Fire Detection in Video Sequences Using Statistical Color Model

Conference Paper *in* Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on · June 2006

DOI: 10.1109/ICASSP.2006.1660317 · Source: IEEE Xplore

4 authors:

Turgay Celik
University of the Witwatersrand
117 PUBLICATIONS **3,882** CITATIONS

Hasan Demirel
Eastern Mediterranean University
195 PUBLICATIONS **5,624** CITATIONS

Huseyin Ozkaramanli
Eastern Mediterranean University
64 PUBLICATIONS **1,043** CITATIONS

Mustafa Kemal Uyguroğlu
Eastern Mediterranean University
30 PUBLICATIONS **520** CITATIONS

Available online at www.sciencedirect.com
ScienceDirect
J. Vis. Commun. Image R. 18 (2007) 176–185
JOURNAL OF VISUAL Communication & IMAGE Representation
www.elsevier.com/locate/jvci

Fire detection using statistical color model in video sequences

Turgay Celik *, Hasan Demirel, Huseyin Ozkaramanli, Mustafa Uyguroglu

Advanced Technologies Research and Development Institute, Eastern Mediterranean University, Gazimagusa TRNC, Mersin 10, Turkey

Received 5 September 2005; accepted 21 December 2006
Available online 10 January 2007

Abstract

In this paper, we propose a real-time fire-detector that combines foreground object information with color pixel statistics of fire. Simple adaptive background model of the scene is generated by using three Gaussian distributions, where each distribution corresponds to the pixel statistics in the respective color channel. The foreground information is extracted by using adaptive background subtraction algorithm, and then verified by the statistical fire color model to determine whether the detected foreground object is a fire candidate or not. A generic fire color model is constructed by statistical analysis of the sample images containing fire pixels. The first contribution of the paper is the application of real-time adaptive background subtraction method that aids the segmentation of the fire candidate pixels from the background. The second contribution is the use of a generic statistical model for refined fire-pixel classification. The two processes are combined to form the fire detection system and applied for the detection of fire in the consecutive frames of video sequences. The frame-processing rate of the detector is about 40 fps with image size of 176×144 pixels, and the algorithm's correct detection rate is 98.89%.

Keywords: Fire detection; Background subtraction; Change detection; Moving object detection; Statistical color model

1. Introduction

Visual fire detection can be useful in conditions where conventional fire detectors cannot be used. Particle and temperature sampling, and air transparency testing are simple methods that are frequently used for fire detection [1,2]. These methods require close proximity to the fire. In addition, these methods are not always reliable, as they do not always detect the combustion itself. Most of them detect smoke, which could be produced in other ways such as the cigarette smoke or smoke carried by wind from other places.

Existing methods of visual fire detection rely almost exclusively upon spectral analysis using rare and usually costly spectroscopy equipment. This limits the fire detection to those individuals who can pay high prices for expensive sensors that are necessary for these methods. Moreover, these methods still produce false alarms in the case of objects whose colors are almost the same as fire, especially sun.

In [3] and [4] two of the previously introduced vision-based methods are presented. However, both of the methods rely on the ideal conditions. In [1], color and motion are used to classify regions as fire or non-fire. Camera initialization requires the manual creation of rectangles based on the distance of portions of a scene from the camera. The second method [2] detects the fire using statistical methods that are applied to grayscale video taken with high-speed cameras. This method is computationally expensive, and only works for the ideal conditions in which there is very little chance that may be mistaken for fire.

In [5] specialized point based thermal sensors are used which changes intensity based on the temperature. A grayscale camera is used to observe these intensity changes at the various locations. Using the heat-transfer flow model gained from these sensors, a computer solves the location, size and intensity of the problem using the appropriately named inverse problem solutions. This method requires

* Corresponding author. Fax: +90 392 3650240.
E-mail address: turgay.celik@emu.edu.tr (T. Celik).

1047-3203/$ - see front matter
doi:10.1016/j.jvcir.2006.12.003

expensive sensors and additionally, the exact position of the sensors must be calibrated for the algorithm to be effective.

In [6] color predicate information and the temporal variation of a small subset of images are used to recognize flame in video sequences. It uses a manually labeled training set in the preprocessing phase where the training set is used to create a look-up table. A generic look-up table is used if the training set is not available.

Ref. [11] gives general information about the progress in the application of image processing algorithms in fire detection. However, the paper does not give enough details about the algorithms mentioned. In [12] and [13] the fire is detected as follows. First the contour of the flame area is extracted using statistical HSV color space. The extracted contour data is calculated by a polar coordinate transformation. The results of the polar coordinate transformation of every input image are placed in time series. Then, a fluctuation data is extracted, as a space-time data on the contour. A pattern of the frequency component distribution is obtained by Fourier transform of the fluctuation data. Then the extracted pattern is then entered into a neural network. The procedure gives good results but the computational complexity of the algorithm described is too high to be placed for real-time sequences. In [14], authors combine geostationary (high frequency overpass) and polar satellites thermal infrared (TIR) data to detect fire, which focuses on physical retrieved quantities, avoiding the use of 'ad hoc', regional or seasonal, thresholds. The radiances of the polar sensor TERRA/MODIS and an atmospheric radiative transfer model (RTM) are combined to estimate the ground emissivity once per week. A multi-temporal, sub-pixel version of the same RTM is applied to the measured radiances of the GOES/IMAGER geo-stationary satellite with 15 minutes revisit time. The output is the sub-pixel extension of possible fires (hot spots in general) and their average temperatures. The emissivity map estimated from the polar sensor is used to reduce the number of unknowns in the RTM equations and to suppress the false alarm rate. Since [14] requires usage of satellite and thermal infrared data to detect fire, it can't be classified as vision based fire detector. In [15], a combination of the surround temperature, smoke density, and carbon monoxide (CO) density is entered into fuzzy inference system to detect fire; the system completely depends on particle measuring, which is not the case in our application. In Ref. [16] a surveillance system for forest environments and protected natural areas is described. The system uses a camera to capture photo of the observed scene, and enables the remote operator to zoom, and rotate the camera, it also provides information about temperature, humidity, wind speed and direction, but could not detect the fire automatically.

In this paper, we propose an algorithm, which combines color information of fire with temporal changes, and background subtraction assisted foreground object segmentation to detect fire. A simple and effective color image based background subtraction method is used to detect changes in the scene observed which is explained in Section 2. The generic fire color model that is embedded into background subtraction output is explained in Section 3. Fire like colored objects may enter into the scene observed; in this situation the system should not consider the detected object as fire, (i.e. the sun). In order to compensate such a problem, motion information is embedded into final decision, this detection mechanism is explained in Section 4. The proposed algorithm's detection rate with computational complexity analysis is explained in Section 5. Future work and conclusions are presented in Section 6.

2. Background modeling

In order to detect possible changes, which may be caused from fire, we need to use an effective background-modeling algorithm. The algorithm should be simple and robust to achieve a real-time detection of the fire.

The background modeling used in our system is similar to the work done in [7] where the scene observed is almost stationary and the camera's position is fixed. The background is modeled with unimodal Gaussian, with mean and covariance matrix extracted from incoming image where incoming image is composed of Luminance Chroma-Blue and ChromaRed (YUV) components. In our system, incoming image is composed of Red, Green, and Blue (RGB) components.

The distributions of color channels of the each pixel are assumed to be independent, and modeled using a unimodal Gaussian whose parameters are settled in the training phase of the system. So for each pixel, an overall distribution model is estimated as follows;

$$p(I(x,y)) = p_{\mathrm{R}}(I_{\mathrm{R}}(x,y))p_{\mathrm{G}}(I_{\mathrm{G}}(x,y))p_{\mathrm{B}}(I_{\mathrm{B}}(x,y)) \tag{1}$$

where p_{R}, p_{G}, and p_{G} are distribution models for Red, Green and Blue channels respectively, and $I(x,y)$ pixel value at spatial location (x,y), and $p(I(x,y))$ is an approximation for probability density of $I(x,y)$. Each distribution is assumed to be independent of other distributions, and estimated as follows;

$$p_i(I_i(x,y)) = \frac{1}{\sqrt{2\pi}\sigma_i(x,y)} \exp\left(\frac{(I_i(x,y)-\mu_i(x,y))^2}{2\sigma_i^2(x,y)}\right), \quad i \in \{\mathrm{R},\mathrm{G},\mathrm{B}\} \tag{2}$$

where $I_i(x,y)$ is the value of $I(x,y)$ in ith color channel, $\mu_i(x,y)$ is the mean value of $I_i(x,y)$, $\sigma_i(x,y)$ is the standard deviation of $I_i(x,y)$, and $i \in \{\mathrm{R},\mathrm{G},\mathrm{B}\}$.

Initial values for $\mu_i(x,y)$ and $\sigma_i(x,y)$ are estimated using a training period of 20 s. Hence the overall scene is observed for 20 s, since the frame rate is 40 fps, a statistics over 800 frames are used to extract initial estimates for $\mu_i(x,y)$ and $\sigma_i(x,y)$.

2.1. Estimation of model parameters and change map

Even though the system is adaptive, and wrong initial estimates of the model parameters will be corrected in time, the wrong estimates may produce false positives in a short time, which is not desirable for a robust system. Because of this requirement, it is the essential to estimate initial values for $\mu_i(x,y)$ and $\sigma_i(x,y)$.

Let us have N frames for the training period. The estimation of $\mu_i(x,y)$ is straight forward, however $\sigma_i(x,y)$ requires two passes; first estimation of $\mu_i(x,y)$, and second variance over N frames. Because of this, we need a storage area for values of $I_i(x,y)$, which brings extra memory requirements to the system and so extra cost. This drawback could be overcome using the idea in [8] where $\sigma_i(x,y)$ is found using maximum absolute difference between consecutive frames. It is required to store only $I_i^{t+1}(x,y)$ and $I_i^t(x,y)$, and the difference between them, which are the pixel values of $I_i(x,y)$ at times $t+1$ and t, respectively. So $\mu_i(x,y)$, and $\sigma_i(x,y)$ can be estimated as follows [8,10];

$$\mu_i(x) = \frac{1}{N}\sum_{t=1}^{N} I_i^t(x) \tag{3}$$

$$\sigma_i(x) = \underset{t=1,\ldots,N-1}{\arg\max} |I_i^{t+1} - I_i^t| \tag{4}$$

$\mu_i(x,y)$, and $\sigma_i(x,y)$ are calculated for each channel ($i \in \{\mathrm{R,G,B}\}$) and put into (1) to find an approximation for $p(I(x,y))$.

Using the model parameters in (3) and (4), a binary change map which shows the pixels that have been changed can be created using the following formula:

$$\mathrm{CM}(x,y) = \begin{cases} 1 & \left(\sum_{i=\{\mathrm{R,G,B}\}} B_i(x,y)\right) \geqslant 2 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$B_i(x,y) = \begin{cases} 1 & (|\mu_i(x,y) - I_i(x,y)| \geqslant \alpha_i\sigma_i(x,y)) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\mathrm{CM}(x,y)$ is a binary change detection map. Value 1 shows a change in spatial location of (x,y), and 0 shows no change, meanwhile $B_i(x,y)$ tries to estimate the change in color channels, and α_i is the global constant, which affects the final change detection map, and its value is settled with trial and error method. Eq. (5) assumes that, if there are changes at least in two color channels at spatial location of (x,y) it is likely that there is a change in (x,y).

2.2. Adaptation of model parameters

In general, the dynamics of the scene observed changes with time by changing illumination, or with other natural effects. It is required to adapt the model parameters with these changes. A simple adaptation of corresponding pixel's parameters can be formulated as follows [7];

$$\begin{aligned} \mu_i^t(x,y) &= \beta_i\mu_i^{t-1}(x,y) + (1-\beta_i)I_i^t(x,y) \\ \sigma_i^t(x,y) &= \beta_i\sigma_i^{t-1}(x,y) + (1-\beta_i)|I_i^t(x,y) - \mu_i^t(x,y)| \end{aligned} \tag{7}$$

where $\mu_i^t(x,y)$, $\mu_i^{t-1}(x,y)$ and $\sigma_i^t(x,y)$, $\sigma_i^{t-1}(x,y)$ are mean values and standard deviations at times t and $t-1$ respectively, β_i is some constant for adapting model parameter in ith color channel. Keeping it small gives more weight to current time, less weight to past model parameters, and vice versa. Its value is settled by trial and error method with respect to scene dynamics.

2.3. Permanent changes in background

Occurrence of permanent changes in the scene is possible, and if this change can not be detected by the system, the system always gives false positive which is undesirable. It is observed that, if the change stays in the system longtime which is in the unit of frame, i.e. if it stays 100 frames, then the change in the system can be added to the background model if we keep a counter for each pixel which keeps how many consecutive frames it stays as background and decide on corresponding pixel using this counter. This counter value is used to decide whether the corresponding pixel should be added to background model or not. The pixel's counter value is compared with a predefined threshold value τ which is a global threshold used to decide whether corresponding pixel should be registered as background pixel which is formulated in (8).

$$S(x,y) = \begin{cases} \text{background}, & C(x,y) \geqslant \tau \\ \text{foreground}, & \text{otherwise} \end{cases} \tag{8}$$

where $S(x,y)$ is a binary map which shows whether corresponding pixel is foreground (with binary value of 1) or background pixel (with binary value of 0), $C(x,y)$ is the corresponding pixel's counter value that keeps for how many consecutive frames it stays as foreground and τ is the global threshold which keeps a counter threshold value to decide change from foreground to background.

The value of τ is settled with respect to dynamics of the environment. The parameter is adaptive and should be changed by the user in different applications depending on the dynamic changes of the environment. If the object movements are fast, then τ can be kept small, however if the movements are slow τ should be kept large.

Algorithm given in Fig. 1 is applied to each pixel in the image taken. The pixel data, which is the raw value,

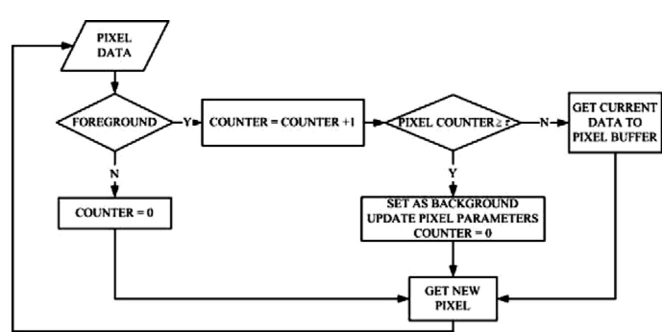

Fig. 1. Permanent scene change detection algorithm.

captured, is classified as the foreground or background with respect to model parameters for the corresponding pixel. If it is classified as foreground pixel, then it is possible that pixel belongs to permanent change in background. Pixel's counter which keeps how many consecutive frames a foreground object stay at spatial location of (x,y), is compared with global threshold τ, if the pixel's counter is equal to τ then; the corresponding foreground pixel is settled as background pixel with model parameters settled over pixel values captured while pixel's counter value is less than τ. While the counter is increasing, if the corresponding pixel's status changes from foreground to background then, the pixel's previous parameters are not changed. This means that, the object is no longer moving in the scene.

As we mentioned before, settling the values for $\alpha_i,\beta_i i \in \{\mathrm{R},\mathrm{G},\mathrm{B}\}$ are completely heuristic, and there are many combinations which can be used, and for simplicity we assume that $\alpha_\mathrm{R}=\alpha_\mathrm{G}=\alpha_\mathrm{B}=\alpha$ and $\beta_\mathrm{R}=\beta_\mathrm{G}=\beta_\mathrm{B}=\beta$. Using this assumption we carry out an experiment in which a camera is placed over an area with no motion and the number of misclassified pixels is counted. The performance criterion is evaluated as follows:

$$J(\alpha,\beta)=\frac{\sum_{\mathrm{all}(x,y)}\mathrm{CM}(x,y)}{\sum_{\mathrm{all}(x,y)}1} \tag{9}$$

where (x,y) refers to spatial locations of current image. The experiment is firstly carried out with fixed $\beta=0.85$ and an estimate for α is found. One thousand frames are captured for each specific α and an average $J(\alpha,\beta)$ is calculated. Fig. 2 shows a graph with different values of α for a specific scene.

The value of α saturates after 1.75. The experimental results, which carried out using different experimental environments, show that the value α is always around 2.0.

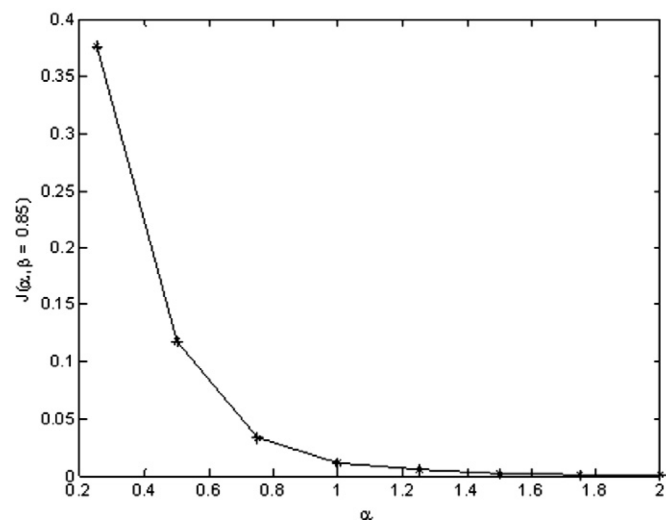

Fig. 2. Effect of α with $\beta=0.85$. (a) Scene over which experiment is carried out. (b) Average $J(\alpha,\beta=0.85)$ with respect to α.

3. Statistical fire color model

In [6], Philips et al. use a look-up table to identify fire in video sequences using a manual training set which is created using the same video. This look-up table is used to identify the possible fire pixels. Since our model should run on real-time video we cannot use such a method, rather we try to model a global generic model for fire colors. Color clues that identify the fire are combined with change detection map $\mathrm{CM}(x,y)$ in order to identify the fire pixels in a video sequence. In this part, a generic model for fire pixels is proposed, and its performance on still images is shown.

3.1. Statistical color model for fire recognition

A set of 150 different resolutions images is collected from Internet containing diversity in illumination and camera effects and segmented into fire and non-fire regions. 16,309,070 pixels are used to create a statistics. With this strategy, a set of rules is generated; these rules are used as backbone for generic fire model.

It is noticed that, for each pixel in fire blob, the value of Red channel is greater than the Green channel, and the value of Green channel is greater than the value of Blue channel and it is further noticed that mostly the fire color is the color with high saturation in Red channel. So for a pixel located at spatial location of (x,y), the first condition must be as follows:

$$\mathrm{R}(x,y)>\mathrm{R}_{\mathrm{mean}} \tag{10}$$

$$\mathrm{R}_{\mathrm{mean}}=\frac{1}{K}\sum_{i=1}^{K}\mathrm{R}(x_i,y_i) \tag{11}$$

$$\mathrm{R}(x,y)>\mathrm{G}(x,y)>\mathrm{B}(x,y) \tag{12}$$

where $\mathrm{R}(x,y)$, $\mathrm{G}(x,y)$, and $\mathrm{B}(x,y)$ are Red, Green and Blue values for a pixel located spatially at (x,y), K is the total number of pixels in image, and $\mathrm{R}_{\mathrm{mean}}$ is the mean of Red channel of pixels, where a change is not detected, caused by the temporary object motions, which are foregrounds. It is dictated in (10)–(12) that a fire pixel should have a larger red component value than the mean of the pixels in the red channel of the frames where a change is not detected.

Human can easily detect fire in images even in bad illumination conditions. On the other hand, change in illumination conditions fails most of the algorithms proposed. There are number of illumination compensation algorithms, however they are expensive in computational complexity. It is well known that, when illumination is changed, both luminance (intensity) and the chrominance change. Instead of using only luminance values, chromaticity can be employed. RGB represents not only the chrominance but also the luminance of the pixels. There are lots of color-transformation algorithms that transform RGB color space to other color spaces [9], which can be used to separate the luminance from the chrominance. However the luminance component of the RGB color space can be eliminated by color channel ratios. Depending on the application different ratios can be formulated to generate luminance independent values containing normalized chrominance information. In our application we use R–G, R–B, and G–B ratios as formulated in (13), which are then used for color based fire classification.

$$\begin{aligned} 0.25 &\leqslant \mathrm{G}(x,y)/(\mathrm{R}(x,y)+1) \leqslant 0.65 \\ 0.05 &\leqslant \mathrm{B}(x,y)/(\mathrm{R}(x,y)+1) \leqslant 0.45 \\ 0.20 &\leqslant \mathrm{B}(x,y)/(\mathrm{G}(x,y)+1) \leqslant 0.60 \end{aligned} \tag{13}$$

The upper and lower bounds for inequalities are estimated by using the hand labeled data set. The rules defined in (10), (12) and (13) are used to generate an overall binary map which is called Fire Map (FM). $\mathrm{FM}(x,y)$ is a binary value indicating whether the pixel located at spatial location (x,y) is classified as fire (binary value of 1) or non-fire (binary value of 0) pixel.

Figs. 3–5 show fire detection with rules on still images where overall still image is assumed to be foreground object. Fig. 3 shows the fire detection mechanism for an image cropped from a forest area, where there is fire that surrounds some parts of trees. Fig. 3(a) shows the original image over which our generic model is applied to segment fire pixels. Fig. 3(b) shows the binary image, where pixels that in line with the rule (10) are labeled with white. It is remarkable that, the assumption for the fire pixels having higher red components than the surrounding environment holds, but produces to many false positives. On the other hand, when rule (12) is employed to complement rule (10), false positives are reduced, where this scenario is depicted in Fig. 3(c). Still there are some false positives over resultant binary image that is formed with rules (10) and (12). The final step is the use of statistics rule (13), and its effect is shown in Fig. 3(d). The resultant binary image shows that rules through (10), (12) and (13) are suitable for efficient segmentation of the fire.

Figs. 4 and 5 show that the fire information from still images can be easily detected using the rules in (10), (12) and (13). The segmentation results can be further processed using morphological operations of dilation and erosion [9]. Images consisting of no flame but flame-like colors can be segmented as fire. In Figs. 4 and 5, there are red objects in the image, which are segmented as the fire regions after using the rules in (10) and (12). After incorporating the rule in (13) into the final segmentation, almost all non-fire pixels are cleared except some pixels, which are very similar to fire pixels (i.e. the pixels given in Fig. 5). These small mistakes can be compensated using a hybrid system, which combines, motion and color which will be explained in the preceeding sections.

4. Combining color and background subtraction

It is observed that the motion of flames in consecutive frames should show a deviation in shape, which is mainly caused from the burning material or the wind in the environment. The flame can be thought as a moving object. The motion of the flame object may be sudden in the case of explosive fire. The type of the motion changes from event to event, but there is only one thing, which does not change, that is the change of size and motion of fire in the consecutive frames.

Because of the fire-like color of the sun, sometimes it is likely to detect the reddish color in horizon as fire or other kind of effects may produce such affect. The case mentioned is compensated using background subtraction procedure, which mainly subtracts background from foreground changes and adapts the background model with time, so that fire-like colors will be removed.

The background subtraction process is the first step in our algorithm. The changes detected by the background subtraction stage are supplied to color verification process. Pixels that are detected by the algorithm as foreground object, and have a fire-like color classified by the rules defined are grouped into blobs with respect to their spatial connectivity. A time analysis of each fire-like blob is considered, and if it grows in size or changes its center location, then each fire blob is considered as a fire candidate. The algorithm of the proposed fire detection is summarized in Fig. 6.

The first step of the algorithm removes the background and detects possible foreground objects that are mainly caused from either temporal changes in the background or an object motion into the scene. The second step is applied if foreground pixels detected with fire-like colors. The output of this step mainly removes foreground objects, which do not have fire-like colors. There are some pixels, which are classified as foreground fire-like objects caused from the noise. In order to remove such a noise, we remove connected component pixel groups of size less than 5 pixels. Second step is followed by the third step that aims the detection of foreground blobs where each blob is detected using connected component labeling algorithm [9]. In connected component labeling algorithm 4-connectivity is used. Detection of each blob is followed by construction of guard area which is rectangular area that covers each blob and used to observe the behavior of

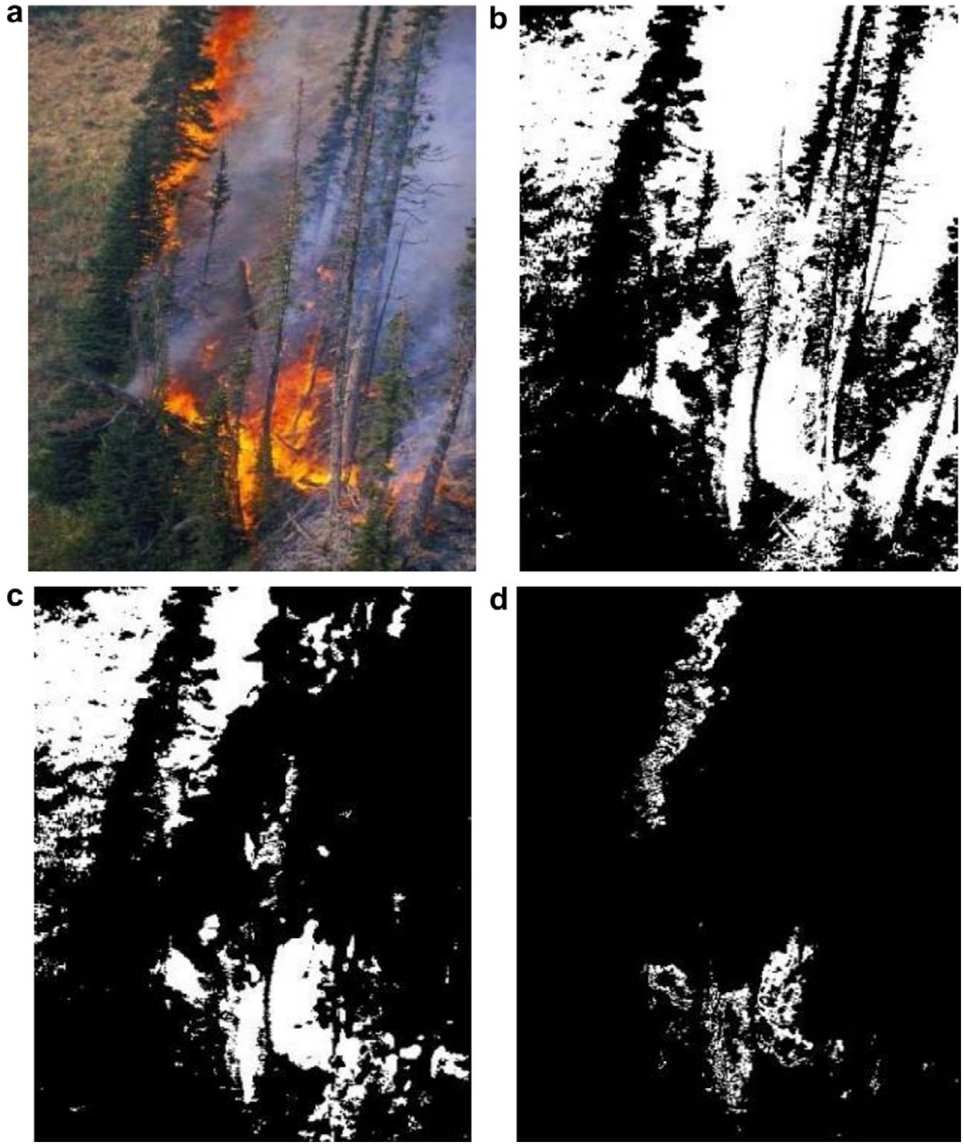

Fig. 3. Fire pixel segmentation in a still image using generic fire model. (a) Original image, (b) fire segmentation using only (10), (c) fire segmentation using (10) and (12) and (d) fire segmentation using (10), (12) and (13).

enclosed blob in consecutive frames in order to decide whether it is fire object or not. In each guard area two measures are carried out; the first one is the spatial mean of the blob in guard area, which is used to measure the behavior of fire which should be changing because the fire has property of swinging. The second measure is spatial area of detected pixels in guard area. It should be either getting larger or smaller in consecutive frames. Fig. 7 shows a foreground object and corresponding guard area. Size of guard area is larger than the size of blob, and it is found using the following equation;

$$\frac{w_g}{w_b} = \frac{h_g}{h_b} = 2.0 \tag{14}$$

where w_b and h_b are width and height of corresponding blob respectively, and w_g and h_g are width and height of guard area. In (14), the ratio of rectangular sides should be equal to 2, and this number can be changed with image size.

Fig. 8 shows the step-by-step visualization of the algorithm. In the first row, background is shown with its binary maps of change detection map (column (b)), fire color detection map (column (c)), and detected fire map respectively (column (d)). The second row shows that there is a foreground object but no fire. The third row and rest of the rows show that there is a foreground object, which is fire.

5. Performance analysis and computational complexity

In order to analyze the system performance of the proposed algorithm, we have recorded 10 video sequences. Each video sequence has 1,00,000 video frames of size 176×144 with a frame rate of 40 fps.

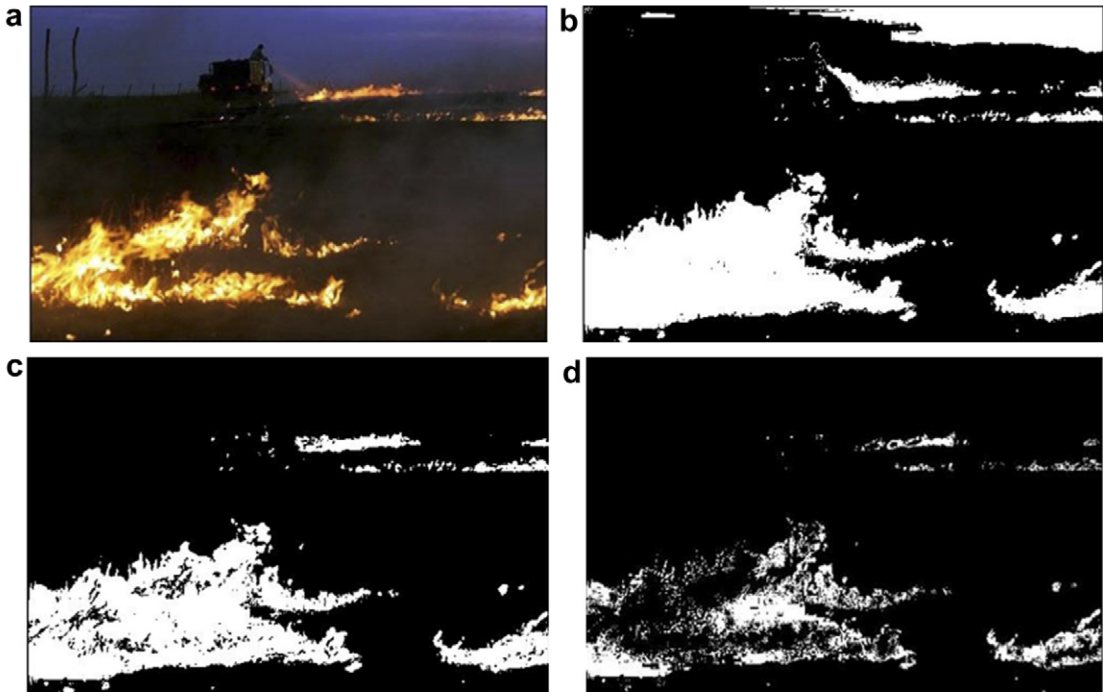

Fig. 4. Fire pixel segmentation in a still image using generic fire model. (a) Original image, (b) fire segmentation using only (10), (c) fire segmentation using (10) and (12) and (d) fire segmentation using (10), (12) and (13).

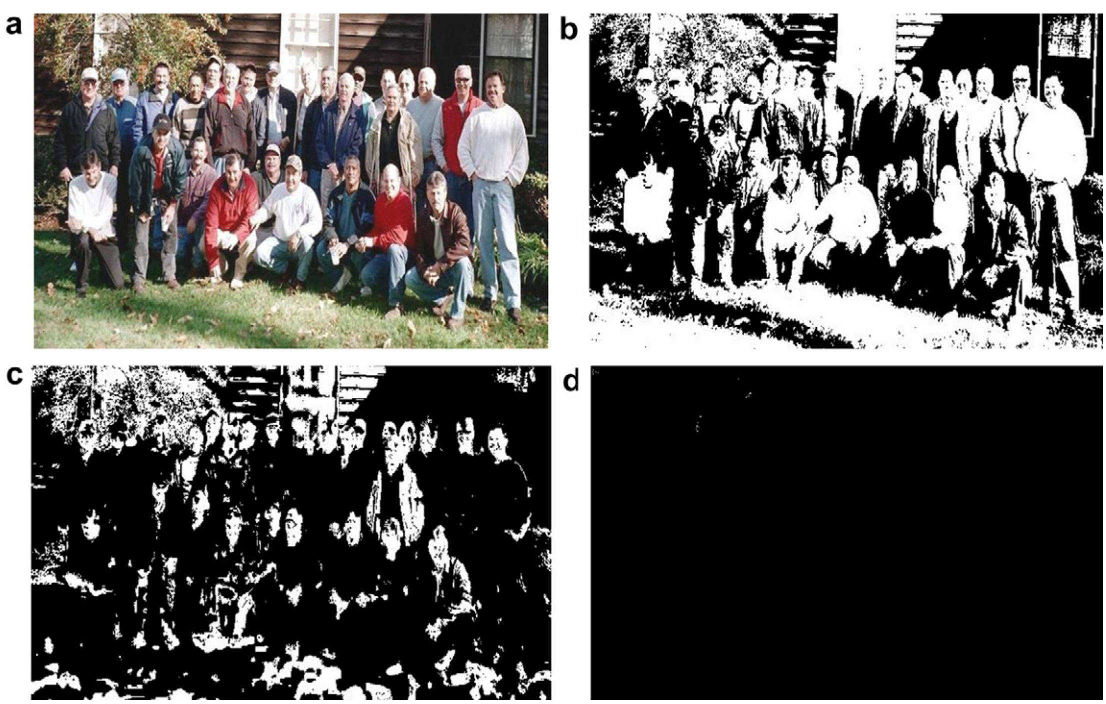

Fig. 5. Fire pixel segmentation in a still image which contains fire-like colors using generic fire model. (a) Original image, (b) fire segmentation using only (10), (c) fire segmentation using (10) and (12) and (d) fire segmentation using (10), (12) and (13).

In order to test the effect of different lighting conditions over algorithm, the size of video sequences are kept large intentionally. The camcorder used is a commercial and cheap camcorder with no special functionality. We have recorded sequences by using the following conditions;

- The camcorder should have a fixed position, so that there is no need for image registration to align consecutive frames.
- Some of sequences should contain scenes, which consist of fire like regions (false positives), i.e. Sunset, sunrise scene.
- Sequences should be recorded in different times in a day.

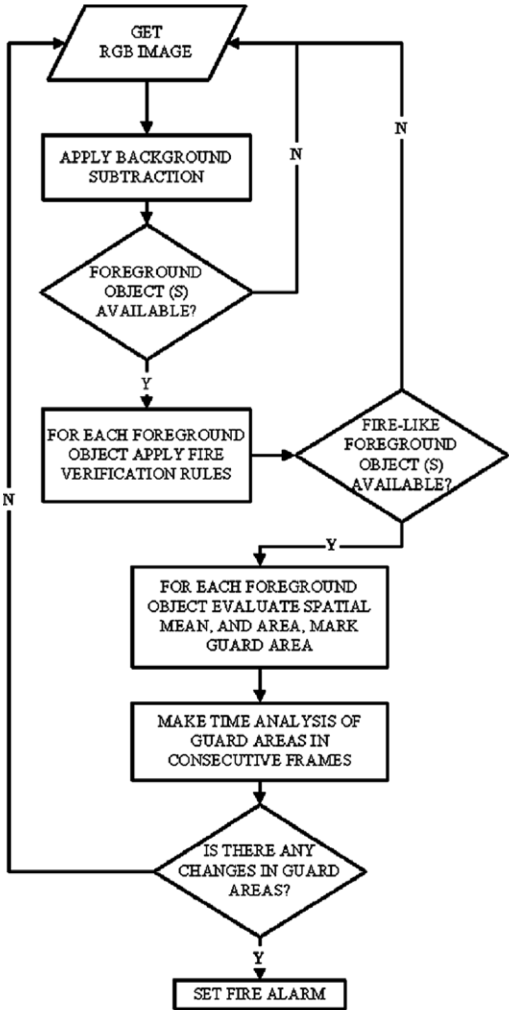

Fig. 6. Proposed fire detection algorithm flow chart.

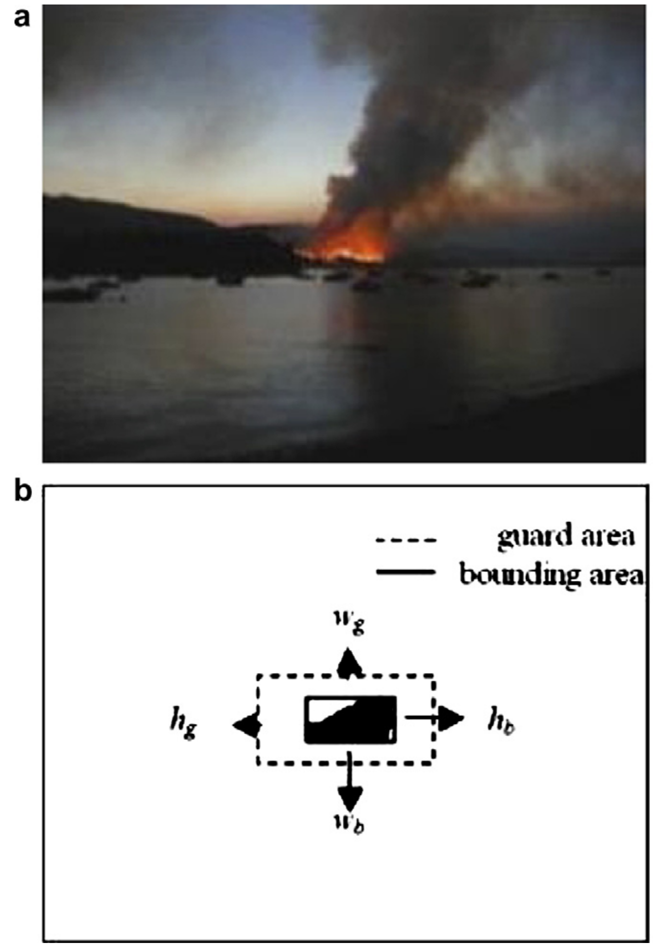

Fig. 7. Foreground object and corresponding guard area: (a) original image and (b) fire detector output and corresponding guard area.

Since video sequences are too many, we have sampled them with a sampling rate of 100 frames. Sampled video sequences are used to test the performance of our algorithm.

We used the following performance criterion. First the algorithm is applied on sampled video sequences. Then the detected fire frames are counted manually, and we have compared them to the original frames. If the algorithm raises the alarm flag for a frame with no fire in it, then there is an error occurred. The correct detection rate is on the order of 98.89%, which we assume to be an acceptable level for fire detection. The 1.11% error in detection rate is mainly caused from camera non-linearities, and sudden change in lighting conditions, and also some kind of materials producing different fire colors while they are burning.

In this part, the computational complexity analysis of the overall system is provided. The complexity analysis is based on the worst case scenario. Table 1 tabulates number of addition/subtraction, multiplication/division, and comparisons. Let us assume that the captured image is in the size of $R \times C$ (R rows, and C columns), N training images are used for model parameter estimation, and τ is assumed to be infinity, then the overall algorithm consumes the following computational power. The total number of addition/subtraction is $3RC + (RC - 1)$, multiplication/division is $21RC + 130$ and comparisons are $22RC$. It is noticeable that overall algorithm has a complexity of $O(RC)$ which is linear with size of captured image. Table 1 depicts that overall computational complexity of the system is low enough for real-time applications.

6. Conclusions

In this paper, we have developed a real-time fire-detector, which combines color information with registered background scene. Color information of fire is determined by the statistical measurement of the sample images containing fire. Simple adaptive background information of the scene is modeled using three Gaussian distributions, where each of them is used to model the pixel values of the colored information in each color channel. The foreground objects detected are combined with color statistics

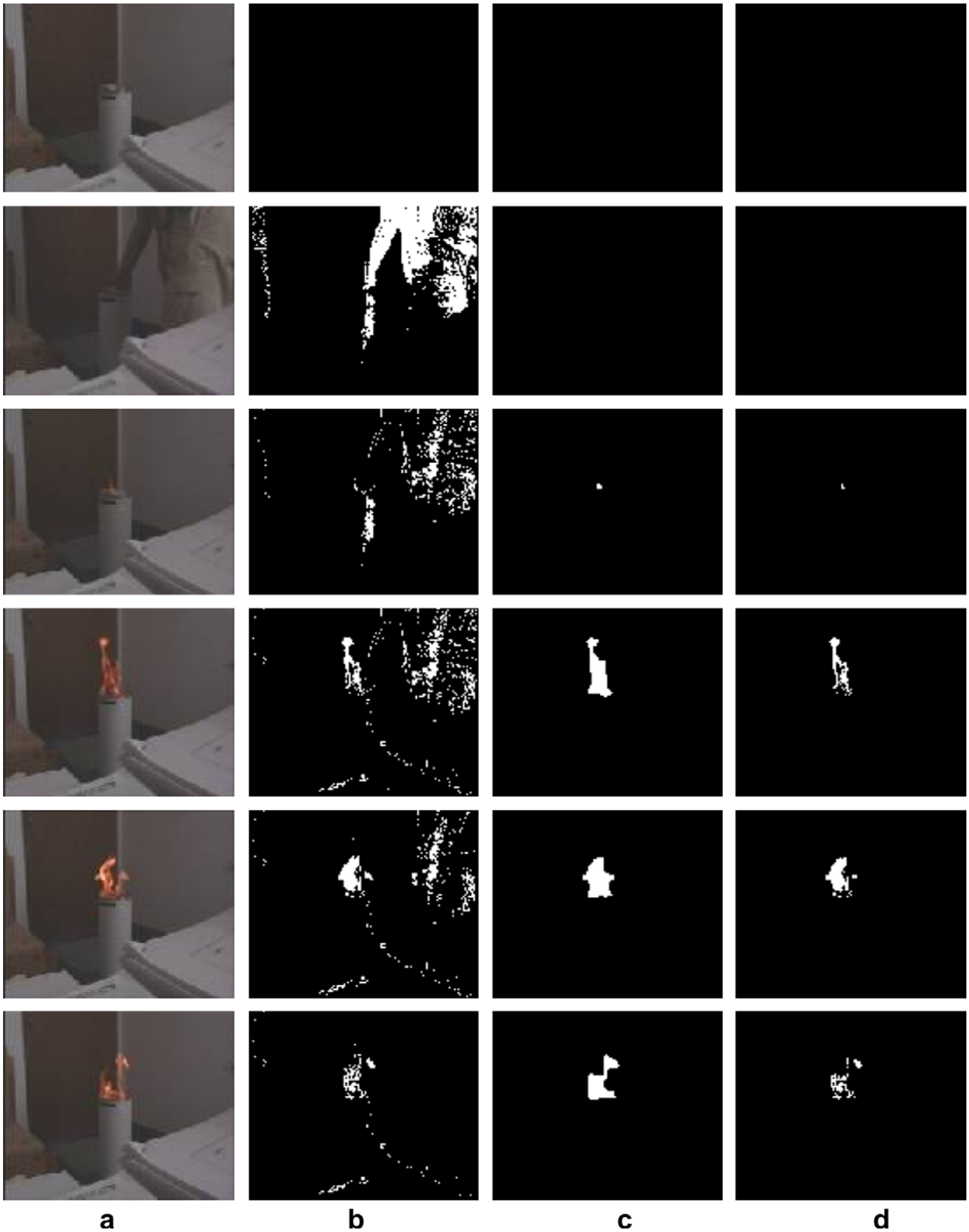

Fig. 8. Experimental results: (a) is input image, (b) is change map, (c) is fire filtered binary map of input image and (d) is detected fire.

Table 1
Computational complexity analysis with full load in the system

Section	Addition/subtraction	Multiplication/division	Comparisons
Estimation of model parameters	3RC $(2N-2)$	3RC	3RC $(N-2)$
1. Change map generation	5RC	3RC	7RC
2. Adaptation of model parameters	9RC	12RC	3RC
3. Permanent changes in background	RC		RC
4. Statistical color model for flame recognition	3RC + (RC − 1)	3RC + 2	9RC
5. Combining color and background subtraction	3RC	128	2RC
Total (1 + 2 + 3 + 4 + 5) for one frame	22RC − 1	18RC + 130	22RC

and output is analyzed in consecutive frames for fire detection. The algorithm's computational complexity is low enough to produce 40 fps with image size of 176×144 colored image over Pentium 4 CPU 1.80 GHz PC. The correct detection rate for our algorithm is 98.89%. This level can be increased with sophisticated background subtraction algorithm, and fire color model, which would make the system more complex.

The system detects the fire as soon as it is started, except in the explosive conditions, in which generally smoke is seen before the fire is started. The proposed algorithm can be extended to incorporate the smoke in the video sequences, which may be used as faster fire alarm detection in such special conditions.

References

[1] T. Cleary, W. Grosshandler, Survey of fire detection technologies and system evaluation/certification methodologies and their suitability for aircraft cargo compartments. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1999.

[2] W. Davis, K. Notarianni, NASA fire detection study. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1999.

[3] G. Healey, D. Slater, T. Lin, B. Drda, A.D. Goedeke, A system for real-time fire detection. in: Computer Vision and Pattern Recognition, 1993, Proceedings CVPR '93, 1993 IEEE Computer Society Conference, 15–17 June 1993, pp. 605–606.

[4] S.Y. Foo, A rule-based machine vision system for fire detection in aircraft dry bays and engine compartments, Knowledge-Based Systems 9 (1995) 531–541.

[5] O.A. Plumb, R.F. Richard, Development of an Economical Video Based Fire Detection and Location System. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1996.

[6] Water Philips III, Mubarak Shah, Niels da Vitoria Lobo, Flame recognition in video, Pattern Recognition Letters 23 (2002) 319–327.

[7] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: Real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997).

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

[9] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ, 2002.

[10] T. Çelik, T. Kabakli, M. Uyguroğlu, H. Özkaramanlı, H. Demirel, Automatic threshold selection for automated visual surveillance, in: SIU'2004, IEEE 12th Signal Processing and Communications Applications Conference, 28–30 April 2004, Kusadasi, Turkey, pp. 478–480.

[11] L.E. Neily, R.A. Neville, W.D. McNoll, Application of real time digital image analysis and enhancement to airborne infrared fire detection and mapping, in: 12th Canadian Symposium on Remote Sensing, vol. 14, 13 July 1989, pp. 2610–2610.

[12] H. Yamagishi, J. Yamaguchi, Fire flame detection algorithm using a color camera, in: Proceedings of 1999 International Symposium on Micromechatronics and Human Science, 1999, pp. 255–260.

[13] H. Yamagishi, J. Yamaguchi, A contour fluctuation data processing method for fire flame detection using a color camera, 26th Annual Conference of the IEEE Industrial Electronics Society 12 (2000) 824–829.

[14] E. Cisbani, A. Bartoloni, M. Marchese, G. Efisei, A. Salvati, Early fire detection system based on multi-temporal images of geostationary and polar satellites, IEEE International Conference on Geosciences and Remote Sensing Symposium 13 (2002) 1506–1508.

[15] Shaohua Chen, Hong Bao, Xianyun Zeng, Yimin Yang, A fire detecting method based on multi-sensor data fusion, IEEE International Conference on Systems, Man and Cybernetics 14 (2003) 3775–3780.

[16] J.N. Moutinho, F.D. Mesquita, N. Martins, R.E. Araujo, Progresses on the design of a surveillance system to protect forests from fire, IEEE Conference on Emerging Technologies and Factory Automation 12 (2003) 191–194.