Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 29 (2008) 925–932
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A fast accumulative motion orientation model based on integral image for video smoke detection

Feiniu Yuan *

School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, Jiangxi, China
State Key Lab of Fire Science, University of Science and Technology of China, Hefei 230027, Anhui, China

Received 1 February 2007; received in revised form 29 November 2007
Available online 29 January 2008

Communicated by S. Dickinson

Abstract

Video smoke detection has many advantages over traditional methods, such as fast response, non-contact, and so on. But most of video smoke detection systems usually have high false alarms. In order to improve the performance of video smoke detection, we propose an accumulative motion model based on the integral image by fast estimating the motion orientation of smoke. But the estimation is not very precise due to block sum. Not very accurate estimation will affect the subsequent decision. To reduce this influence, the accumulation of the orientation over time is performed to compensate results for the inaccuracy of orientation. The model is able to mostly eliminate the disturbance of artificial lights and non-smoke moving objects by using the accumulation of motion. The model together with chrominance detection can correctly detect the existence of smoke. Experimental results show that our algorithm has good robustness for smoke detection.

Keywords: Computer vision; Video smoke detection; Fire detection; Motion estimation; Orientation accumulation

1. Introduction

Many existing fire detectors are based on particle sampling, temperature sampling, relative humidity sampling and smoke analysis (Chen et al., 2003). Those sensors are applied very widely due to their low cost and simplicity. But they must be installed in the close proximity of fire, otherwise they cannot detect fire at all. That's one of major drawbacks of traditional fire detection techniques. So those detectors cannot be successfully applied in large or open spaces. And those detectors often get invalid by dust or other non-fire particles due to direct contact with them. Some detectors make use of infrared sensors to detect flame, but that directly leads to high cost for surveillance. To avoid fire disasters and reduce the cost of surveillance systems, a lot of novel techniques have recently been proposed to improve the performance of fire detection. One of them is vision based fire detection. Vision based fire detection approaches not only detect fire far from the detectors, but also provide abundant visual information about fire. Vision based fire detection can be classified into two categories: video flame detection and video smoke detection.

Up to now, there are a lot of video flame detection algorithms in the literature. Most of methods make use of various visual characteristics including color, motion and geometrical contour of flame regions. Yamagishi and Yamaguchi (1999, 2000) presented a flame detection algorithm based on the spatio-temporal fluctuation data of the flame contour. Their method only used color information to segment flame pixels. However, we can observe that many non-fire objects have the same color distributions as fire. Although accurate parameters of color distribution can be acquired by learning, false segmentation exists inevitably. False segmentation intensively affects the subsequent

* Tel.: +86 791 6335181; fax: +86 791 3891952.
E-mail address: yfn@ustc.edu

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.01.013

decision. Noda et al. (1994) developed a fire detection based on gray scale images applied in tunnels. They analyzed the relationship between temperature and the ratio of G to R, and used the gray level histogram features to recognize fire. Phillips et al. (2000) proposed a flame detection technique based on video. They used the Gaussian-smoothed color histogram to generate a color lookup table of fire pixel and then took advantage of temporal variation of pixel values to determine whether it is a fire pixel or not. The method is insensitive to the motion of camera. But it cannot segment fire pixels very well when there are objects having the same color distribution as fire, and it is very difficult to estimate the location of fire (Healey et al., 1993). Ugur Toreyin et al. (2006) synthetically utilized motion, flicker, edge blurring and color features for video flame detection. Temporal and spatial wavelet transform are performed to extract the characteristics of flicker and edge blurring. We also proposed a video flame detection using the mixture Gaussian model to extract temporal features (Yuan et al., 2005).

In most cases, smoke usually appears before ignition. So detection of smoke provides an earlier alarm of fire. There are three categories of smoke detection (Healey et al., 1993; Aird and Brown, 1997; Cappellini et al., 1989; Vicente and Guillemant, 2002; Wieser et al., 2001). The first category is the histogram based smoke detection. In these techniques, the histogram of smoke is used to detect the presence of smoke in videos. Some statistical measures, such as the mean and standard deviation, are calculated to determine the probability of the presence of smoke. The second one is temporal analysis based techniques. Differences of successive frames and wavelet transform of temporal values of pixels are used to extract time-varying features. The third one is the rule based techniques (Foo, 1996). Knowledge of fire is coded as rules to infer the presence of smoke from image sequences. And some approaches utilized combined techniques to determine the appearance of smoke. In (Ugur Toreyin et al., 2005), motion, flicker, edge blurring and color were used to determine the appearance of smoke. Motion can be detected by simple frame differences or background subtraction. Motion detection is able to eliminate the disturbance of stationary non-smoke objects. A flicker feature was extracted by analyzing the temporal variation of the red channel value in the wavelet domain. Smoke usually blurs the edges of background. This feature of edge blurring was extracted in the wavelet domain. In the beginning of a fire, smoke is semi-transparent. As smoke gets denser, smoke causes a decrease in the U and V chrominance values. Although shadows also have the same property, they do not cause a decrease in the local edge-sharpness within an image frame. Guillemant and Vicente (2001) proposed a smoke detection method for forest fire detection, which includes temporal embedding of gray-levels, fractal indexing of points, chaining points into a linked list and motion extraction from point sequences of the linked list. Ferrari et al. (2007) presented a real-time image processing technique for the detection of steam in video images. In their method, a statistical hidden Markov tree (HMT) model derived from the coefficients of the dual-tree complex wavelet transform (DT-CWT) in small local regions of the image sequences is used to characterize the steam texture pattern. And an SVM classifier is used to detect steam. Gottuk et al. (2006) evaluated the effectiveness of commercial video image fire detection systems for small, cluttered spaces on Navy ships. They found that the system detected more fires and faster than the traditional fire detection systems.

Due to wide applications of inexpensive color cameras in military and commercial securities (Stauffer and Grimson, 2000; Pavlidis et al., 2001), we also use a stationary CCD camera to monitor experimental fires. In this paper, the difference of two successive frames is computed to remove the disturbance of static objects. Being enlightened on the fact that human can easily recognize the presence or absence of smoke only according to the motion of smoke, we decide to utilize motion characteristics for smoke detection. And a fast orientation estimation model is presented to extract the motion characteristics. Using the motion information, we can capture the time varying features of smoke very well. Accurate motion estimation is usually time-consuming. But the video smoke detection system is computationally prohibitive and the system is required to detect smoke interactively. Hence, any time-consuming algorithms are avoided in this speed-critical application. To save computation time, we only estimate the orientation of motion and accelerate the computation using an integral image instead of an iterative process. Although the orientation estimated by our model is not very precise, it is fast and it can be compensated by accumulation.

This paper is organized as follows. Section 2 describes block moving detection and chrominance detection. In Section 3, estimation of motion orientation is presented in detail. In Section 4, acceleration of the estimation method is proposed by integral images (Dempster et al., 1977; Viola et al., 2004). In Section 5, several experiments are described. Section 6 points out drawbacks of the method and briefly describes future work. In the last section, this paper is concluded.

2. Block processing

2.1. Moving detection

There are a lot of approaches to moving detection. Background subtraction, temporal difference of two successive frames and optical flow are the three classic methods for moving detection. In background subtraction based methods, given a model of background image, the values of a particular pixel over time are considered as a stochastic process (Yuan et al., 2005; Viola et al., 2004; Betke et al., 1996; Soh et al., 1995), i.e. a time series of scalars for gray values or vectors for color pixel values. At any time, color value for each pixel belongs to one of several Gaussian distributions. After these statistical methods model the background image, the difference of the background image and current frame is calculated to determine the moving property of each pixel. For the sake of simplicity and computation efficiency, the difference of two successive frames is used for moving detection in our algorithm. The difference between two frames taken at time t and $t-\Delta t$ may be defined as

$$d(x,y,t)=\begin{cases}1 & \text{if } |f(x,y,t)-f(x,y,t-\Delta t)|>T_d\\ 0 & \text{otherwise}\end{cases}\tag{1}$$

where T_d is a predetermined threshold, $f(x,y,t)$ is an image taken at time t.

In the dynamic image analysis, all pixels in the difference image $d(x,y,t)$ with value "1" are considered as moving objects in the scene. As we know, video images usually have a great amount of noises due to intrinsic electronic noises and quantification. So the difference of two successive frames pixel by pixel inevitably produces false segmentation. To reduce the disturbance of noises, we firstly sum up the values of pixels within a block and then calculate the difference of the sums block by block. The block difference is defined as

$$b(i,j,t)=\begin{cases}1 & \text{if } \left|\sum_{x,y\in b_{ij}}[f(x,y,t)-f(x,y,\Delta t-t)]\right|>T_b\\ 0 & \text{otherwise}\end{cases}\tag{2}$$

where T_b is a predetermined threshold, and block b_{ij} is on the ith row and jth column in the video.

Supposed that image width, image height, block width and block height are W_{i}, H_{i}, W_{b} and H_{b}, respectively. So the row number N_{r} and column number N_{c} are expressed by

$$N_{\mathrm{r}}=\left\lfloor\frac{H_{\mathrm{i}}}{H_{\mathrm{b}}}\right\rfloor,\quad N_{\mathrm{c}}=\left\lfloor\frac{W_{\mathrm{i}}}{W_{\mathrm{b}}}\right\rfloor\tag{3}$$

where $\lfloor\rfloor$ is an operator of truncating a floating point number to get an integer number.

To make the system adapted to a variety of lighting, the threshold T_b is automatically specified according to the average of all the block differences. If the average of block difference increases, the threshold T_b should also be able to increase.

From Eq. (3) above, we can infer that it will not process the top and right margins if the image size cannot be divided exactly by the block size on a bottom-up image, as shown in the right of Fig. 1. Fig. 2 illustrates some experimental results produced by block moving detection.

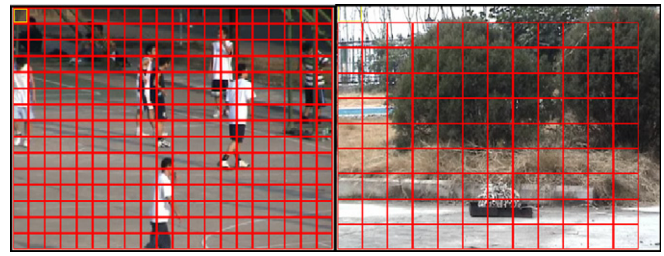


Fig. 1. Block division of video images.

Fig. 2. Results of moving detection block by block.

2.2. Chrominance detection

For many combustible materials, the smoke color will range from white-bluish to white when the temperature of smoke is low and from black-grayish to black when the temperature rises until it catches fire. Color distributions of smoke can be modeled by using a mixture of Gaussians or experiential specification (Chen et al., 2006). We make use of a manual method to specify the possible color range of smoke in the RGB color space, which is slightly similar to the method proposed by Chen et al. (2006).

For a grayish or black smoke pixel, R, G and B values are very close to each other. In other words, the absolute difference of the maximum and minimum values among the three components should be less than a predetermined threshold T_1. And the intensity I of smoke pixel ranges from T_2 to T_3. In some cases, color of smoke may be prone to white-blue. Therefore, the value of B component is a little larger than two other components. The rules are described as follows:

$C_{\min}=\min(R,\ G,\ B)$
$C_{\max}=\max(R,\ G,\ B)$
$I=(R+G+B)/3$
Rule 1: $|C_{\max}-C_{\min}|<T_1$
Rule 2: $T_2<I<T_3$
Rule 3: $C_{\max}=B$ AND $|C_{\max}-C_{\min}|<T_4$
If (**Rule 1** AND **Rule 2**) OR (**Rule 3** AND **Rule 2**)
{
  Smoke pixel
}
Else
{
  Non-smoke pixel
}

In these rules described above, the rule 1 represents characteristics of grayish color of smoke, and the rule 2 limits the range of intensity variation and the rule 3 permits smoke pixels to be white-bluish. T_4 is also a predetermined threshold, which is a little larger than T_1. The intensity I can also be equal to $R^* 0.30 + G^* 0.59 + B^* 0.11$, according to the transformation from the RGB color space into the YUV color space. But for computation performance, I can be simply computed by averaging R, G, and B to avoid floating point computation.

According to these color detection rules, the simple chrominance detection is performed to validate each moving block. If most of pixels in a block pass the chrominance detection, we think that the block is a possible smoke block and the block value is set to 1. Otherwise, the block value should be set to 0.

3. Fast motion orientation estimation

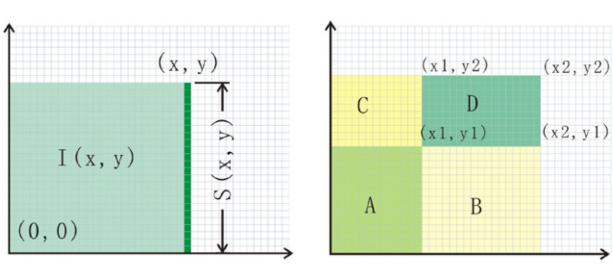
Motion plays an important role in video smoke detection. There are a lot of methods (Wang et al., 2003; Aggarwal and Nandhahumar, 1998; Barron et al., 1994) proposed for motion estimation, such as optical flow. Most of methods are prohibitive because it is time consuming. An integral image based block motion estimation algorithm is presented here. The algorithm only estimates the orientation of motion with eight discrete directions and three pixel steps.

3.1. Integral image

The concept of integral image was first introduced by Viola et al. (2001) and further developed by Lienhart et al. (2002). The integral image techniques greatly speed up extraction of rectangular features and it is widely applied in many areas, such as face detection. As shown in Fig. 3a, an integral image $I(x,y)$ is an intermediate representation for the image $f(x,y)$ and contains the sum of gray scale pixel values of the image with height x and width y, i.e.,

$$I(x,y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} f(x',y') \tag{4}$$

The integral image is computed recursively, by the following formulas:



(a) Recursive computation (b) Block sum by Integral Image

Fig. 3. Integral image.

$$\begin{aligned} I(x,y) &= I(x-1,y) + S(x,y) \\ S(x,y) &= S(x,y-1) + f(x,y) \end{aligned} \tag{5}$$

where $S(x,y,t)$ is the column integral, expressed by

$$S(x,y) = \sum_{y'=0}^{y} f(x,y') \tag{6}$$

And we define $S(-1,y) = 0$ and $S(x,-1) = 0$. Therefore, recursive computation of integral image requires only one scan over the input data. This recursive computation is very efficient.

According to the attribute of integral image and after observing Fig. 3b, we have

$$I(x_1,y_1) = \sum_{x,y \in A} f(x,y) \tag{7}$$

$$I(x_2,y_1) = \sum_{x,y \in A} f(x,y) + \sum_{x,y \in B} f(x,y) \tag{8}$$

$$I(x_1,y_2) = \sum_{x,y \in A} f(x,y) + \sum_{x,y \in C} f(x,y) \tag{9}$$

$$\begin{aligned} I(x_2,y_2) = &\sum_{x,y \in A} f(x,y) + \sum_{x,y \in B} f(x,y) \\ &+ \sum_{x,y \in C} f(x,y) + \sum_{x,y \in D} f(x,y) \end{aligned} \tag{10}$$

So, the sum of pixel value within a block with the bottom left point (x_1,y_1) and top right point (x_2,y_2) is computed as follows:

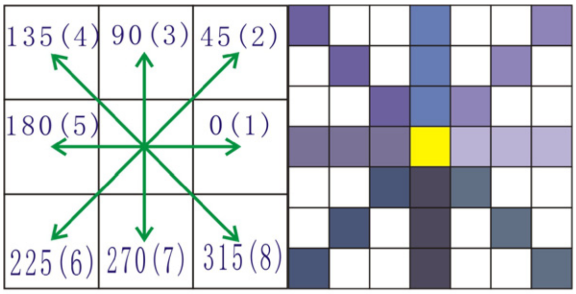
$$\begin{aligned} S(x_1,y_1,x_2,y_2) &= \sum_{y=y_1}^{y_2} \sum_{x=x_1}^{x_2} f(x,y) \\ &= I(x_1,y_1) + I(x_2,y_2) - I(x_1,y_2) \\ &\quad - I(x_2,y_1) \end{aligned} \tag{11}$$

From Eq. (11), we can see that the sum is only computed with several lookups. So the block size does not influence computation performance and the sum by integral image is very computationally efficient.

3.2. Motion orientation estimation

The velocity is a vector composed of a magnitude and an orientation. In this paper, only the orientation of motion is estimated from sequential images to save computation time. Accurate estimation is usually time-consuming. So we use a block motion orientation model and suppose that all pixels in the same block have the same motion parameters. As shown in Fig. 4a, the orientation of motion is discretized into 8-directions which are 0, 45, 90, 135, 180, 225, 270 and 315 degrees. These discrete directions are coded as 1, 2, 3, 4, 5, 6, 7 and 8, respectively.

Since the direction is discrete, the searching displacement needs to be also discretized at each direction. After analyzing a lot of smoke videos, we find that the 3-pixel



(a) Discrete directions (b) Searching scheme

Fig. 4. Discrete direction of motion.

displacement is enough in most cases. And our objective is to estimate motion orientation, not motion magnitude. The search scheme is shown in Fig. 4b. And the error function for matching is defined as follows:

$$E(i,j,\theta,\text{dis}) = \left| \sum_{x,y \in b(i,j)} f(x,y,t) - \sum_{x,y \in b'(i,j)} f(x,y,t-\Delta t) \right| \tag{12}$$

where $b(i,j)$ is the ith row and jth column block at time t, $b'(i,j)$ is the searching block at time $t-\Delta t$, dis is the displacement, and θ is the direction code. And $b'(i,j)$ is uniquely determined by dis and θ.

For each direction, we compute the error function with displacement dis = 1, 2 and 3. Hence, each direction has three error values. We define the minimum error as the error of the direction

$$E(i,j,\theta) = \min_{\text{dis}} E(i,j,\theta,\text{dis}) \tag{13}$$

And we think that the motion orientation is likely to be along the direction where the error value is minimal. So we have

$$\theta = \operatorname{Arg}\min_{\theta} E(i,j,\theta) \tag{14}$$

After analyzing the relationship between the minimum error and the motion orientation, we find that the direction is exactly contrary to the real motion direction. So the direction code must be modified by

$$\theta = \begin{cases} 8 & \text{if } \operatorname{mod}(\theta+4,8) = 0 \\ \operatorname{mod}(\theta+4,8) & \text{else} \end{cases} \tag{15}$$

where $\operatorname{mod}(x,y)$ is a function that divides one numeric x by another numeric y and returns the remainder.

According to Eq. (2), we find that some blocks may have no motion. So when a block is stationary, the motion orientation needs not to be estimated and the direction is coded as 0 for the static block.

3.3. Estimation acceleration by integral image

If the block is stationary, the Eq. (12) is only computed one time. For a changing block, we must compute the differences at 8-directions with 3-pixel displacement, so it totals to $8^* 3+1$ times. Therefore, we have to compute the Eq. (12) at least one time for stationary blocks and 25 times for a changing block. As shown in Eq. (12), direct calculation of the formula is computationally expensive. As described above, the sum of pixel values within a block has the same computation time for different block sizes. So the error function can be computed by

$$\begin{aligned} E(i,j,\theta,\text{dis}) = |I(x_1,y_1,t) + I(x_2,y_2,t) - I(x_1,y_2,t) \\ - (x_2,y_1,t) - I(x'_1,y'_1,t-\Delta t) - I(x'_2,y'_2,t-\Delta t) \\ + (x'_1,y'_2,t-\Delta t) + I(x'_2,y'_1,t-\Delta t)| \end{aligned} \tag{16}$$

where (x_1,y_1) and (x_2,y_2) are the bottom left and top right points of block $b(i,j)$,and (x'_1,y'_1) and (x'_2,y'_2) are the bottom left and top right points of block $b'(i,j)$ determined by the searching parameters.

The block motion orientation can be computed interactively by using the integral image. Fig. 5 illustrates the results of motion orientation estimated by our algorithm. We can see that it has low accuracy. In despite of low accuracy, it proves effective for smoke detection in the following section.

4. Orientation accumulation

The estimated orientation is not very precise but computationally efficient. This imprecise orientation will severely affect the results of subsequent decision. So we present an accumulation method to reduce the influence. Let us consider the sequence of motion orientation $\theta(i,j,t)$ at a block b_{ij} along the time axis t, and the temporal histogram $H(i,j,t)$ of motion direction is calculated on a time window W_T, as shown in Fig. 6. We naturally think that the direc-

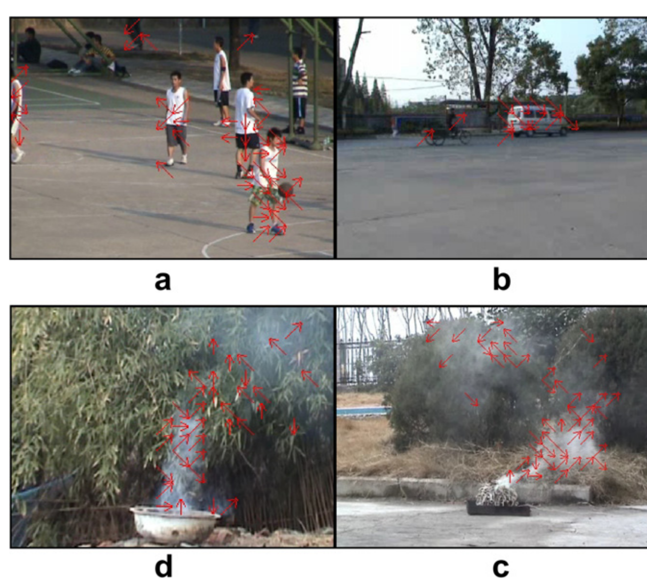


Fig. 5. Each arrow stands for motion orientation. (a) and (b) are results of motion orientation estimated by our algorithm for a playground video and a traffic video, respectively. (c) and (d) are results for a smoke video of dry leaves and a smoke video of 60 cotton ropes, respectively.

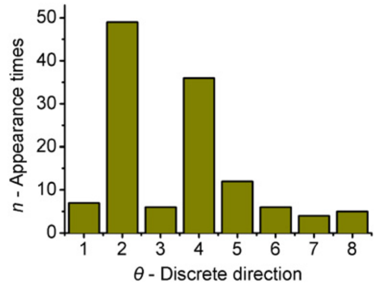


Fig. 6. Histogram of a series of directions within a time window.

tion code with maximal entry in the histogram is considered as the motion orientation of the block. In the case of Fig. 6, the motion orientation code of the block is 2, i.e., the orientation is actually 45 degrees. So the motion orientation of block b_{ij} at time t can be expressed by

$$\theta(i,j,t) = \arg\max_{\theta}\{H(i,j,t,\theta)\} \tag{17}$$

The direction with maximal entry in the histogram can also be regarded as the maximum value of accumulation of each direction along the time axis. Fig. 7b demonstrates the results of motion accumulation.

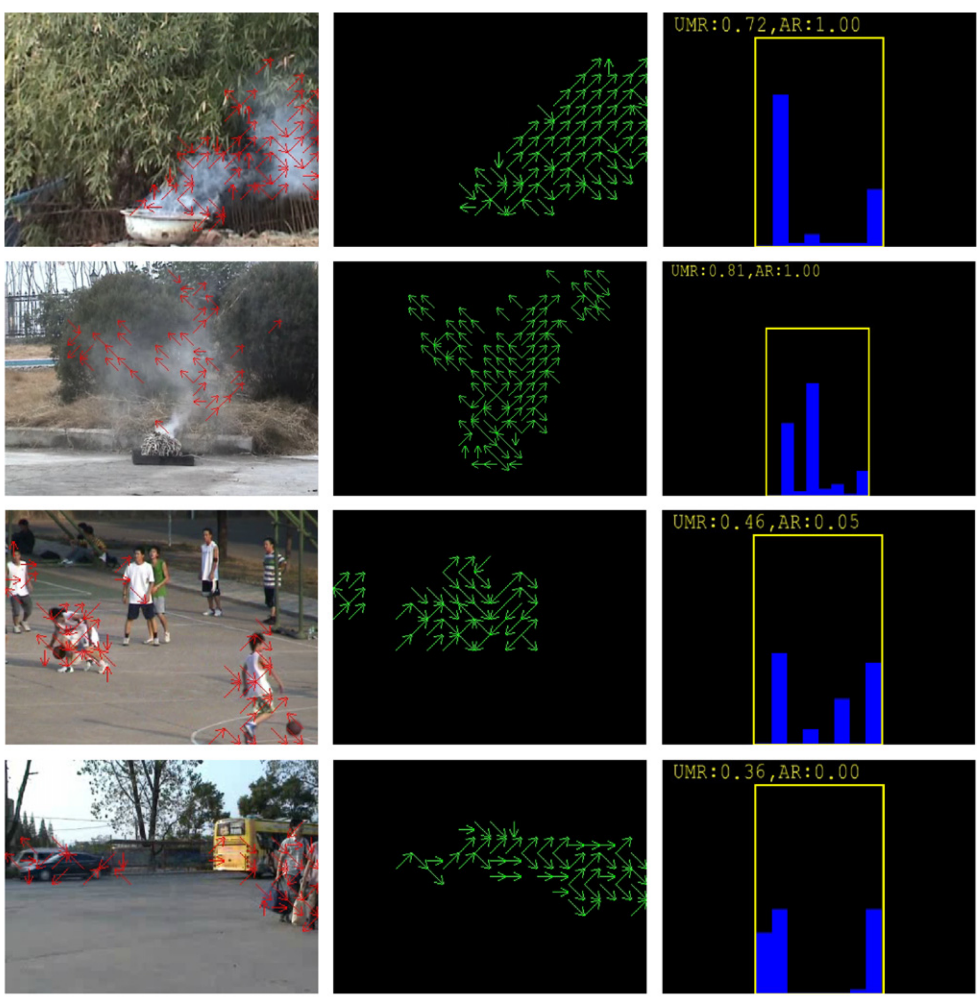
5. Experimental results

Our motion model can extract the accumulative motion orientation of each block as shown in Fig. 7b. Because smoke usually drifts upwards continually by hot airflows, the accumulative orientation code of smoke block is upwards, i.e., 2, 3 and 4. In fact, we can find the phenomenon by carefully observing the Fig. 7. Fig. 7c gives the corresponding spatial histogram $H_b(\theta)$ of the block orientation codes in a frame. To validate features obtained by our motion model and simplify the complexity of recognition, a simple feature is computed by

$$\mathrm{UMR} = \frac{\sum_{\theta=2}^{4} H_b(\theta)}{\sum_{\theta=1}^{8} H_b(\theta)} \tag{18}$$

where UMR is the ratio of the sum of frequencies at upward motion directions to the sum at all directions in the histogram.

Fig. 8 gives the UMR curves for the four experimental videos. For the similar purpose of simplicity, we use a thresholding process to segment the UMR feature to determine



(a) Motion orientation (b) Orientation accumulation (c) Histogram of accumulation

Fig. 7. Results for two smoke and two non-smoke experiments.

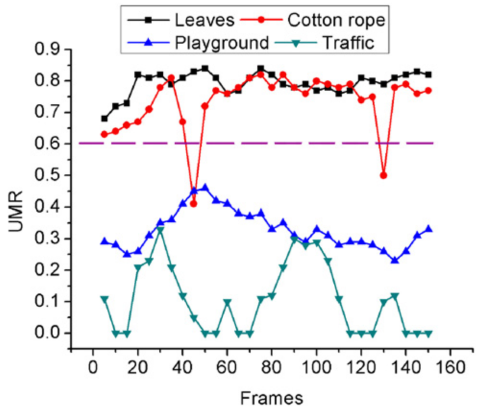


Fig. 8. UMR values over time.

whether there is smoke in the scene or not. In Fig. 8, when the threshold is set to 0.6, smoke and non-smoke objects can be easily discriminated. In order to reduce the false alarm, accumulation of alarms is also used. In other words, only an alarm that has last for a few of seconds is considered as a reliable fire alarm.

The algorithm presented in this paper is implemented using Visual C++, and tested on a PC platform. We used two smoke videos and two non-smoke videos to validate our video smoke detection model. For smoke videos as shown in the 1st and 2nd rows of Fig. 7, our model can accurately raise smoke alarms in time. The 1st row was a smoke video produced by dry leaves. All the UMR values over time are greater than the threshold. The 2nd row was a smoke video of cotton ropes. Only few UMR values are below the threshold due to strong wind in the scene, so these values can be considered as noises. Accumulation of alarms will eliminate the influences of these noisy UMR values and the system can also give the fire alarm in time. As for non-smoke videos as shown in the 3rd and 4th rows of Fig. 7, our model did not give any alarm all the time. The 3rd row gives a typical playground video including several students moving irregularly. And the 4th row illustrates an ordinary traffic video on a street. The two videos did not raise any alarm. Experiments show that our proposed model can extract robust features for video smoke detection.

We also have tested our algorithm on two videos publicly free available from http://signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html. As shown in the 1st rows of Fig. 9, it is a smoke video behind a fence, and our algorithm can raise smoke alarms in time. As shown in the 2nd rows of Fig. 9, our model did not give any alarm all the time. The 2nd row shows a typical car light video.

6. Future work

For the efficiency of computation, we did not estimate the whole vector of motion. We used a simplified searching scheme to estimate the motion orientation and the impact of the simple motion model was compensated by subsequent accumulation. So in future work, we will explore fast and accurate method to estimate motion for smoke detection. Because we assume that smoke usually drifts upward, the model is insufficient in some specialized and rarely occurring cases where there is no upward drifting smoke. So in the future, we will analyze the pattern of velocity field over time and make use of SVM, HMM, et al. to overcome this disadvantage.

7. Conclusions

In this paper, an accumulative motion model is proposed. Although the motion is roughly estimated, the computation is efficient and the precision is improved by accumulation. The estimation is greatly accelerated by using the integral image technique. This not very accurate estimation will affect the subsequent decision. Therefore, an accumulation of the orientation over time is presented

(a) Motion orientation (b) Orientation accumulation (c) Histogram of accumulation

Fig. 9. Experimental results for two videos publicly available.

to compensate results for the inaccuracy of orientation. The model can mostly eliminate the disturbance of artificial lights and non-smoke moving objects by using the accumulation of motion. In order to further reduce the false alarm, the accumulation of alarms is also used. Experiments show that our model can detect the presence of smoke correctly in most cases.

Acknowledgements

This project was supported by Open Foundation of the State Key Laboratory of Fire Science (HZ2006-KF03), National Science & Technology Supporting Program (2006BAK06B07), China postdoctoral science foundation (20070410792), Natural Science Foundation of Jiangxi province (2007GQS0076) and Foundation of Education Department of Jiangxi province (2007[272]). We are grateful to Professor Zhang Yongmin and Liao Guangxuan of the State Key Lab of Fire Science who highly supported the work and offered helpful suggestions.

References

Aggarwal, J.K., Nandhahumar, N., 1998. On the computation of motion from sequences of images – A review. Proc. IEEE, 917–935.

Aird, B., Brown, A., 1997. Detection and alarming of the early appearance of fire using CCTV cameras. In: Nuclear Engineering Internat. Fire & Safety Conference, London, 24–26 February.

Barron, J.L., Fleet, D.J., Beauchemin, S.S., 1994. Performance of optical flow techniques. Internat. J. Comput. Vision 12, 43–77.

Betke, Margrit, Haritaoglu, Esin, Davis, Larry S., 1996. Multiple vehicle detection and tracking in hard real time [J]. IEEE (9), 351–356.

Cappellini, V., Mattii, L., Mecocci, A., 1989. An Intelligent System for Automatic Fire Detection in Forests. University of Florence, Italy.

Chen, Thou-Ho, Kao, Cheng-Liang, Chang, Sju-Ma, 2003. An intelligent real-time fire-detection method based on video processing, In: IEEE 37th Annual 2003 Internat. Carnahan Conf. on Security Technology, October 14–16, pp. 104–111.

Chen, Thou-Ho, Yin, Yen-Hui, Huang, Shi-Feng, et al., 2006. The smoke detection for early fire-alarming system base on video processing. In: Proc. 2006 Internat. Conf. on Intelligent Information Hiding and Multimedia Signal Processing.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. B 39, 1–3.

Ferrari, R.J., Zhang, H., Kube, C.R., 2007. Real-time detection of steam in video images. Pattern Recognition 40 (3), 1148–1159.

Foo, Simon Y., 1996. A rule-based machine vision system for fire detection in aircraft dry bays and engine compartments. Knowl.-Based Systems (9), 531–540.

Gottuk, D.T., Lynch, J.A., Rose-Pehrssonb, S.L., Owrutskyb, J.C., Williams, F.W., 2006. Video image fire detection for shipboard use. Fire Safety J. 41, 321–326.

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

Healey, G., Slater, D., Lin, T., Drda, B., Goedeke, A.D., 1993. A system for real-time fire detection. In: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition, 15–17 June, pp. 605–606.

Lienhart, R., Kuranov, A., Pisarevsky, V., 2002. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. Intel Corporation, Santa Clara, USA.

Noda, S., Ueda, K., 1994. Fire detection in tunnels using an image processing method. In: Proc. Vehicle Navigation and Information Systems Conference, 31 August–2 September, pp. 57–62.

Pavlidis, I., Morellas, V., Tsiamyrtzis, P., Harp, S., 2001. Urban surveillance systems: From the laboratory to the commercial world. Proc. IEEE 89 (10), 1478–1497.

Phillips III, Walter, Shah, Mubarak, Lobo, Niels Da Vitoria, 2000. Flame recognition in video. In: Fifth IEEE Workshop on Applications of Computer Vision, 4–6 December, pp. 224–229.

Soh, Jung, Chun, Byung Tae, Wang, Min, 1995. Analysis of road image sequences for vehicle counting [J]. IEEE Internat. Conf. on Systems, Man and Cybernetics (1), 679–683.

Stauffer, C., Grimson, W.E.L., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 747–757.

Ugur Toreyin, B., Dedeoglu, Yigithan, et al., 2006. Pattern Recognition Lett. 27 (1), 49–58.

Ugur Toreyin, B., Dedeoglu, Yigithan, Cetin, A. Enis, 2005. Wavelet based real-time smoke detection in video. In: 13th European Signal Process Conf. EUSIPCO2005, Antalya, Turkey.

Vicente, Jerome, Guillemant, Philippe, 2002. An image processing technique for automatically detecting forest fire. Internat. J. Therm. Sci. (4), 1113–1120.

Viola, P., Jones, M., 2001. Robust real time object detection. In: IEEE ICCV Workshop Statistical and Computational Theories of vision, Vancouver, Canada.

Viola, P., Jones, M., Robust, J., 2004. Real-time face detection. Internat. J. Comput. Vision 57 (2), 137–154.

Wang, Yao, Ostermann, Jorn, Zhang, YaQin, 2003. Video Processing and Communications. Pearson Education, Inc.

Wieser, Dieter, Brupbacher, Thomas, 2001. Smoke detection in tunnels using video images. In: 12th Internat. Conf. on Automatic Fire Detection, March 25–28, Maryland, USA.

Yamagishi, H., Yamaguchi, J., 1999. Fire flame detection algorithm using a color camera, 1999. MHS ’99. In: Proc. 1999 Internat. Symposium on Micromechatronics and Human Science, 23–26 November, pp. 255–260.

Yamagishi, H., Yamaguchi, J., 2000. A contour fluctuation data processing method for fire flame detection using a color camera. In: IEEE 26th Annual Conf. on IECON of the Industrial Electronics Society, vol. 2, 22–28 October, pp. 824–829.

Yuan, Feiniu, Liao Guangxuan, Fan, Weicheng, Zhou, Heqin, 2005. Vision based fire detection using mixture Gaussian model. In: The 8th Internat. Symposium on Fire Safety Science 2005, September 18–22, Beijing, China.