A Method of Video Flame Detection Based on Multi-Feature Fusion

Shidong Wang, Juan Chen

Abstract

A method of video flame detection based on multi-feature fusion is presented in this paper. Physical characteristics of flame, including color clues, flame movement and flame flicker are incorporated into the scheme to detect fires in color video frames. Firstly, mean filtering was used to smooth the video frames and a flame color filtering algorithm was adopted to extract flame candidate regions from video frames. Secondly, detected flame candidate regions were categorized into dynamic candidate regions and static non-candidate regions by using XOR algorithm. Finally, a flame flicker identification algorithm based on flame brightness variation was used to extract true flames from dynamic flame candidate regions. Experiments show that the proposed method is effective and robust which remains with strong anti-disturbance ability.

Keywords: Flame Detection, XOR, Flicker, Anti-Disturbance

1. Introduction

Fire detection is an important issue as it is closely related to life safety, property and environment protection. Early and reliable detection and warning are crucial for building occupants to evacuate to safety area during fire emergencies [1,2]. In recent years, a new technology of video fire detection has attracted considerable attention with development of video surveillance system [3-7,9-16]. The technology uses CCD cameras to capture image frames of the observed scene which provides abundant and intuitive information. These image frames are processed by computerized algorithms to determine the possible existence of fires within the vision field of the cameras. This technology can be easily incorporated into building video surveillance systems and intelligent building systems [8].

Spectral, spatial and temporal models of fire regions were presented by Liu and Ahuja [3]. The shape of a fire region is represented in terms of the spatial frequency of the region contour using its Fourier coefficients. The temporal changes in these coefficients are seen as the temporal signatures of fire regions. However, the problem is that spatial quantization errors for small regions are likely to introduce considerable noise in the Fourier domain. To avoid this problem, a threshold is introduced to eliminate small and elongated narrow regions in ref. [3]. Consequently, the algorithm is limited to detect developed flames which occupy significant portions of video frames. Dedeoglu et al. [9] and Toreyin et al. [10] employed temporal wavelet analysis to extract the quasi-periodic behavior of flickers and used spatial wavelet analysis to capture spatial color variations of flames. Moreover, they proposed a method using three-state hidden Markov models to model flame flicker in visible video images [11]. The above three methods are effective, but time consuming. Marbach et al. [12] proposed to use a temporal accumulation of time derivative images to select a best candidate fire region. Fire features are extracted from the candidate region, and they are combined to determine the presence of fires. Schultze et al. [13] used not only sonagram and spectrogram to investigate the flickering property, but also a simplified flow analysis to trace the characteristic of upward flame turbulent movement. This method represents a fusion of flickering and flow movement analyses. It delivers interesting and promising results which can be applied with video cameras for video fire detection. Byoung et al. [14] used vision sensor and support vector machines to detect fire. The proposed approach is robust to noise, such as smoke, and subtle differences between consecutive frames. However, the support vector machine algorithm has high time-complexity, and it is time consuming.

A flame detection algorithm based on multi-feature fusion was designed by Zhang et al. [15] to quickly recognize fire flame using the theory of the degree of belief, where they combined static and dynamic flame features. The algorithm has rapid processing rate. However, disturbances such as
fire-like objects and noises may be mistakenly seen as flame in their approach. Chen et al. [16] introduced additional algorithms into the multi-feature fusion technique. The method consists of three main features. Firstly an improved Gaussian mixture model method is applied to image frames to extract moving foreground objects from the static background of detection scenes. Secondly, the detected moving objects are categorized into flame candidate and non-candidate regions by using a flame color filtering algorithm. Finally, a flame flicker identification algorithm based on statistical frequency counting is used to distinguish true flames from fire-like objects in video images. This method is effective. However, the flame color filtering algorithm in the scheme will reject the pixels in the flame region where B values may be greater than R values. This problem limits the method to the effective detection of flames.

In this paper, a method to realize video fire detection is described. This method represents a further development of the multi-feature fusion technique. Experiments show that the proposed method is effective and robust which remains with strong anti-disturbance ability. Full descriptions of the method are presented in Section 2 of this paper. Testing and analysis to a set of fire video clips are presented in Section 3 followed by concluding remarks in Section 4.

2. Method

The proposed method consists of three major steps:

1. Flame color feature discrimination.
2. Flame dynamics feature discrimination and elimination of static flame candidate regions.
3. Flame flickering test and elimination of non-flickering dynamic flame candidate regions.

As flames have distinctive visual features, a flame color filtering algorithm is used to extract flame candidate regions from image frames in the first step. In the second step, detected flame candidate regions are then categorized into dynamic flame candidate and static non-candidate regions by using an XOR algorithm [17]. The third step is achieved by a flame flicker identification algorithm to extract true flames from the dynamic flame candidate regions. Detailed discussions of these steps are presented in the following subsections.

2.1 Flame color feature discrimination

Flames have distinctive visual features. Nearly all diffusion fire flames display bright white color in the core region. The color changes from white to yellow then to red as the temperature decreases from the core region towards the flame edges [3]. The RGB values of these pixels form a flame color pattern. Fig.1 is a two-dimensional representation of the flame color RGB distribution. The abscissa in this figure is the G value of flame pixels and the ordinate are the corresponding R, G and B values of the same pixel [18]. It can be seen from Fig. 1 that the majority of the red dots are above a threshold value of 180. The mapping of G component of flame pixels naturally collapses on to a straight line, while the R and B components are scattered in two regions separated by the G component line. The feature of flame pixels can be set as:

\[ R(x, y) > G(x, y) > B(x, y) \]  \hspace{1cm} (1)

where \( R(x, y), G(x, y), B(x, y) \) denote respectively the R, G, B value of a given pixel at \((x, y)\). This feature was used to filter the flame region in ref. [16]. However, it is found through experiments of different fire scenes that there are often a few scattered flame pixels whose G and B values are greater than R values in flame region. In this paper, mean filtering is used to smooth R, G and B values of flame region pixels for further flame color identification.
2.1.1 Mean filtering

Image smoothing is to reduce the noise without deteriorating the important features in the images [19]. Mean filtering is to replace each pixel value with the mean value of its neighbours, including itself [20]. This has the effect of smoothing pixel values which are unrepresentative of their surroundings. Mean filtering is based on a kernel which represents the shape and size of the neighbourhood when calculating the mean value.

2.1.2. Flame color filtering algorithm

In this paper, a 5×5 matrix \( T(u,v) \) is set as kernel in which each element is 1/25. \( u,v \) represent row and column of kernel respectively. After mean filtering, \( R, G \) and \( B \) value of image pixel can be expressed as:

\[
R_f(x,y) = \sum_{u=-2}^{3} \sum_{v=-2}^{3} R(u+x,v+y) \times T(u,v) \quad (2)
\]

\[
G_f(x,y) = \sum_{u=-2}^{3} \sum_{v=-2}^{3} G(u+x,v+y) \times T(u,v) \quad (3)
\]

\[
B_f(x,y) = \sum_{u=-2}^{3} \sum_{v=-2}^{3} B(u+x,v+y) \times T(u,v) \quad (4)
\]

where \( R(x,y), G(x,y) \) and \( B(x,y) \) represent the \( R, G \) and \( B \) value of image pixel respectively. The following Eq. (5) is derived from Eq. (1).

\[
R_f(x,y) > G_f(x,y) > B_f(x,y) \quad (5)
\]

After mean filtering, a few scattered pixels in flame region which are inconsistent with Eq.(1) accord with Eq.(5); Some common disturbances in video frames such as Gaussian noise, salt and pepper noise, are also removed. Fig. 1 shows there are gaps between \( R \) values and \( G \) values which are same between \( G \) values and \( B \) values for most pixels. For further reducing flame detection range, the following two Eqs are adopted to extract candidate flame regions.

\[
R_f(x,y) - d_1 > G_f(x,y) > B_f(x,y) + d_2 \quad (6)
\]

Figure 1. Two-dimensional RGB color value representation of flame pixels.
where \(d_1\) and \(d_2\) are the gaps. They are set as 8 and 7 respectively after empirical tests with 10 video clips of a variety of fire scenes. A few pixels in flame region whose R values are less than 180 are also smoothed by mean filtering and accord with Eq.(7) after empirical tests. The pixels that satisfy Eq.(6) and Eq.(7) are regarded as flame candidate pixels.

Fig. 2 shows the results of the detection of flame candidate regions using flame color discrimination algorithm. Fig. 2(a) is a scene of a camp fire beside a road with street lights and moving vehicle lights. Fig. 2(c) presents a scene with an outdoor flame and a walking man. Fig. 2(b) and Fig. 2(d) are binary images after the color filtering of the images in Fig. 2(a) and Fig. 2(c). The white regions in the binary images represent the flame candidate regions. These figures show fires are segmented correctly. However, the vehicle headlights and street lights cannot be eliminated in Fig.2(b). Fig. 3 shows the results of detecting flame candidate regions with noise. Fig. 3(a) is a frame with Gaussian noise; Fig. 3(c) is a frame with salt and pepper noise. Fig.3(b) and (d) show the noise does not influence the segment of flame candidate regions. The shape of flame candidate regions in Fig. 3(b) and Fig. 3(d) are almost consistent with those in Fig. 2(b). Chen et al. [16] used Eq. (1) to get the flame candidate region. This result of flame detection not only omits a few pixels in flame region which are inconsistent with Eq.(1) but also is easily disturbed by noise. Comparing to Chen’s method, the two questions are both solved in this paper.

2.2 Flame dynamics feature discrimination

In video frames of a fixed surveillance camera, majority of objects are static while natural fire flames are seen as dynamic regions. This feature can thus be utilized to further distinguish a fire flame from flame candidate regions. In this paper, XOR operator is used to detect moving objects.
This method is based on the application of the XOR operator to the binary images obtained by the color filtering described in subsection 2.1. Considering that flame is a dynamic region, its size and shape always change with the passage of time. As a result, most flames do not completely overlap in two consecutive binary flame images. To quantify this specific characteristic, the XOR algorithm is employed to eliminate the static flame candidate regions.

The XOR operation produces a binary output of 1 if two of its binary inputs are different, and it produces a binary output of 0 if both of its binary inputs are same (either 0 or 1) [17]. Thus, XOR operation can be used as motion detector. The non-overlap regions (the regions of XOR output of 1) are obtained using the XOR operation between two binary images obtained by the color filtering. The existence of non-overlap region testifies the dynamics feature of the flame candidate regions, and the static candidate regions are eliminated from the binary image. To achieve a reasonably judgment of dynamic regions with different sizes, the following equation should be fulfilled:

\[ Ra = \frac{\text{Count}_{\{\text{Reg}_i \cap \text{Reg}_{i+1}\}}}{\text{Count}_{\{\text{Reg}_i \cup \text{Reg}_{i+1}\}}} \quad (8) \]

where \( \text{Reg}_i \) and \( \text{Reg}_{i+1} \) represent the corresponding flame candidate regions in two consecutive binary images obtained by the color filtering. \( \text{Count}_{\{\text{Reg}_i \cap \text{Reg}_{i+1}\}} \) represents the pixels number that XOR outputs are 1 between corresponding pixels covered by \( \text{Reg}_i \) and \( \text{Reg}_{i+1} \); \( \text{Count}_{\{\text{Reg}_i \cup \text{Reg}_{i+1}\}} \) represents the pixels number covered by \( \text{Reg}_i \) and \( \text{Reg}_{i+1} \). \( Ra \) reflects the ratio of non-overlap region to the region covered by \( \text{Reg}_i \) and \( \text{Reg}_{i+1} \). To eliminate the effect of noise, \( Ra \) is set as the following equation after empirical tests.

\[ 0.7 > Ra > 0.15 \quad (9) \]

The corresponding binary region satisfy Eq.(9) is regarded as a dynamic region; otherwise, the region is regarded as a static region and is eliminated from binary images.

Fig. 4 shows the motion detection results after the color filtering. Fig. 4(a) and (b) are two consecutive frames, and so are Fig. 4(e) and (f). Binary images are obtained from Fig. 4(a), (b), (e) and (f) by the color filtering. The non-overlap regions of the flame candidate regions which are obtained using XOR operation between two consecutive binary images are shown in Fig. 4(c) and (g). The white regions in Fig. 4(c) and (g) testify the corresponding regions are dynamic regions which are kept. Other static candidate regions such as street lights in Fig. 2(b) are eliminated. Fig. 4(d) and (h) shows the results after the color and motion filtering. However, vehicle headlights can not be excluded in Fig. 4(d).
2.3 Flame flicker feature extraction

Fire has a unique characteristic of flickering. Yang and Wang [21] indicated that the contours and chrominance or brightness of flame generally oscillates with a frequency range of 0.5–20 Hz. Fourier analysis may be conducted to identify the characteristic frequencies of flickering. However, transforming signals from the time domain to the frequency domain is time-consuming and will affect the efficiency of the detection algorithm. Chen et al. [16] developed a flickering detection algorithm that demonstrates a good effect. If the brightness value of any pixel changes between two consecutive image frames in this algorithm, counter is added by 1. If any pixel counter exceeds a threshold, the pixel is regarded as a part of the flickering flame. However, the algorithm must calculate all pixels counter value and is time consuming. A different approach is employed to resolve this problem in the current study. With the passage of time, the flame pixel brightness values, especially the flame edge pixels, changes significantly. Meanwhile, the pixels distribution of flame brightness level also varies significantly. This feature can be used to test flame flickering. The approach is described as follows.

The image brightness is divided into 10 levels in this paper. If the distribution of flame brightness level changes between two corresponding flame candidate regions in two consecutive frames, the flame...
candidate region is regarded as the flame region. The above scheme is formulated in Eq. (10) and Eq. (11).

\[
T = \sum_{d=1}^{10} \left( \frac{\Delta B_d(i)}{B_d(i)} \right) \sum_{d=1}^{10} \left| B_d(i) \right|
\]

\[
T > T_1
\]

where \(T\) represents the brightness changing extent of flame candidate regions detected by the color and motion filtering described in subsection 2.1 and 2.2. To eliminate the effect of noise, a threshold \(T_1\) is introduced. It is set as 20% after empirical tests. \(\Delta B_d(i)\) is defined as:

\[
\Delta B_d(i) = \left| B_d(i) - B_d(i+1) \right|
\]

where \(B_d(i)\) register the number of pixels at brightness level \(d\) of the \(i\) frame. The flame candidate region is regarded as the flame region if \(T\) satisfy Eq.(11); otherwise, the corresponding region is eliminated from binary images obtained by the color and motion filtering.

Fig. 5 depicts flicker filtering results of Fig. 2(a) after the color and motion filtering. It shows that vehicle headlight in Fig. 4(d) is eliminated. After the color, motion filtering and flicker filtering, the flame detection result of Fig. 2(c) is shown as Fig. 2(d). As can be seen in Fig. 5 and Fig. 2(d), the fire regions have been correctly recognized.

3. Testing and analysis

The proposed algorithm which was implemented using Matlab and Visual C++ has been tested on a PC with a Pentium 2.50 GHz processor and 1 GB memory. The algorithm was applied to 10 video clips of a variety of fire scenes including different environmental background and illumination conditions. The video clips were collected from State Key Laboratory of Fire Science, University of Science and Technology of China. Some of them are downloaded from the Internet (http://signal.ee.bilkent.edu.tr/VisiFire/Demo/). The resolution of the video images is 320×240. The processing rate of the proposed method achieved 24 fps in the sample applications. Flame regions are all detected. Fig. 6 shows examples of different testing and the corresponding detection results of these scenes. Fig. 6(a) depicts a fire in a corridor with a strong floor reflection. The fire in Fig. 6(c) is located besides floor with some shaking leaves. Fig. 6(e) demonstrates a scattered fire. Fig. 6(b), (d) and (f) show the fire detection result.

As can be seen from Fig. (6), whether it is big fire or small fire, even with a shadow in Fig. 6(a), the proposed method can effectively successfully detect fire flame. Experiments results indicate that the method has a strong anti-disturbance ability and a strong robustness under different scenes and illumination conditions. The algorithm has time-complexity of \(O(n)\) where \(n\) is image resolution.
4. Concluding remarks

In this study, a method of video flame detection based on multi-feature fusion has been developed. Numerical filters were constructed to identify fire flames based on their color, motion and flickering features. Comparing to previous methods reported in the literature, the current method is more accurate to extract candidate regions and detect the existence of flames in video images.

The experimental results, which were obtained under a variety of background environments, demonstrate that the multi-feature fusion video flame detection algorithm is effective and robust which remains with strong anti-disturbance ability. It should be pointed out that the testing of the algorithm was conducted on video clips obtained from independent sources, which were taken under limited environmental conditions. Further laboratory realtime experiments with wider a range of environment conditions are required to achieve more quantifiable performance assessment. Detection method based on smoke and flame features will be further explored in our future work.

![Testing images of three different scenes](image)

**Figure 6.** Testing images of three different scenes

5. References