Real-time multi-feature based fire flame detection in video

Rui Chi[1], Zhe-Ming Lu[2*] and Qing-Ge Ji[1*]

[1]School of Data and Computer Science, Sun Yat-sen University,

Guangzhou 510006, P. R. China

[2]School of Aeronautics and Astronautics, Zhejiang University,

Hangzhou 310027, P. R. China

[*]Corresponding Authors. Tel. +86-451-87953349

Email: Zhe-Ming Lu, zheminglu@zju.edu.cn

Qing-Ge Ji, issjqg@mail.sysu.edu.cn

Abstract. In this paper, we present a new approach to detect fire flame by processing and analyzing the stationary camera videos. For a fire detection system, it is desired to be sensitive and reliable. The proposed method improves not only the sensitivity but also the reliability through reducing the susceptibility to false alarms. The proposed approach based on multi-feature, i.e., chromatic features, dynamic features, texture features, and contour features, can both improve the sensitivity and reliability in fire detection. In our approach, we adopt a novel algorithm to extract the moving region and analyze the frequency of flickers. Experimental results show that the proposed method can run in real-time and performs favorably against the state-of-art methods with higher

accuracy in fire videos, lower false alarm rates in non-fire videos and faster response time.

Keywords: fire detection, video analysis, multi-feature, sensitivity, reliability

1. Introduction

Nowadays, more and more fire detection systems can be seen in airports, supermarkets, hospitals, schools, etc. However, most of these systems are based on traditional sensors. Although traditional sensors have been widely used, but they still have many disadvantages in practice. Traditional sensors cannot be used in outdoor or large open areas and cannot locate the fire regions. Besides, conventional fire detection systems need to take a long time for particles to reach the “point” detector.

Video-based fire detection systems can be an effective method to prevent fires in large spaces and areas where conventional fire detectors cannot be used. Fire detection methods based on video are also possible to alarm immediately. Video-based fire detection techniques are viable alternatives or complements to the existing fire detection techniques and have been shown to be useful to solve several problems related to traditional sensors [1].

In literatures, several video-based fire flame detection algorithms have been proposed. Generally, most of these methods are mainly based on color detection [2-16], dynamic feature analysis [2-4,7-13,15,16], etc. Chen et al. analyzed the dynamics of fire growth and disorder combined with a RGB model to detect fires [2]. In References [3] and [4], flame boundary was represented in the wavelet domain and the high frequency nature of the boundaries of fire regions was used as a clue to model the flame flicker spatially. Celik et al. used a rule-based generic color model for flame pixel classification [6]. This algorithm adopts the YCbCr color space to separate the luminance from the chrominance more effectively than other color spaces such as RGB. In Reference [7], fire detection based on the luminance map and support vector machine was proposed. Borges et al. proposed a probabilistic model for color-based fire detection [8]. In addition, this method analyzes several features, such as colors, area sizes, surface coarseness, boundary roughness, and skewness within the estimated fire regions. In Reference [9], fire detection algorithm based on CIE L*a*b* color space was proposed. Habiboglu et al. proposed a method to detect fires using a spatio-temporal covariance matrix of video data [10]. In Reference [11], the authors extracted special parameters based on tempo-spatial characteristics and used the support vector machine to distinguish between fire areas and non-fire areas. In Reference [12], the combination growth rate and Lucas-Kanade optical flow were used for dynamic analysis, which could easily distinguish between fire and the disturbances. In Reference [13], two optical flow methods were specifically designed for describing the characteristic features of fire motion. Zhang et al. proposed an improved probabilistic approach [14] using two improved features, i.e., color and motion. In Reference [15], the authors proposed a fire detection method using an ensemble of experts based on the information about color, shape, and flame movements. In addition, Qureshi et al. proposed a real time early fire detection system [16] which used parallel image processing streams to detect flame and smoke.

Although many algorithms have been done to detect fires, however, few methods are both sensitive and reliable. To overcome this problem and enhance the fire detection performance, this paper proposes an effective multi-feature based approach for short range (<50m) fire flame detection, considering multiple features including chromatic features, dynamic features, texture features and contour features of the flames. Experimental results indicate that the performance of the proposed approach exceeds other fire detection algorithms, with higher accuracy in fire videos, higher reliability in non-fire videos and faster response time.

The contributions of this paper are: First, a novel method to extract the moving region is proposed, which can also be used to calculate the frequency of flickers simply and effectively. Second, we use a new way to analyze the fire contour based on fractal dimension. Third, this paper presents an algorithm, which combines multi-feature, to detect fire flame with high sensitivity, high reliability and fast response time.

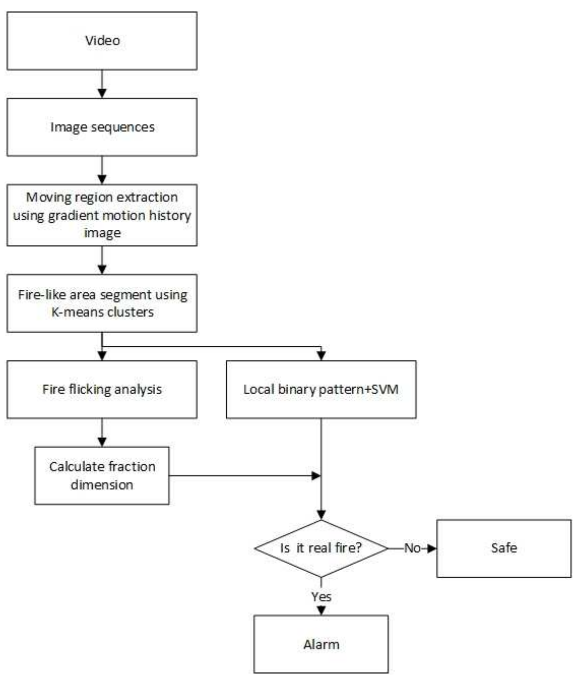

Fig. 1 The flow chart of the proposed algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited.
Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

2. The proposed fire detection algorithm

The proposed algorithm consists of six stages, as shown in Fig.1. First, we extract the moving region based on the gradient motion history image (GMHI). Then we check the moving pixels whether they match to the fire colors that we have got using the K-means algorithm in advance. Next, we select candidate pixels whose flickering frequencies meet fire flickering frequency by the motion history image information. Fourthly, we extract the LBP feature from the candidate pixels, and put the feature into SVM classification. Fifthly, we analyze the contour of each region near the candidate pixels based on the fractal dimension. Finally, we judge that there is a fire if both results of fractal dimension and SVM classification are positive.

2.1 Moving region extraction

In Reference [17], Bradski et al. presented a fast and simple method using a timed motion history image (tMHI) for representing motion from the silhouettes. Silhouettes could be generated from frame differencing, background subtraction, etc. However, silhouettes generated by these methods cannot represent contour changes directly and timely. So we use the binary gradient image as the history template, which is quite sensitive to contour changes and necessary for flicker analysis later.

Moving regions in the video are determined by using the gradient motion history image (GMHI). Let $T(x,y)$ represent the binary gradient image, it can be calculated as follows:

$$T(x,y)=\begin{cases}255 & T_x(x,y)=255 \quad or \quad T_y(x,y)=255 \\ 0 & otherwise\end{cases} \tag{1}$$

where $T_x(x,y)$ and $T_y(x,y)$ are the binary gradient images along the x-axis and y-axis respectively, and gradient images are computed using grayscale images. Let $tMHI(x,y)$ represent the gradient moving image, which records the pixel's moving history information, it can be calculated as follows:

$$tMHI(x,y)=\begin{cases}\tau & T(x,y)=255 \\ 0 & tMHI(x,y)<(\tau-\delta)\end{cases} \quad (2)$$

where τ is the current time, and δ is the timestamp defined in advance. Normally, we use frame number to indicate time. So we can guarantee that we only record the latest δ moving history information. Let $EM(x,y)$ represent the effective motion image, it can be calculated as follows:

$$EM(x,y)=\begin{cases}255 & min \le MAX(x,y)-MIN(x,y) \le max \\ 0 & otherwise\end{cases} \quad (3)$$

where $MAX(x,y)$ and $MIN(x,y)$ are the maximum and minimum $tMHI$ values of the 5×5-neighborhood of point (x,y) in the gradient moving image, and min and max are given constants. When the difference between $MAX(x,y)$ and $MIN(x,y)$ is greater than max, that means point (x,y) is stationary pixel or on the boundary of the background. On the contrary, the smaller difference between $MAX(x,y)$ and $MIN(x,y)$ than min indicates that point (x,y) is stationary pixel or on the inner edge in the background. Thus, the effective motion image records the moving regions.

2.2 Detection of fire-like pixels

As we all know, there may be other moving objects in the video, such as people, cars and so on. However, the colors of these moving objects generally will be different from the color of fire.

In this part, we use the K-means algorithm to group N known fire pixels (pixels from known fire regions) into k clusters according to RGB color feature during the training phase, which will be used to determine whether a moving pixel in the RGB color space belongs to a fire-like pixel or not. So, the basic idea is composed of three steps: (1) collect a RGB dataset of known fire pixels; (2) employ the K-means algorithm; (3) select the moving pixels having fire-like colors. These steps of this section can be described as follows:

Step1: For every fire image, we select n random pixels from the fire region in every fire picture, and get their RGB values. After the data collection phase, we obtain a dataset with 700 samples. Obviously, the training data determines which flame can be detected, such as yellow flame or blue flame.

Step2: In this phase, the K-means algorithm is used to group n known fire pixels selected in Step1 into k clusters. The number of clusters k is related to the training fire datasets. In practice, the number of clusters k *may be different according to the different training datasets.* Since we have got k clusters, we can then calculate the distance of each pixel to the center $k[i]$ it belongs to. And we can get the average distance ave_dis of all pixels to their centers.

Step3: Calculate the distances $d_i(x,y)$ of every moving pixel (x,y) to the centers of clusters. Check the condition:

$$\frac{d_i(x,y)}{ave\_dis} < \alpha, i = 0,1,...,k-1 \tag{4}$$

If the above condition is satisfied, which means the moving pixel (x,y) and the known fire pixels in the cluster i share some similarity in RGB space, then we define the pixel as a fire-like pixel, α is the correlation coefficient.

Next, we are going to analyze the fire-like pixels to determine whether they are candidate pixels or not.

2.3 Fire flicking analysis

Lots of researchers have pointed out that the frequency of fire flickers is about 10Hz [18][3]. We have recorded gradient moving images in δ time intervals, and we usually define $\delta = 15$ (the unit of δ is frame interval), so we can easily get the moving information $move\_count(x, y)$ of all pixels in δ time intervals according the GMHI. $move\_count(x, y)$ is the number of times of movement of the pixel (x, y) in δ time intervals. If a pixel's $tMHI$ is updated (not becomes to 0) and the RGB color of this pixel satisfies Eq. (4) in δ time intervals, then $move\_count(x, y)$ increases by 1. $move\_count(x, y)$ updates in current timestamp (δ time intervals) and will become to 0 at the beginning of next timestamp. Now we can evaluate the frequency of fire-like pixels, since we have got the moving information in a period of time.

However, different surveillance systems have different video capture rates. To capture 10Hz fire flickers, the device has to capture the video clip at the rate of more than 20 frames per second (fps). If the capture rate is below 20 frames per second, the frequency of flickers in the video is also below 10Hz obviously. So we set a threshold T, if a fire-like pixel (x, y) moved more than T times in δ time intervals, we firmly believe that this fire-like pixel is a candidate pixel. For example, when the video capture rate is 15 fps, we take $\delta = 15$ and $T = 5$.

2.4 Local binary pattern of candidate pixels

The original LBP operator, introduced by Ojala, is a very useful method of texture description. The operator labels the pixels of an image by thresholding the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited.
Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

3×3 -neighborhood of each pixel with the center value and considering the result as a binary number. Then the normalized histogram of the labels can be used as a texture descriptor. An illustration of the original LBP operator is shown in Fig.2.

In our work, we extract the LBP features from known fire regions and trained these features in SVM in advance. In the method of fire detection, we extract the LBP feature from the moving fire-like area in each frame. Then we put the LBP feature to SVM classification to distinguish.

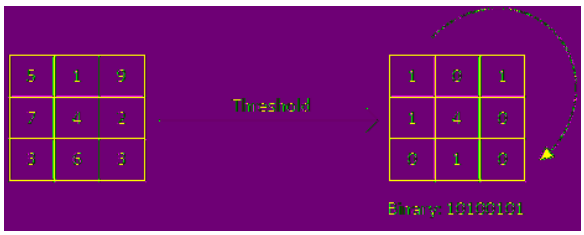

Fig. 2 *The original LBP operator*

2.5 Fractal dimension

A fractal is an image that can be completely described by a mathematical algorithm in its infinitely fine texture and detail. The "fractal property" is defined by the repetition of parts of a pattern within the pattern itself, and is called "self-similarity". Fractal dimension used in image processing is based on two points. One of the unique things of fractals is that they have non-integer dimensions. Secondly, based on the hypothesis of Pentland, fractals in nature correspond to gray description of images at a certain extent. There are many methods to compute the fractal dimension such as box counting dimension [19], capacity dimension, correlation dimension and Hausdorff dimension. In our work, we adopt the box counting dimension for reducing computation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited.
Content may change prior to final publication in an issue of the journal. To cite the paper please use the doi provided on the Digital Library page.

Mandelbrot stated that one criterion of a surface being fractal is its similarity. Self-similarity can be explained as follows. Consider a bounded set A to be in the n-dimensional Euclidean space. The set is regarded to be self-similar when A is the union of N_r distinct (non-overlapping) copies of itself each of which is similar to A scaled down by a ratio r. The fractal dimension D of A can be derived from the following relationship.

$$1 = N_r r^D \tag{5}$$

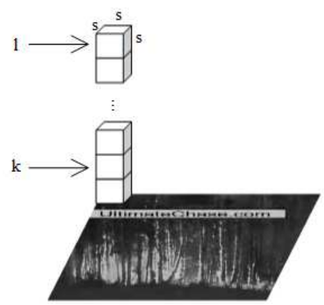

Fig. 3 *Fractal dimension calculation*

However, in practice, natural scenes do not exhibit deterministic self-similarity. Instead, they exhibit some statistical self-similarity. Consider that the image of size $M \times M$ has been scaled down to a size of $s \times s$ where $1 < s \le M/2$ and s is an integer. Then we have an estimation of $r = s/M$. Now, we consider the image to be in a 3D space with (x, y) denoting the 2D position and the third coordinate (z) denoting the gray level. The (x, y) space is partitioned into grids of size $s \times s$. Let the minimum and maximum gray levels of the image in the $(i, j)^{th}$ grid fall in box numbers k and l, respectively. The fractal dimension can be calculated as shown in Fig.3. Thus we have

$$n_r = l - k + 1 \tag{6}$$

where n_r is the contribution in the $(i,j)^{th}$ grid. Taking contributions from all grids, we have

$$N_r = \sum_{i,j} n_r(i,j) \tag{7}$$

N_r is counted for different values of r (i.e. different values of s). Then we can estimate D, the fractal dimension, from least square linear fit of $\log(N_r)$ against $\log(1/r)$.

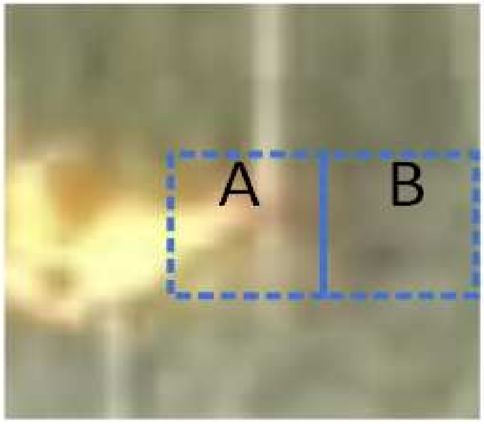

Fig. 4 *Fractal dimension calculation in our method*

In our method, we only want to analyze the contour information of the region near candidate pixels. So we select the $M \times M$ region around each candidate pixel. Besides, to avoid the influences from the background, we only calculate n_r of boxes that contain boundaries of the fire-like area and background, and for the rest boxes we assume that the pixels in a box have the same gray value, which means $n_r = 1$. In Fig.4 we show an example to calculate n_r in Grid A, and let $n_r = 1$ in Grid B. In this way, we can get the contour information of the region near each candidate pixel. Since a flame has a complicated outline, and the obtained value of fraction dimension can indicate the complexity of the outline, so we can exclude false pixels from candidate pixels. Besides, to avoid the influences of outliers, we also demand the number of fired pixels must exceed a certain value.

3. Experimental results

3.1 Experimental evaluation

In order to evaluate the performance of our method, we apply our method to a set of video clips from variety of scenes, including different environmental backgrounds and illumination conditions, and show them in Fig.5. Test videos were downloaded from http://signal.ee.bilkent.edu.tr/VisiFire/, NIST and [16]. As shown in Table 1, we collect a total of 20 video clips, 15 of which contain flame and 5 of which are distracters containing no fire.

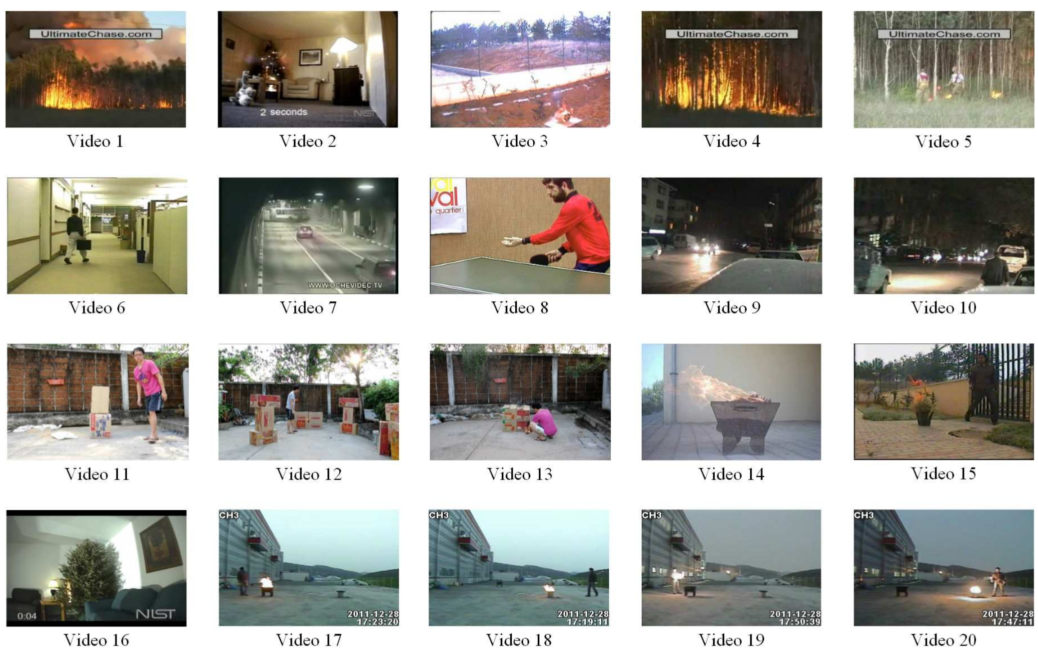

Fig. 5 *Test video clips*

Through the analysis of several video clips, we empirically set values of the parameters for the proposed approach as shown in Table 2. Besides, the number of clusters in K-means is 5, and centers of these clusters in RGB color space are (238.588, 121.327, 21.455), (250.176, 230.536, 74.5229), (187.874, 57.5747, 12.8276), (251.793, 247.424, 194.391) and (243.766, 160.475, 81.8038). The average distance is 34.2476.

Table 1

Test video clips in our experiments

	Total Frames	Resolution	Fire or Not	Description
Video 1	190	400x256	Yes	Fire in the forest
Video 2	75	352x240	Yes	Christmas tree room 1
Video 3	400	320x240	Yes	Fire in the backyard
Video 4	208	400x256	Yes	Fire in the forest
Video 5	260	400x256	Yes	Two man firing grass and trees
Video 6	300	352x288	No	A man walking through a hall
Video 7	294	320x240	No	Tunnel accident
Video 8	112	352x240	No	A man with red shirt playing ping-pong
Video 9	155	320x240	No	Car lights in the night 1
Video 10	160	320x240	No	Car lights in the night 2
Video 11	2208	640x424	Yes	Outdoor plain box
Video 12	4301	640x424	Yes	Multiple boxes-1
Video 13	4500	640x424	Yes	Multiple boxes-2
Video 14	439	320x240	Yes	Barbeque stand
Video 15	707	320x240	Yes	Plant pot
Video 16	1464	720x480	Yes	Christmas tree room 2
Video 17	3490	320x240	Yes	Outdoor daytime 10 m gasoline
Video 18	4547	320x240	Yes	Outdoor daytime 10 m heptane
Video 19	1207	320x240	Yes	Outdoor night 10 m gasoline
Video 20	3274	320x240	Yes	Outdoor night 10 m heptane

Table 2

Values of the parameters for the proposed approach when video capture rate is 15 fps

Parameter	Value
δ	15
min	0
max	15
α	1.8
T	5

The proposed algorithm runs at 52 frames per second with a C++ implementation on an i3 Quad-Core machine with 3.40 GHz CPU and 8 GB RAM.

The performance of the proposed fire detection algorithm is compared with other six algorithms, i.e., fire detection in video sequences using a generic color model (Algorithm 1)[6], fire detection based on vision sensors and support vector machines(Algorithm 2)[7], the computer vision-based method for real-time fire and flame detection(Algorithm 3)[3], the probabilistic approach for vision-based fire detection in videos(Algorithm 4)[8], fire flame detection in video sequences using multi-stage pattern recognition techniques(Algorithm 5)[11], and QuickBlaze: early fire detection using combined video processing approach(Algorithm 6)[16] . The quantitative results of these six methods are obtained from [11][16].

The comparison results about fire detection accuracies among algorithm1-5 on fire video clips are presented in Table 3. For Video clip 1, though fire in this video is clearly visible, however the total frames are not many and the fire behavior is stable, since the proposed algorithm can detect $N-T+1$ fired frames at most (N is number of frames, T is threshold in flicking analysis), so the accuracy is low. For the Video clip 2, the fire in the Christmas room breaks out rapidly, all the algorithms perform well. For Video clip 3, because the proposed algorithm is quite sensitive to the moving region, our algorithm outperforms other algorithms obviously. For Video clip 4, the fire is clearly visible and the moving fire region is legible, so our algorithm has the highest accuracy in this video. For Video clip 5, the fire region is quite small, however the proposed algorithm still has the best performance among these algorithms. In a word, for fire video clips, the proposed algorithm has higher accuracies than the other algorithms.

Table 3

Fire detection results of the proposed algorithm and conventional algorithms on fire video clips

	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5	Proposed
	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)
Video 1	94.74	95.26	94.74	95.26	95.79	**96.50**
Video 2	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**	**96.00**
Video3	92.25	92.50	92.75	92.25	93.00	**98.00**
Video 4	94.50	94.50	94.00	94.50	95.00	**98.08**
Video 5	94.59	94.21	94.59	94.59	94.98	**96.54**
Average	94.42	94.49	94.42	94.52	94.95	**97.02**

Table 4 shows the error rates of fire detection in 3 video clips without fire compared with algorithm1-5. For Video clip 6, the main moving region does not include fire-like colors, though the light is slightly flickering, all algorithms do not detect the fire. For Video clip 7, the moving region contains a car with colors similar to fire, however the moving region does not satisfy the fire flicker, thus, our algorithm does not detect the fire. For Video clip 8, the clothes are red, but the moving region is different from the fire. As shown clearly, the average error rate of our algorithm is 0.00% versus average error rates of 3.12%, 2.97%, 2.91%, 2.47%, and 1.76% for other algorithms.

Table 4

False fire detection rates of the proposed algorithm and conventional algorithms on non-fire video clips

	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4	Algorithm 5	Proposed
	Error Rate (%)	Error Rate (%)	Error Rate (%)	Error Rate (%)	Error Rate (%)	Error Rate (%)
Video 6	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Video 7	4.90	4.55	4.55	3.85	3.50	**0.00**
Video 8	4.46	4.46	3.57	3.57	1.79	**0.00**
Average	3.12	2.97	2.91	2.47	1.76	**0.00**

The comparison of proposed approach and Algorithm 6 for response time, first fire frame, is shown in Table 5. In general, we observe that on 8 of 10 test video clips that containing fire, fire detection using the proposed approach results in better response time than using Algorithm 6. Especially, when the fire is small and not obvious, the proposed approach responses much faster than Algorithm 6, such as Video clip 12, Video clip 13, Video clip 15, Video clip 16, and Video clip 20. Because the proposed approach is sensitive to the contour change extremely.

Table 5

Experimental results for video clips containing fire

	First Fire Frame (Ground truth)	First Fire Frame (Algorithm 6)	First Fire Frame (Proposed)
Video 11	1037	**1158**	1164
Video 12	20	694	**24**
Video 13	258	440	**265**
Video 14	1	**11**	**11**
Video 15	1	323	**8**
Video 16	161	305	**194**
Video 17	90	105	**101**
Video 18	90	207	**120**
Video 19	60	104	**67**
Video 20	1	168	**9**

Overall, the proposed algorithm clearly outperforms other algorithms in terms of the accuracy of fire detection, the error rate of false fire detection and response time.

3.2 Different methods to extract moving regions

We also compared detection results of using different methods to extract moving regions. In our experiment, we compared GMHI with Gaussian mixture model (GMM) on fire video and non-fire video clips. The results are shown in Table 6 and Table 7.

Table 6

Fire detection results of using different methods to extract moving regions on fire video clips

	Proposed(GMHI+LBP+FD)	Method using GMM+LBP+FD
	Accuracy (%)	Accuracy (%)
Video 1	**96.50**	59.50
Video 2	**96.00**	91.50
Video 3	**98.00**	91.75
Video 4	**98.08**	**98.08**
Video 5	**96.54**	90.00
Average	**97.02**	86.17

Table 7

Fire detection results of using different methods to extract moving regions on non-fire video clips

	Proposed(GMHI+LBP+FD)	Method using GMM+LBP+FD
	Error Rate (%)	Error Rate (%)
Video 6	**0.00**	**0.00**
Video 7	**0.00**	**0.00**
Video 8	**0.00**	5.36
Video 9	**0.00**	**0.00**
Video 10	**0.00**	0.63
Average	**0.00**	2.40

For Video clip 1, the accuracy of the compared method is quite low, the reason is that in Video clip 1 the fire behavior is stable, and the flicker is not obvious. After a period of adapting, GMM was not sensitive to the fire flicker, which means the method could not extract the valid moving fire region. The fire detection results of Video clip 1 are shown in Fig. 6, the detected flickering points were marked with red squares. Under interference of moving smoke, the compared method cannot find the accurate fire region, while the proposed method can still locate the fire flickering points accurately.

Overall, moving region extraction using GMHI outperforms using GMM to detect fire in our method.

(a)

(b)

Fig. 6 *Comparison between fire detection algorithms using different moving region extraction methods:(a) Moving region extraction using GMHI;(b) Moving region extraction using GMM.*

3.3 Comparison among GMHI, LBP, GMHI+LBP and the proposed method

Besides, we also compared the fire detection results of using GMHI, LBP+SVM, GMHI+LBP and the proposed method. In the proposed method, to avoid the final results from being determined by the worst performance method and improve the sensitivity in fire scenes, if there are more than 30 frames which are detected as fire in the past 40 frames, we assume that the detection result of LBP is positive in the current frame even if the result of SVM classification is negative.

The results are shown in Table 8 and Table 9. For the fire detection method using GMHI, it has the best performance in fire video clips, however it has the worse performance in non-fire video clips. For the fire detection method using LBP, it recognizes a non-fire as a fire badly in every negative fire video clip. For the fire detection method using GMHI and LBP, it has better performance in non-fire video clips, but it still may be affected by some flickering light. For the proposed method, generally speaking, though the average accuracy of the proposed method lowers slightly, the error rates are lowered to zero meanwhile.

Table 8

Fire detection results of GMHI, LBP+SVM, GMHI+LBP and the proposed method on fire video clips

	GMHI	LBP+SVM	GMHI+LBP	Proposed
	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)
Video 1	97.50	95.00	97.50	96.50
Video 2	97.50	91.50	93.50	96.00
Video 3	99.00	96.00	99.00	98.00
Video 4	98.08	99.04	98.08	98.08
Video 5	96.92	99.23	96.92	96.54
Average	97.80	96.15	97.00	97.02

Table 9

Fire detection results of GMHI, LBP+SVM, GMHI+LBP and the proposed method on non-fire video clips

	GMHI	LBP+SVM	GMHI+LBP	Proposed
	Error Rate (%)	Error Rate (%)	Error Rate (%)	Error Rate (%)
Video 6	1.33	44.00	0.33	0.00
Video 7	0.00	8.00	0.00	0.00
Video 8	3.57	19.64	0.00	0.00
Video 9	0.00	7.74	0.00	0.00
Video 10	1.88	45.00	1.25	0.00
Average	1.36	24.88	0.32	0.00

4. Conclusion

This paper presents a method based multi-feature for fire detection in color video clips. Our method employs information obtained from chromatic features, dynamic features, texture features and contour features of flame. The algorithm uses gradient motion history images to extract moving regions and the fractal dimension to analyze contour information. The experimental results show that the proposed approach can be used for fire detection in video and fixed surveillance, with high sensitivity, high reliability and fast response time.

5. Acknowledgement

This research was supported by National Natural Science Foundation of China (No.60473109), NSFC-Guangdong Union Foundation of China (No.U0735001), and Natural Science Foundation of Guangdong Province, China (No.2016A030313288). We would like to thank the anonymous reviewers for their suggestions and comments.

References

[1] Çetin, A.E., Dimitropoulos, K., Gouverneur, B., et al.: 'Video fire detection–Review', Digital Signal Processing, 2013, 23, (6), pp.1827-1843.

[2] Chen, T.H., Wu, P.H., Chiou, Y.C.: 'An early fire-detection method based on image processing'. Proc. IEEE Int. Conf. Image Processing, ICIP 2004, 24-27 October 2004, vol.3, pp. 1707-1710.

[3] Töreyin, B.U., Dedeoğlu, Y., Güdükbay, U., et al.: 'Computer vision based method for real-time fire and flame detection', Pattern recognition letters, 2006, 27, (1), pp. 49-58.

[4] Toreyin, B.U., Cetin, A.E.: 'Online detection of fire in video'. Poc. IEEE Int. Conf. Computer Vision and Pattern Recognition, CVPR 2007, Minneapolis, MN, 17-22 June 2007, pp. 1-5.

[5] Yu, F.X., Su, J.Y., Lu, Z.M., et al.: 'Multi-feature based fire detection in video', International Journal of Innovative Computing, Information and Control, 2008, 4, (8), pp. 1987-1993.

[6] Celik, T., Demirel, H.: 'Fire detection in video sequences using a generic color model', Fire Safety Journal, 2009, 44, (2), pp. 147-158.

[7] Ko, B.C., Cheong K.H., Nam, J.Y.: 'Fire detection based on vision sensor and support vector machines', Fire Safety Journal, 2009, 44, (3), pp. 322-329.

[8] Borges, P.V.K., Izquierdo, E.: 'A probabilistic approach for vision-based fire detection in videos', IEEE Trans. Circuits and Systems for Video Technology, 2010, 20, (5), pp. 721-731.

[9] Celik, T.: 'Fast and efficient method for fire detection using image processing', ETRI journal, 2010, 32, (6), pp. 881-890.

[10] Habiboglu, Y., Gunay, O., Cetin, A.E.: 'Flame detection method in video using covariance descriptors', Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Prague, 22-27 May 2011, pp. 1817-1820.

[11] Xuan Truong, T., Kim, J.M.: 'Fire flame detection in video sequences using multi-stage pattern recognition techniques', Engineering Applications of Artificial Intelligence, 2012, 25, (7), pp. 1365-1372.

[12] Rinsurongkawong, S., Ekpanyapong, M., Dailey, M.N.: 'Fire detection for early fire alarm based on optical flow video processing', Proc. IEEE Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, May 2012, pp. 1-4.

[13] Mueller, M., Karasev, P., Kolesov, I., et al.: 'Optical Flow Estimation for Flame Detection in Videos', IEEE Trans. Image Processing, 2013, 22, (7), pp. 2786-2797.

[14] Zhang, Z., Shen, T., Zou, J.: 'An Improved Probabilistic Approach for Fire Detection in Videos', Fire Technology, 2014, 50, (3), pp. 745-752.

[15] Foggia, P., Saggese, A., Vento, M.: 'Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion', IEEE Trans. Circuits and Systems for Video Technology, 2015, 25, (9), pp. 1545-1556.

[16] Qureshi, W.S., Ekpanyapong, M., Dailey, M.N., et al.: 'QuickBlaze: Early Fire Detection Using a Combined Video Processing Approach', Fire Technology, 2015, pp. 1-25.

[17] Bradski, G.R., Davis, J.W.: 'Motion segmentation and pose recognition with motion history gradients', Machine Vision and Applications, 2002, 13, (3), pp. 174-184.

[18] Albers, B.W., Agrawal, A.K.: 'Schlieren analysis of an oscillating gas-jet diffusion flame', Combustion and flame, 1999, 119, (1), pp. 84-94.

[19] Falconer, K.J.: 'Fractal geometry: mathematical foundations and applications'(John Wiley Press, Chichester, 1990)