14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

AUTOMATIC FIRE DETECTION IN VIDEO SEQUENCES

T. Celik, H. Demirel, H. Ozkaramanli
Department of Electrical and Electronic Engineering, Eastern Mediterranean University,
email: turgay.celik@emu.edu.tr

ABSTRACT

In this paper, we propose a real-time fire-detector which combines foreground information with statistical color information to detect fires. The foreground information which is obtained using adaptive background information is verified by the statistical color information which is extracted using hand labeled fire pixels to determine whether the detected foreground object is a candidate for fire or not. The output of the both stages is analyzed in consecutive frames which is the verification process of fire that uses the fact that fire never stays stable in visual appearance. The frame processing rate of the detector is about 30 fps with image size of 176x144 which enables the proposed detector to be applied for real-time applications.

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.

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 with fire, especially sun.

In [3] and [4] previous vision-based methods are presented, which rely upon ideal conditions. In [1], color and motion are used to classify regions as fire or nonfire. However this method requires manual camera initialization. Statistical methods are used in [2] to detect the fire in grayscale video taken with high-speed cameras, which is computationally expensive. Specialized point based thermal sensors are used in [5] along with a grayscale camera to observe intensity changes in temperature. This method requires sensors which are expensive and the exact position of the sensors must be calibrated for method 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. An initial training set is used to create a look-up table and in the absence of the initial training set a generic look-up table is formed. However the details of how to create the generic look-up table are not given.

In [11] and [12] first the contour of the flame area is extracted using statistical HSV color space. Then the extracted contour data is transformed into the polar coordinate. The extracted polar coordinates of every input image are placed in time series. A fluctuation data is extracted, as a space-time data on the contour. A frequency domain pattern of the fluctuation data is obtained and the pattern is processed in a neural network, which gives good results but the computational complexity is too high to be used in real-time sequences. In [13], a combination of the surround temperature, smoke density, and 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 this paper, an algorithm is proposed which combines color information of fire with temporal changes in video sequences. The background subtraction is assisted by the foreground objects to detect fire.

2. BACKGROUND MODELING

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 images containing Luminance, ChromaBlue and ChromaRed (YUV) components. Initially the acquired images are 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 training phase of the system. So for each pixel, an overall probability model is estimated as follows;

$$p(I(x,y)) = p_R(I_R(x,y))p_G(I_G(x,y))p_B(I_B(x,y)) \quad (1)$$

where p_R, p_G, and p_G are distributions 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;

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

where $I_i(x,y)$ is the value of $I(x,y)$ in i^{th} 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 $i \in \{R,G,B\}$. The model parameters of $\mu_i(x,y)$ and $\sigma_i(x,y)$ should be estimated.

2.1. Estimation of Model Parameters and Change Map

Let us have N frames for the training period. The estimation of $\mu_i(x,y)$ is straight forward, but $\sigma_i(x,y)$ 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 and extra cost to the system. This drawback could be overcome using the idea in [8] where $\sigma_i(x,y)$ is found using maximum absolute difference between consecutive frames which requires only storage of $I_i^{t+1}(x,y)$ and $I_i^t(x,y)$, which are the 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 in [8], [10] by,

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

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

$\mu_i(x,y)$, and $\sigma_i(x,y)$ are evaluated for each channel ($i \in \{R,G,B\}$) 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:

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

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

$B_i(x,y)$ shows the change in color channels, and α_i is the global constant which affects final change detection map. Its value is determined by trial and error method. Equation (5) implies 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 illumination changes, or other natural effects. Therefore, it is required to adapt model parameters with these changes. A simple adaptation of corresponding pixel's parameters can be formulated by the following formula given in [7];

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

where μ_i^{t+1}, μ_i^t and $\sigma_i^{t+1}(x,y)$, $\sigma_i^t(x,y)$ are mean values and standard deviations at times t + 1 and t respectively, β_i is some constant for adapting model parameter in i^{th} color channel.

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 long-time (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. The counter value is used to decide whether the corresponding

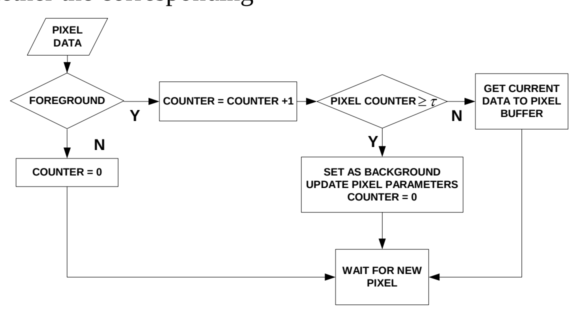


Figure 1: Permanent scene change detection algorithm

pixel should be added to the background or not. The pixel's counter value is compared with a predefined threshold value τ which is a global threshold used to decide whether a pixel should be accepted as background pixel or not with respect to its current counter value which is formulated in (8).

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

where $S(x,y)$ is the state of the corresponding pixel, $C(x,y)$ 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 and this is illustrate in Figure 1. As we mentioned before, settling the values for $\alpha_i, \beta_i \; i \in \{R, G, B\}$ are completely heuristic, and there are many combinations which can be used. For simplicity, we assume that $\alpha_R = \alpha_G = \alpha_B = \alpha$ and $\beta_R = \beta_G = \beta_B = \beta$. The experimental results carried out using different experimental environments, show that the value α is always around 2.0.

3. STATISTICAL 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 possible fire pixels. Since our model should run on real-time video we can not 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 *CM* in order to identify the fire pixels.

3.1. Statistical Color Model for Fire Recognition

A set of images have been collected and they are segmented into fire and non-fire regions. The statistics of fire regions are used to generate a rule base for the fire detection. We have noticed that, for each pixel in a fire blob, the value of the Red channel is greater than the Green channel, and the value of Green channel is greater than the value of Blue channel. Furthermore, it is 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, the first condition must be as follows:

$$R(x, y) > R_{mean} \tag{9}$$

$$R(x, y) > G(x, y) > B(x, y) \tag{10}$$

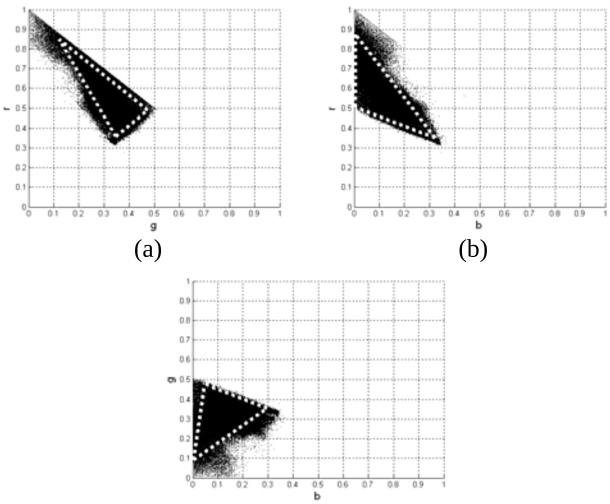
$$R_{mean} = \frac{1}{K} \sum_{i=1}^{K} R(x_i, y_i) \tag{11}$$

where $R(x, y)$, $G(x, y)$, and $B(x, y)$ are Red, Green and Blue values of pixel (x, y), K is the total number of pixels in image, and R_{mean} is the mean of Red channel of pixels where a change is not detected. It is dictated in (9),(10),(11) that the fire should be redder than the mean of the red components of the image where a change is not detected.

Human-being can easily detect fire in images even in bad illumination. There are lots of color transformation techniques that rely upon intensity of the pixels. If the illumination, with which a chromaticity based fire model is changed than the model fails. It is well known that, the normalized *rgb* color space is more robust with respect to illumination changes, and normalize *rgb* is found as follows [9]:

$$\begin{aligned} r &= R/(R+G+B) \\ g &= G/(R+G+B) \\ b &= B/(R+G+B) \end{aligned} \tag{12}$$

We have collected 1000 fire pictures from Internet over them there is diversity in illumination and camera effects. We have manually classified fire regions in 1000 fire pictures and histogram of 16,309,070 pixels are created in *r-g*, *r-b*, and *g-b* planes. Figure 2 shows distribution of fire pixels in *r-g*, *r-b*, and *g-b* planes. It is noticeable in Figure 2 that, we can classify fire pixel regions in three planes by using three linear lines which construct rectangular are and are depicted as blue lines in Figure 2, and we can use these regions in order to classify corresponding pixel is fire pixel or not.



(a) (b) (c)

Figure 2: Fire pixels distributions in (a) *r-g* plane (b) *r-b* plane and (c) *g-b* plane.

The regions in Figure 2 can be classified as follows, the region in Figure 2 (a), (b) and (c) are triangular regions and classified as in (13.a), (13.b) and (13.c) respectively.

$$\begin{aligned} r &\geq 1.1403g - 0.0759 \\ r &\leq -0.9889g + 0.9913 \\ r &\geq -2.0771g + 1.0251 \end{aligned} \tag{13.a}$$

$$\begin{aligned} r &\leq -1.8813b + 0.95824 \\ r &\leq 95.3478b + 0.1707 \\ r &\geq -0.5427b + 0.5063 \end{aligned} \tag{13.b}$$

$$\begin{aligned} g &\geq 0.8459b + 0.0482 \\ g &\leq -0.4608b + 0.4954 \\ g &\leq 191.9565b - 0.6207 \end{aligned} \tag{13.c}$$

Using combination of (9), (10), and (13), fire pixels are segmented. Note that, while generating region in (13), triangle borders are selected in a manner to remove most of the boundary pixels, which are mostly misclassified because of uncertainty in visual manual classification, that's why we select triangle a little bit narrower than overall coverage area. In Figure 3, we show the fire pixel segmentation in still image in a step by step manner. Figure 3 (a), is the original still image which consist of fire or not, Figure 3 (b) is the binary image where fire pixels are marked as white and non-fire pixels are marked black using only rule defined in(9), Figure 3 (c),(d) are fire pixel segmentation using rules defined in (10) and (13) respectively, and overall segmentation result, which is the combination of rules defined in (9),(10) and 13 is shown in Figure 3 (e).


(a)


(b)


(c)


(d)


(e)

Figure 3: Fire detection in a still image; (a) Original image (b) Fire segmentation using only (9) (c) Fire segmentation using (10) (d) Fire segmentation using (13) (e) Fire segmentation using combination (9),(10) and (13)

4. COMBINING MOTION INFORMATION WITH FIRE COLOR MODEL

The motion of flames in consecutive frames should show a deviation which is mainly affected by the burning material or the wind in the environment. The type of the motion changes from event to event, but there is only one thing which does not change which is the change of size and motion of fire in the consecutive frames.

Because of the fire-like color of sun, sometimes it is likely to detect the reddish color in horizon or fire-like colored objects as fire. The case mentioned is compensated using background subtraction procedure, which mainly subtracts background from foreground changes and adapts the background model with time so sun affected colors will be removed from the system in order to detect fire.

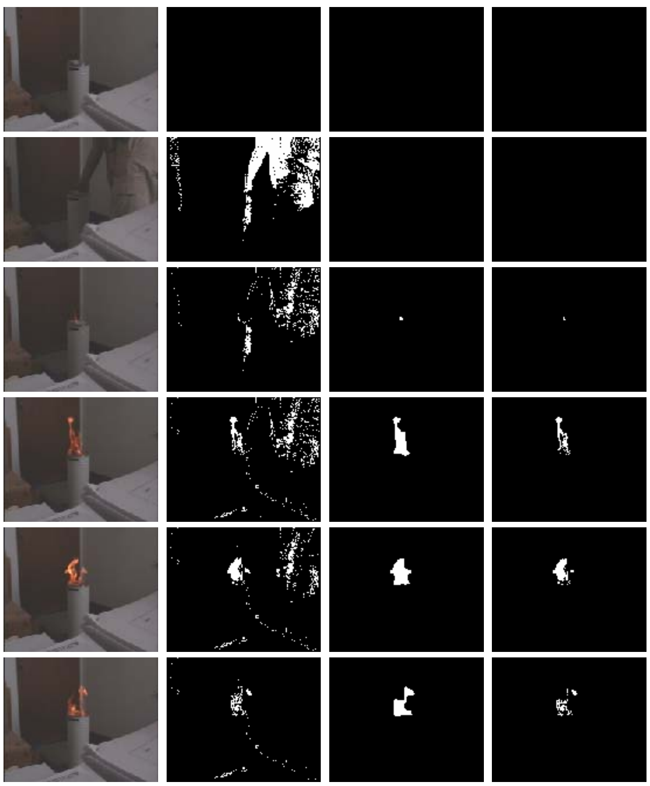
The algorithm of the proposed fire detection method is as follows;

1) Apply background subtraction
2) For each foreground pixel, check rules in (9),(10) and (13)
3) Apply morphological erosion and dilation operations
4) Detect blobs
    a. Evaluate spatial mean of each detected blob
    b. Evaluate spatial area of each detected blob
    c. Mark a guard area (area enclosing the blob) over detected blob
5) If detected blobs' mean and spatial areas change in consecutive frames in guard area, then a fire is detected.

The first step of the algorithm removes the background and detects possible foreground motions that are mainly caused by either temporal changes in the background or an object motion into the scene. The second step is applied if foreground pixels detected are kind of fire-like colors. The output of this step mainly removes foreground objects which doesn't have fire-like colors. Third step is applied in order to remove noise in change detection map where it assumes that the change should be larger than two pixels in size. The order of morphological operations is first binary erosion and then dilation which is called binary closing operation [9]. In fourth step the blobs are detected using connected component labeling (CCL) algorithm [9]. In CCL algorithm 4-connectivity is used. Detection of each blob is followed by the construction of a guard area which is a rectangular area that covers each blob and used to observe the behavior of enclosed blob in consecutive frames in order to determine 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 identity 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, which should be either getting larger or smaller in consecutive frames. Size of guard area is larger than the size of blob, and it is found using the following equation;

$$\frac{w_g}{w_b} = \frac{h_g}{h_b} = 1.25 \tag{14}$$

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 can be changed with image size. Figure 4 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 there is a foreground object which is fire.



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

Figure 3: Experimental Results ; (a) is input Image, (b) is change map, (c) is fire filtered binary map of input image, (d) is detected fire

5. 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. Color histograms of *r-g*, *r-b* and *g-b* are used to extract fire regions over which a possible fire pixel is defined. Simple adaptive background information of the scene is modeled by using three Gaussian distributions, where each of them is used to model the pixel values in each color channel. The foreground objects detected are combined with color information and output is analyzed in consecutive frames to detect fire. The frame processing rate of the detector is about 30 fps with image size of 176x144 which enables the proposed detector to be applied for real-time applications.

REFERENCES

[1] Cleary, T., Grosshandler, W., 1999. 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.

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

[3] Healey, G.; Slater, D.; Lin, T.; Drda, B.; Goedeke, A.D, **“**A system for real-time fire detection**”.** Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on 15-17 June 1993 Page(s):605 – 606.

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

[5] Plumb O.A., Richard, R.F., 1996. Development of an economical video based fire detection and location system. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.

[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 and A. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” IEEE Trans. Pattern Analysis and Machine Intelligence vol. 19, no. 7, July 1997.

[8] Haritaoglu, I.; Harwood, D.; Davis, L.S., “W4: real-time surveillance of people and their activities,” Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22, Issue 8, Aug. 2000 Page(s):809 – 830.

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

[10] T. Celik, T. Kabakli, M. Uyguroglu, H. Ozkaramanli, H. Demirel, ”Automatic Threshold Selection for Automated Visual Surveillance”, SIU’2004, IEEE 12th Signal Processing and Communications Applications Conference, April 28-30, 2004, p 478-480, Kusadasi Turkey.

[11] Yamagishi, H.; Yamaguchi, J.” Fire flame detection algorithm using a color camera”, Proceedings of 1999 International Symposium on Micromechatronics and Human Science, 1999, pp. 255-260.

[12] Yamagishi, H.; Yamaguchi, J.,”A contour fluctuation data processing method for fire flame detection using a color camera”, 26th Annual Confjerence of the IEEE Industrial Electronics Society, Volume: 12, 2000, pp. 824-829.

[13] 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, Volume: 14 15-8 Oct. 2003, pp. 3775- 3780.