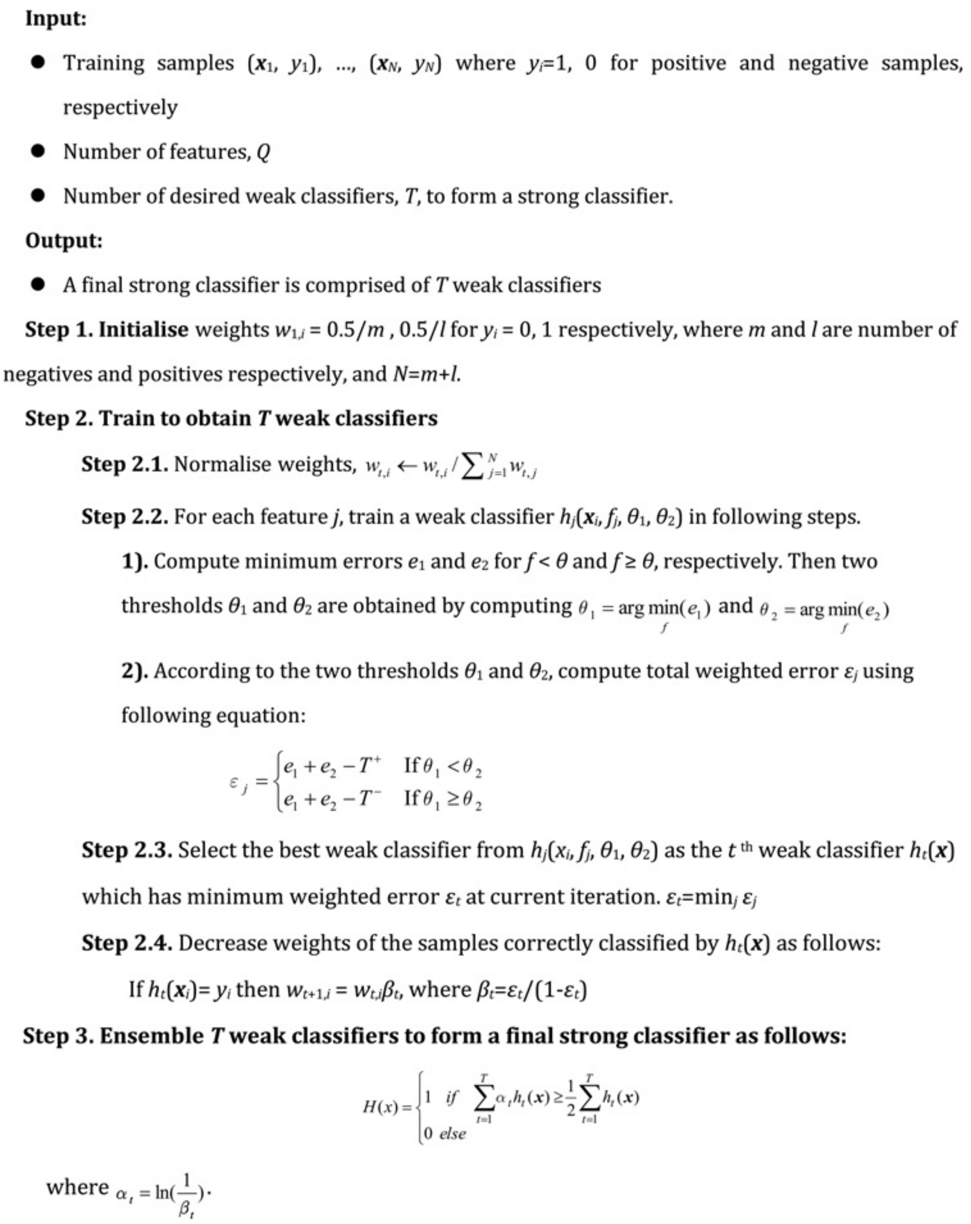

Ascendingly sorted list based on feature values
Positive
Negative
Positive
θ1
θ2
$S^{+}(\theta_1)$
$T^{+} - S^{+}(\theta_1)$
$S^{-}(\theta_1)$
$T^{-} - S^{-}(\theta_1)$
$S^{+}(\theta_2)$
$T^{+} - S^{+}(\theta_2)$
$S^{-}(\theta_2)$
$T^{-} - S^{-}(\theta_2)$
a
Ascendingly sorted list based on feature values
Negative
Positive
Negative
θ2
θ1
$S^{+}(\theta_2)$
$T^{+} - S^{+}(\theta_2)$
$S^{-}(\theta_2)$
$T^{-} - S^{-}(\theta_2)$
$S^{+}(\theta_1)$
$T^{+} - S^{+}(\theta_1)$
$S^{-}(\theta_1)$
$T^{-} - S^{-}(\theta_1)$
b

Fig. 3 *Searching of two optimal thresholds θ_1 and θ_2*

a $\theta_1 < \theta_2$
b $\theta_1 \geq \theta_2$



```
Input:
  ● Training samples ($\boldsymbol{x}_1$, $y_1$), …, ($\boldsymbol{x}_N$, $y_N$) where $y_i$=1, 0 for positive and negative samples, respectively
  ● Number of features, $Q$
  ● Number of desired weak classifiers, $T$, to form a strong classifier.
Output:
  ● A final strong classifier is comprised of $T$ weak classifiers
Step 1. Initialise weights $w_{1,i} = 0.5/m$ , $0.5/l$ for $y_i = 0, 1$ respectively, where $m$ and $l$ are number of negatives and positives respectively, and $N=m+l$.
Step 2. Train to obtain T weak classifiers
    Step 2.1. Normalise weights, $w_{t,i} \leftarrow w_{t,i} / \sum_{j=1}^{N} w_{t,j}$
    Step 2.2. For each feature $j$, train a weak classifier $h_j(\boldsymbol{x}_i, f_j, \theta_1, \theta_2)$ in following steps.
        1). Compute minimum errors $e_1$ and $e_2$ for $f < \theta$ and $f \geq \theta$, respectively. Then two thresholds $\theta_1$ and $\theta_2$ are obtained by computing $\theta_1 = \arg\min_f(e_1)$ and $\theta_2 = \arg\min_f(e_2)$
        2). According to the two thresholds $\theta_1$ and $\theta_2$, compute total weighted error $\varepsilon_j$ using following equation:
            $\varepsilon_j = \begin{cases} e_1 + e_2 - T^+ & \text{If } \theta_1 < \theta_2 \\ e_1 + e_2 - T^- & \text{If } \theta_1 \geq \theta_2 \end{cases}$
    Step 2.3. Select the best weak classifier from $h_j(x_i, f_j, \theta_1, \theta_2)$ as the $t$ th weak classifier $h_t(\boldsymbol{x})$ which has minimum weighted error $\varepsilon_t$ at current iteration. $\varepsilon_t = \min_j \varepsilon_j$
    Step 2.4. Decrease weights of the samples correctly classified by $h_t(\boldsymbol{x})$ as follows:
        If $h_t(\boldsymbol{x}_i) = y_i$ then $w_{t+1,i} = w_{t,i}\beta_t$, where $\beta_t = \varepsilon_t/(1-\varepsilon_t)$
Step 3. Ensemble T weak classifiers to form a final strong classifier as follows:
    $H(x) = \begin{cases} 1 & if \quad \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \geq \frac{1}{2} \sum_{t=1}^{T} h_t(\boldsymbol{x}) \\ 0 & else \end{cases}$
  where $\alpha_t = \ln(\frac{1}{\beta_t})$.
```

Fig. 4 *Pseudo code of the dual threshold AdaBoost algorithm*

negative samples is equal to the sum of negative weights below θ_1 and above θ_2. Therefore, the misclassified rates of positive and negative samples, denoted by e_a^+ and e_a^-, are formulated as follows

$$e_a^+ = S^+(\theta_2) - S^+(\theta_1) \quad (8)$$

$$e_a^- = S^-(\theta_1) + T^- - S^-(\theta_2) \quad (9)$$

The total error e_a is equal to the sum of the positive and negative misclassified rates

$$e_a = e_a^+ + e_a^- = e_1 + e_2 - T^+ \quad (10)$$

where

$$e_1 = S^-(\theta_1) + T^+ - S^+(\theta_1) \quad (11)$$

$$e_2 = S^+(\theta_2) + T^- - S^-(\theta_2) \quad (12)$$

The total error e_a gets minimised when the errors e_1 and e_2 are minimal as T^+ is constant. The two minimum errors e_1 and e_2 for that feature can be computed in a single pass over the sorted list. θ_1 and θ_2 are the feature values when the errors e_1 and e_2 are minimal as follows

$$\theta_1 = \arg\min_f(e_1) \quad (13)$$

$$\theta_2 = \arg\min_f(e_2) \quad (14)$$

Next, we consider another case, where $\theta_1 < \theta_2$, as shown in Fig. 3*b*. The total error e_b is computed as

$$e_b = e_b^+ + e_b^- = e_1 + e_2 - T^- \quad (15)$$

where

$$e_b^+ = S^+(\theta_2) + T^+ - S^+(\theta_1) \quad (16)$$

$$e_b^- = S^-(\theta_1) - S^-(\theta_2) \quad (17)$$

Similarly, the total error e_b can be minimised by obtaining minimal e_1 and e_2.

In summary, we define a new weak classifier $h(\boldsymbol{x}, f, \theta_1, \theta_2)$ consisting of a sample $\boldsymbol{x}$, a feature f, two thresholds θ_1 and θ_2, as the following equation

$$h(\boldsymbol{x}, f, \theta_1, \theta_2) = \begin{cases} \begin{cases} 1 & \text{if } f < \theta_1 \text{ or } f \geq \theta_2 \\ 0 & \text{otherwise} \end{cases} & \text{if } \theta_1 < \theta_2 \\ \begin{cases} 1 & \text{if } \theta_2 \leq f < \theta_1 \\ 0 & \text{otherwise} \end{cases} & \text{if} \theta_1 \geq \theta_2 \end{cases} \quad (18)$$

In fact, the inequations of $\theta_1 < \theta_2$ and $\theta_1 \geq \theta_2$ in the dual threshold AdaBoost are equivalent to the polarity p in the single threshold AdaBoost, that is, $p = \text{sign}(\theta_2 - \theta_1)$, where sign($x$) is a function that returns 1 and −1 for $x > 0$ and $x \leq 0$, respectively. In other words, the polarity p is hidden and replaced by the relationship of the two thresholds θ_1 and θ_2. The dual threshold AdaBoost has two parameters, θ_1 and θ_2, and the single threshold AdaBoost also has two parameters, θ and p. Pseudo code of the dual threshold AdaBoost algorithm is summarised in Fig. 4.

3.3 Staircase searching

A huge number of smoke features are extracted from many training samples, so a very large feature pool is produced and huge memory is required. To reduce memory consumption, each feature is usually stored in an 8 or 16-bit integer. To efficiently search two thresholds for a weak feature f, all the samples are sorted in ascending order of the feature f, as shown in Fig. 5. Then, the thresholds θ_1 and θ_2 are found by searching the minimal classification errors defined in (11) and (12) for all features.

It is noted that there may be many samples corresponding to the same feature values as shown in Fig. 5. The procedure of minimisation is often implemented by changing thresholds θ_1 and θ_2, so that it does not consider the equality of the feature values. Therefore the position of a threshold found by the traditional searching method may be located in the middle of the sorted samples with the same feature value. As shown in Fig. 5, θ_1 is located at the n_3th sample, and the traditional searching method regards the n_1th and n_2th samples as one category and the n_3th, n_4th and n_5th samples as another category. It is highlighted that the weak classifier with the threshold θ_1 actually considers all samples with the same features as the same category, that is, the n_1th, n_2th, n_3th, n_4th and n_5th samples are classified as the same object, which results in inconsistent training and classifying.

Accordingly, a new searching method based on staircases is proposed to keep such consistency. First, we sort the samples in an ascending order of features and create a sorted sample list for each feature. Then, we find all change points of features in the sorted sample list. The change points denoted as $s_0, s_1, \ldots, s_m$ are called staircases, as shown in Fig. 5. Finally, we search the thresholds on the space of staircases. Similarly, four sums based on staircases are maintained and evaluated for each element in the sorted sample list, that is, total sum of positive sample weights T^+, total sum of negative sample weights T^-, sum of positive weights below current staircase C^+ and sum of negative weights below current staircase C^-. To find the thresholds on the space of staircases, we replace S^+ and S^- in (8), (9), (11), (12), (16) and (17) with C^+ and C^-, respectively. Thus, we can make the training and classifying stages to be consistent and quickly find the thresholds.

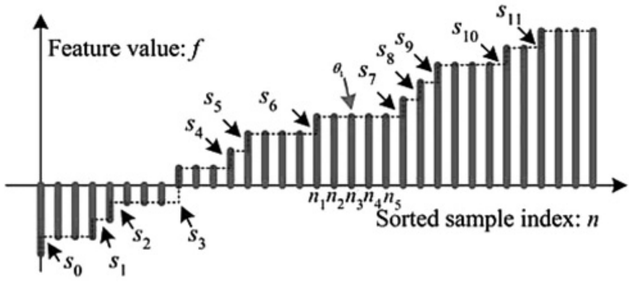


Fig. 5 *Staircase searching*

4 Dynamic analysis of candidate smoke

It is very challenging to accurately detect smoke from images since smoke region has largely varying shapes, textures, colours and so on. Large variances of within-class will lead to high false alarm rates that are not allowed in real applications. However, it is also noted that a short-time fire alarm, which lasts for less than a few seconds, can be viewed as safety without any anxiety. On the basis of this idea, the candidate smoke regions detected by AdaBoost are further analysed in successive frames of video.

The detection rate R_D and false positive rate R_F can be properly pre-specified from the receiver operating characteristic (ROC) curve of the proposed dual threshold AdaBoost detector. If the probability of smoke captured in a specified region is P_S, the probability of non-smoke is then $1-P_S$ for the same region. To estimate the probability of smoke detected by AdaBoost, a temporal window with size N_W consisting of a fixed number of frames is selected. As shown in Fig. 6, a red rectangle denotes a possible smoke region detected by the AdaBoost detector. The estimated probability of detected smoke in a specified region is written as follows

$$P_{\text{Smoke}} = \frac{1}{N_{\text{w}}} \sum_{i=0}^{N_{\text{w}}-1} D_{\text{s}}(n-i) \cong R_{\text{D}} \cdot P_{\text{S}} \quad (19)$$

where

$$D_{\text{s}}(n) = \begin{cases} 1 & \text{if smoke is detected} \\ 0 & \text{else} \end{cases} \quad (20)$$

The estimated probability of false smoke detected by the AdaBoost detector is given by

$$P_{\text{Non}} = \frac{1}{N_{\text{w}}} \sum_{i=0}^{N_{\text{w}}-1} D_{\text{s}}(n-i)[1 - A_{\text{S}}(n-i)] \cong R_{\text{F}} \cdot (1 - P_{\text{S}}) \quad (21)$$

where

$$A_{\text{s}}(n) = \begin{cases} 1 & \text{if there is actual smoke} \\ 0 & \text{else} \end{cases} \quad (22)$$

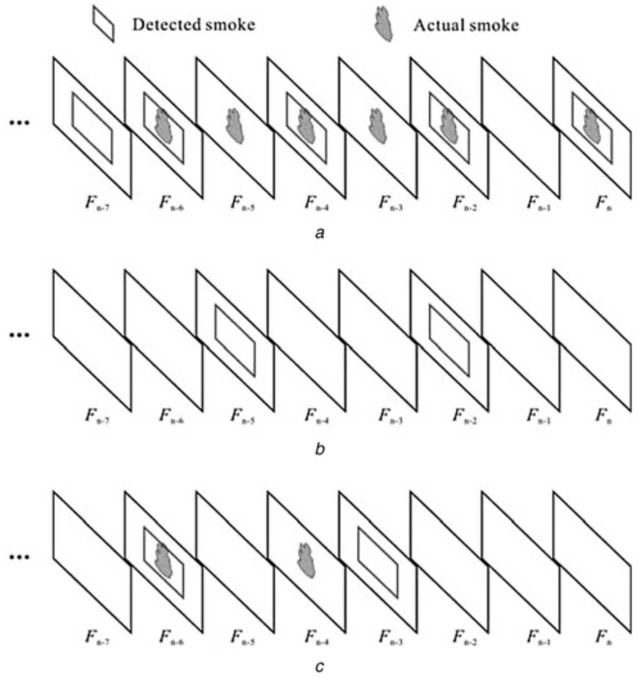


Fig. 6 *Dynamic estimation of smoke probability detected by the AdaBoost detector*

a Sequence of dense smoke
b Sequence of non-smoke
c Sequence containing sparse smoke

It is noted that $A_S(n)$ is usually unknown. Equation (22) is given only for theoretic analysis. A predetermined threshold T is properly selected to meet the following equation

$$P_{\text{Non}} < T < P_{\text{Smoke}} \tag{23}$$

The equation is approximately written as follows

$$R_F \cdot (1 - P_S) < T < R_D \cdot P_S \tag{24}$$

The final decision function can be written as

$$H_D = \begin{cases} \text{smoke alarm} & \text{if } P_{\text{Smoke}} > T \\ \text{non-smoke} & \text{else} \end{cases} \tag{25}$$

If there is dense smoke in the specified region, the probability P_S of smoke is relatively high (usually above 0.8), and the probability of non-smoke ($1-P_S$) is low (usually below 0.2). If R_D and R_F are set to 0.8 and 0.1, respectively, then $R_F{*}(1-P_S) = 0.02$ and $R_D{*}P_S = 0.64$. In our experiments, we select $T = R_D{*}0.5 = 0.4$ s, which is usually smaller than P_{Smoke} computed by (19). This implies that the number of smoke alarms decreases via index T. If there is no smoke, the probability P_S is low, usually below 0.2. Then P_{Smoke} is less than T and no alarm is produced.

In Fig. 6*a*, there is dense smoke in the specified region and the window size N_W is 8. The AdaBoost detector misclassifies two smoke regions as non-smoke and one non-smoke region as smoke. P_{Smoke} is $5/8 = 0.625$, which is greater than $T = 0.4$, and an alarm is raised. In Fig. 6*b*, there is no smoke in the specified region and two non-smoke regions are misclassified. P_{Smoke} is $2/8 = 0.25$, which is less than T, and no alarm is produced. If there is sparse smoke in the specified region as shown in Fig. 6*c*, no alarm is made in this case because the P_{Smoke} (2/8) is less than T. Even if P_{Smoke} is above T, our system also does not make alarm as long as it lasts less than a few of seconds.

5 Experimental results

5.1 Data and implementation

We manually established two image databases containing smoke and non-smoke images for training and testing the AdaBoost algorithm. Some image samples were captured by ourselves, and others were collected from the internet using searching engines. Each image is cropped, resized and labelled as a positive or negative sample. The training set *lib*1, containing 2201 smoke images and 8511 non-smoke images, is used to select features by the AdaBoost algorithms. The testing set *lib*2, including 2254 smoke images and 8363 non-smoke images, is used for testing. The two image databases totally consist of 4455 positive images and 16 874 negative samples. In our implementation, the minimum searching window has the size of 24 × 24, so the positive and negative images are also normalised to 24 × 24 before feature extraction. The algorithm is implemented using Visual C++, and tested on PC platforms.

5.2 Performance analysis

To evaluate the performance of the dual threshold AdaBoost algorithm, the training set *lib*1 is used to train the single and dual threshold AdaBoost algorithms. Experiments show that the training error with dual thresholds is obviously lower than that with single threshold. In other words, the dual threshold AdaBoost algorithm has better convergence performance than the single threshold AdaBoost algorithm so that we can use less features to achieve the same classification accuracy by using the dual threshold AdaBoost algorithm.

To validate the significance of extracted features, we train the dual threshold AdaBoost algorithm using one feature pool $\mathbf{F_1}$ containing extended Haar features and another feature pool $\mathbf{F_2}$ consisting of extended Haar features as well as means and variances extracted from the training set *lib*1, respectively. Performances of different features on the testing set *lib*2 are evaluated by ROC curves as shown in Fig. 7*a*. It is seen that the integration of extended Haar features with statistical features can improve detection performances.

Using the feature pool $\mathbf{F_2}$, we train the single threshold AdaBoost algorithm, and the dual threshold AdaBoost algorithm with or without the staircase searching technique on the training set *lib*1. Then the ROC curves of the three methods on the testing set *lib*2 are depicted in Fig. 7*b*. As we can see, the ROC curve of the dual threshold AdaBoost method is approximately located in the lowest position, and the ROC curve of the dual threshold AdaBoost method with the staircase searching technique is nearest to upper left corner of the chart among the three ROC curves. This implies that the dual threshold AdaBoost method with the staircase searching technique has the best generalisation among them because the staircase searching technique can keep training and classifying to be consistent.

It should be highlighted that an appropriate trade-off must be made from the ROC curve depending on different applications. As shown in Fig. 7*b*, when the false alarm rate is around 0.02, the detection rates of the dual threshold AdaBoost algorithms with and without the staircase searching techniques are about 0.95 and 0.8, respectively. Low detection rate leads to high missing rate of fire alarms, which are not allowed in applications, while increasing detection rate yields high false alarm rate. So we propose a dynamic analysis of motion to reduce false alarm rate.

5.3 Smoke detection in video

As the system is proposed for real-time video surveillance, we also test our system in real videos. Fig. 8 visually illustrates

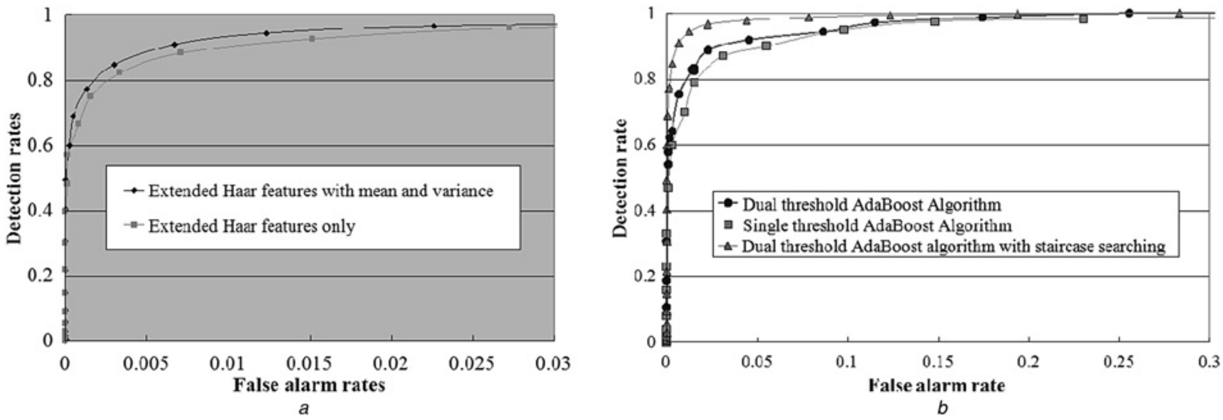


Fig. 7 *ROC curves*

a Extended Haar features with/without mean and variance
b Single and dual threshold AdaBoost algorithms

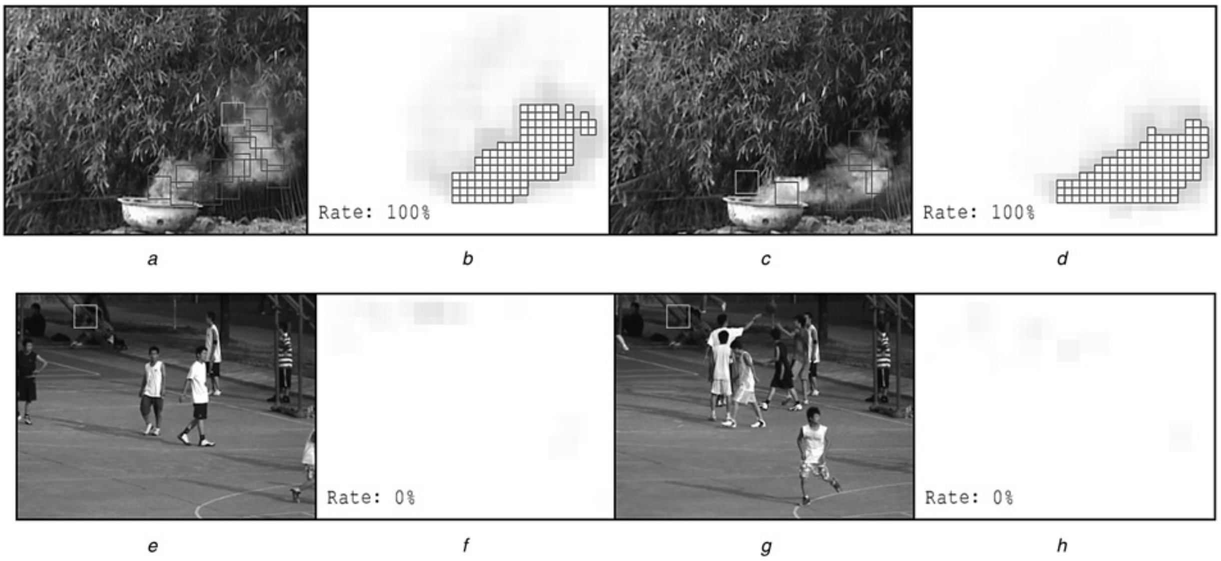


Fig. 8 *Results for smoke and non-smoke videos*

a Frame from a smoke video with detected smoke candidate regions
b Accumulated regions at the frame a
c Another frame from the smoke video with detected smoke candidate regions
d Accumulated regions at the frame c
e Frame from a non-smoke video with falsely detected regions
f Accumulated regions at the frame e
g Another frame from the non-smoke video with falsely detected regions
h Accumulated regions at the frame g

experimental results for a smoke video and a non-smoke video. First, the candidate regions of smoke are detected by the dual threshold AdaBoost algorithm. Then, dynamic analysis is used to eliminate noisy alarms. As shown in Fig. 8, green rectangles indicate that the probability of smoke regions is less than the predetermined threshold T and yellow ones stand for the smoke probability greater than the predetermined threshold T. If the total number of candidate smoke regions in yellow is more than a pre-defined threshold for several seconds, a fire alarm is raised. There may be false regions detected in both smoke and non-smoke videos. For example, the green rectangles in Figs. 8*c*, *e* and *g* denote the candidate rectangles falsely detected as smoke regions by the AdaBoost algorithm, but classified as non-smoke regions by the dynamic analysis method. Of course, the dynamic analysis method may sometime mis-classify actual smoke regions as non-smoke regions, as shown in Fig. 8*a*. Experimental results show that the probability of mis-classification of the dynamic analysis method is very low, and it is a useful technique to eliminate noisy alarm, thus it can reduce false alarm rates.

5.4 Comparisons

We compare the proposed algorithm with Toreyin's algorithm [5] and LBP/LBPV method [17]. Toreyin's algorithm discriminates smoke and non-smoke objects by fusing features of motion, flicker, edge blurring and colour. LBP/LBPV method utilises variants of LBPs to propose a video-based smoke detection with histogram sequences of LBP and LBPV pyramids. We used four fire smoke videos and four non-smoke videos, two of which are downloaded from http://www.signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html and six of which were captured by ourselves, respectively. Our training and testing image databases, and videos can be downloaded from http://www.staff.ustc.edu.cn/~yfn/vsd.html. The top row of Fig. 9 illustrates snapshots of four smoke video and the bottom row shows four non-smoke videos.

Table 1 Smoke detection on smoke videos

Videos	Duration	Alarm at frame number			Description
		Our method	Toreyin's method	LBP/ LBPV method	
movie 1	517	92	164	89	black smoke
movie 2	2886	87	216	94	white smoke of cotton rope fire
movie 3	1200	402	894	432	white smoke of dry leaf
movie 4	900	77	68	99	white smoke

Table 2 False alarms on non-smoke videos

Videos	Duration	Number of false alarms			Description
		Our method	Toreyin's method	LBP/LBPV method	
movie 5	895	0	0	0	waving leaves
movie 6	155	0	0	1	car lights in the night
movie 7	4536	0	4	1	basketball yard
movie 8	1000	0	0	0	traffic

As shown in Table 1, our method provides earlier fire alarms than Toreyin's method for Movies 1–3, but our method is a little later to give a fire alarm than Toreyin's method for Movies 4. The proposed method raised earlier fire alarms than LBP/LBPV method for Movies 2–4. However, it later gives alarms than LBP/LBPV method for Movies 1. It is seen from Table 2 that our method has generally fewer false alarms than Toreyin's method and LBP/LBPV method for Movies 5–8.

We conducted experiments for videos on a PC that is equipped with an AMD Phenom(tm) II X4 955 3.2 GHz Processor, and 8G memory. The sizes of all the videos are 320 × 240. The processing time for a frame is less than 40 ms. In other words, our method can process videos with size of 320 × 240 at above 25 frames per seconds. In the future, we need to further optimise our method to improve its processing speed for low power equipment's, such as mobile devices.

6 Conclusions

When smoke emerges, the quality of videos will greatly decrease, resulting in inaccurate, unreliable features. Up to now, most of video smoke detection methods suffer from high false alarm rates and low detection rates. To enhance the robustness of image smoke detection, we use extended Haar-like features and statistical features extracted from both intensity and saturation images. As the features are computed via integral images, the speed is very fast. Then, a dual threshold AdaBoost algorithm with a staircase searching technique is proposed for video smoke detection. This method avoids inconsistency of training and classifying, and greatly improves generalisation in comparison with the standard AdaBoost algorithm. To reduce false alarm rates, the dynamic analysis is proposed to further validate the existence of smoke by computing the smoke possibility of candidate regions from sequential images. Experiments show that our algorithm has a good robustness for smoke detection at interactive frame rates.

7 Acknowledgment

This work was supported by the Natural Science Foundation of China (61363038, 61371190), Cultivated Talent Program for Young Scientists of Jiangxi Province (20142BCB23014) and Science Technology Application Project of Jiangxi Province (KJLD12066).

8 References

1 Chen, T.H., Kao, C.L., Chang, S.M.: 'An intelligent real-time fire-detection method based on video'. Proc. IEEE 37th Annual Int. Carnahan Conf. Security Technology, Taiwan, October 2003, pp. 104–111

2 Yamagishi, H., Yamaguchi, J.: 'A contour fluctuation data processing method for fire flame detection using a color camera'. Proc. IEEE 26th Annual Conf. on IECON of Industrial Electronics Society, Nagoya, Japan, 2000, vol. 2, no. 22–28, pp. 824–829

3 Noda, S., Ueda, K.: 'Fire detection in tunnels using an image processing method'. Proc. Vehicle Navigation & Information Systems Conf., Yokohama, Japan, 1994, pp. 57–62

4 Phillips, W., Shah, M., Da Vitoria Lobo, N.: 'Flame recognition in video'. Proc. Fifth III IEEE Workshop Applications of Computer Vision, Palm Springs, CA, USA, 2000, pp. 224–229

5 Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: 'Computer vision based method for real-time fire and flame detection', *Pattern Recognit. Lett.*, 2006, **27**, (1), pp. 49–58

6 Yuan, F.N., Liao, G.X., Fan, W.C., *et al.*: 'Vision based fire detection using mixture Gaussian model'. Proc. Eighth Int. Symp. Fire Safety Science, Beijing, China, September 2005, pp. 1575–1583

7 Vicente, J., Guillemant, P.: 'An image processing technique for automatically detecting forest fire', *Int. J. Thermal Sci.*, 2002, **41**, (12), pp. 1113–1120

8 Dieter, W., Thomas, B.: 'Smoke detection in tunnels using video images'. Proc. 12th Int. Conf. Automatic Fire Detection, Maryland, USA, March 2001, pp. 79–90

9 Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: 'Wavelet based real-time smoke detection in video'. Proc. 13th European Signal Processing Conf., Antalya, Turkey, 2005

10 Simon, Y.F.: 'A rule-based machine vision system for fire detection in aircraft dry bays and engine compartments', *Knowl.-Based Syst.*, 1996, **9**, (8), pp. 531–540

11 Gubbi, J., Marusic, S., Palaniswami, M.: 'Smoke detection in video using wavelets and support vector machines', *Fire Safety J.*, 2009, **44**, (8), pp. 1110–1115

12 Guillemant, P., Vicente, J.: 'Real-time identification of smoke images by clustering motions on a fractal curve with a temporal embedding method', *Opt. Eng.*, 2001, **40**, (4), pp. 554–563

13 Ferrari, R.J., Zhang, H., Kube, C.R.: 'Real-time detection of steam in video images', *Pattern Recognit.*, 2007, **40**, (3), pp. 1148–1159

14 Gottuk, D.T., Lynch, J.A., Rose-Pehrssonb, S.L., *et al.*: 'Video image fire detection for shipboard use', *Fire Safety J.*, 2006, **41**, (4), pp. 321–326

15 Yuan, F.N.: 'A fast accumulative motion orientation model based on integral image for video smoke detection', *Pattern Recognit. Lett.*, 2008, **29**, (7), pp. 925–932

16 Yu, C.Y., Fang, J., Wang, J.J., *et al.*: 'Video fire smoke detection using motion and color features', *Fire Technol.*, 2010, **46**, (3), pp. 651–663

17 Yuan, F.N.: 'Video-based smoke detection with histogram sequence of LBP and LBPV pyramids', *Fire Safety J.*, 2011, **46**, (3), pp. 132–139

18 Yuan, F.N.: 'A double mapping framework for extraction of shape-invariant features based on multi-scale partitions with Adaboost for video smoke detection', *Pattern Recognit.*, 2012, **45**, (12), pp. 4326–4336

19 Saponara, S., Luca, P., Luca, F.: 'Early video smoke detection system to improve fire protection in rolling stocks'. Proc. of SPIE – The Int. Society for Optical Engineering, 2014, vol. 9139

20 Freund, Y., Schapire, R.E.: 'A decision-theoretic generalization of online learning and an application to boosting', *J. Comput. Syst. Sci.*, 1997, **55**, (1), pp. 119–139

21 Viola, P., Jones, M.J.: 'Robust real-time face detection', *Int. J. Comput. Vis.*, 2004, **57**, (2), pp. 137–154

22 Liu, M.H.: 'Fingerprint classification based on AdaBoost learning from singularity features', *Pattern Recognit.*, 2010, **43**, (3), pp. 1062–1070

23 Lienhart, R., Kuranov, A., Pisarevsky, V.: 'Empirical analysis of detection cascades of boosted classifiers for rapid object detection'. Proc. Pattern Recognition Symp., Otto-von-Guericke University, Magdeburg, Germany, September 2003, pp. 297–304

24 Lienhart, R., Maydt, J.: 'An extended set of Haar-like features for rapid object detection'. Proc. IEEE Int. Conf. Image Processing, Rochester, New York, USA, 2002, pp. 900–903

25 Wang, H.C., Zhang, L.M.: 'A novel fast training algorithm for AdaBoost', *J. Fudan Univ., Nat. Sci.*, 2004, **43**, (1), pp. 27–33