Fire Detection Based on Hidden Markov Models

Article *in* International Journal of Control Automation and Systems · August 2010

DOI: 10.1007/s12555-010-0414-2

3 authors:

Zhu Teng
Beijing Jiaotong University
28 PUBLICATIONS **183** CITATIONS
SEE PROFILE

Jeong-Hyun Kim
Pusan National University
161 PUBLICATIONS **1,470** CITATIONS
SEE PROFILE

Dong Joong Kang
Pusan National University
82 PUBLICATIONS **569** CITATIONS
SEE PROFILE

International Journal of Control, Automation, and Systems (2010) 8(4):1-9 http://www.springer.com/12555
DOI

Fire Detection Based on Hidden Markov Models

Zhu Teng, Jeong-Hyun Kim, and Dong-Joong Kang*

Abstract: In this paper, a novel method of real-time fire detection based on HMMs is presented. First, we present an analysis of fire characteristics that provides evidence supporting the use of HMMs to detect fire; second, we propose an algorithm for detecting candidate fire pixels that entails the detection of moving pixels, fire-color inspection, and pixels clustering. The main contribution of this paper is the establishment and application of a hidden Markov fire model by combining the state transition between fire and non-fire with fire motion information to reduce data redundancy. The final decision is based on this model which includes training and application; the training provides parameters for the HMM application. The experimental results show that the method provides both a high detection rate and a low false alarm rate. Furthermore, real-time detection has been effectively realized via the learned parameters of the HMM, since the most time-consuming components such as HMM training are performed off-line.

Keywords: Fire detection, HMM, real-time processing, visual surveillance.

1. INTRODUCTION

A fire detection system that enables accurate and prompt detection of fire is one of the most significant components of video surveillance and monitoring systems, since fire can cause loss of life and severe damage to property. Conventional methods generally utilize narrow range detectors such as chemical or gas sensors; however, these conventional sensors are sub-optimal. First, the accuracy of a fire detection system closely depends on the precision and reliability of its sensors, the size of the sensing space, and the distribution of the sensors. If a higher precision fire detection system is needed, it is necessary to distribute the sensors densely in the detection space. Second, false detection can easily occur from other sources of smoke or fire, including even a lit cigarette. Third, there are fatal time delays for sensors to detect fire or smoke, which may result in the spread of fire, because the alarm is not issued until fire gas or chemical particles reach the sensors and activate them. Lastly, the expense of sensors can impact the system cost. In contrast, video processing techniques offer many advantages. Video surveillance and monitoring systems have already been used in many buildings and human environments due to rapid developments with affordable digital cameras and video processing techniques. Thus, there is no need to factor in additional costs, and it is only necessary to add software to process the output of the surveillance and monitoring system in real-time. Moreover, a video-based method can detect fires earlier and achieve better reliability.

In recent decades, video processing techniques for fire detection have been studied by many researchers [1-7]. Primarily, only color clues of the fire were used [2]. Fire pixel classification in color video sequences was developed in [3], in which fire was detected via three fire-color conditions that were established in an RGB color system and a dynamics analysis of flames. Recently, the motion and geometry of fire regions have also been utilized, in combination with the fire-color conditions. For instance, a wavelet analysis was added in order to analyze high-frequency information about fire [4]. Although good results were demonstrated for several kinds of test data, the method used too many heuristic thresholds, which is impractical in real-life applications. A rule-based generic color model for flame pixel classification was introduced in [5], which proposed an algorithm that used YCbCr color space to separate the luminance from the chrominance. Although testing involved two sets of images that contained fire and fire-like regions, only color information was used, and this was insufficient. A fire detection method based on HMMs (Hidden Markov Models) was presented [6]; however, the method of HMM training was slightly simplistic, entailing only a simple count of the number of state transitions for each state followed by calculation of the fraction of total state transitions as the transition probabilities. As such, this method fails to fully exploit the HMM.

In this paper, we employ the three fire-color

Manuscript received May 4, 2009; revised January 26, 2010; accepted March 5, 2010. Recommended by Editorial Board member Jang Myung Lee under the direction of Editor Jae-Bok Song. This research was financially supported by theMinistry of Education, Science Technology (MEST) and Korea Institute for Advancement of Technology (KIAT) through the Human Resource Training Project for Regional Innovation, and partially supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0090165).

Zhu Teng, Jeong-Hyun Kim, and Dong-Joong Kang are with the School of Mechanical Engineering, Pusan National University, 30 Jangjeon-dong, Geumjeong-gu, Busan 609-735, Korea (e-mails: {tengzhu, mare, djkang}@pusan. ac.kr).
* Corresponding author.

conditions in RGB color space developed by Chen [3] and improve the original method by combining the motion of fire with the learning module of a HMM, which leads to reduction of data redundancy and improvement of reliability. The HMM is a mature decision model that has been successfully applied in many areas such as speech recognition, segmentation, motion analysis and objects tracking in video. This paper is organized as follows. In Section 2, we present an analysis of fire characteristics that establishes the rationale for using HMMs in fire detection. The proposed candidate fire pixels detection method is illustrated in Section 3, which includes moving pixels detection (3.1), fire-color pixels detection (3.2), and pixels clustering (3.3). The final fire pixels decision based on HMMs is described in Section 4, including HMM training (4.1) and HMM application (4.2). Experimental results are provided in Section 5 and our conclusion is presented in Section 6.

2. FIRE CHARACTERISTICS

Prior to dealing with formal fire detection, it is necessary to analyze the characteristics of fire in order to provide the rationale for the use of our technique in fire detection.

We collect data generated by the values of red components of fire and non-fire pixels with respect to frames in order to create two databases, a fire database and a non-fire database. The fire database is achieved via monitoring the red color component of pixels that are already known as fire pixels in most frames, while the non-fire database is obtained by observing background pixels or red color component of pixels that have similar motions with fire but are known to be non-fire pixels. In order to explain the differences between fire pixels and ordinary red pixels, behaviors in the frequency-domain of fire and non-fire data are shown in Fig. 1. The original non-fire data in Fig. 1 is a pixel with ordinary red color. The vertical axis of the graphs of the original database is the red component of the pixel in RGB space. Frequency graphs are obtained by executing a Fast Fourier Transformation (FFT) based on the original data; since the graph is symmetrical, only non-negative values are presented. The vertical axis and the horizontal axis of the frequency-domain graphs represent the magnitude and frequency, respectively.

Figs. 1(a) and (b) show that the fire data have other frequencies besides zero, while the non-fire data only have frequency in the neighborhood of zero. Based on the graphs in (c) and (d), fire varies more drastically and regularly with the elapse of time compared to the non-fire situation, which leads to a greater change in the rate of fire than that of the non-fire situation. For example, if a pixel at a fixed position in an image is thought to be a fire pixel, it may vary very frequently from a fire state to a non-fire state (for example, a sequence such as F(fire) NF(non-fire) F NF F F NF), but if it is a non-fire pixel, the frequency will be slightly lower (for example, F(fire) NF(non-fire) NF NF NF F NF). Markov models

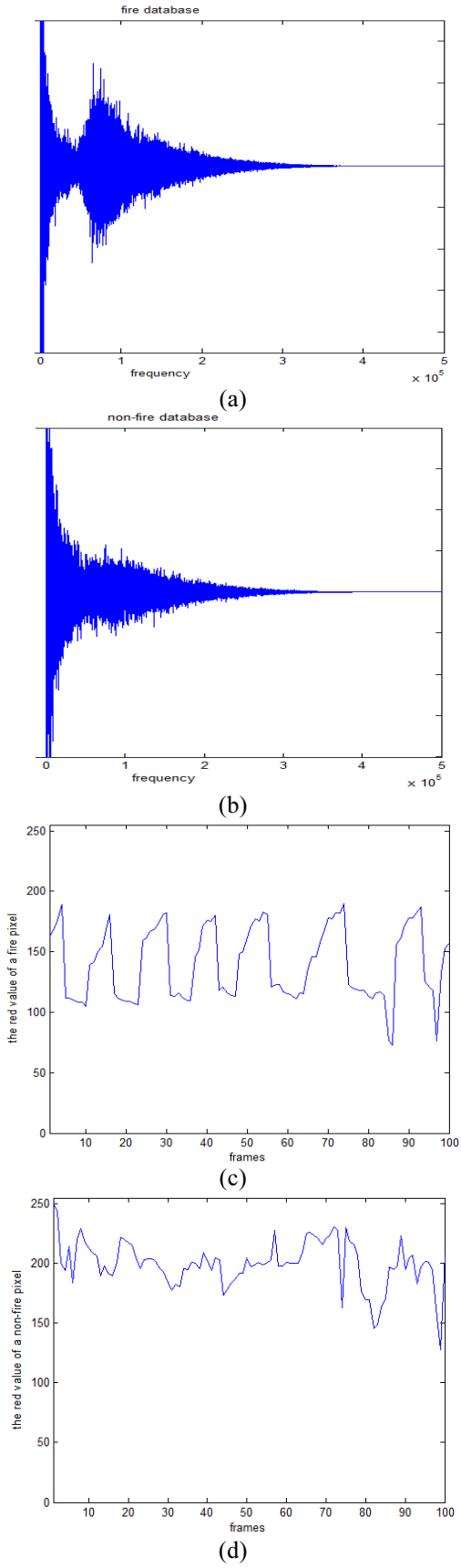


Fig. 1. Signal properties of fire and non-fire pixels in video sequences. Frequency-domain graph of (a) fire and (b) non-fire pixel values; Red channel of (c) fire and (d) non-fire pixel values.

successfully treat this kind of time varying problem under a probabilistic or statistical framework. An HMM is a doubly stochastic process with an underlying stochastic process that is not observable, but can only be observed through another set of stochastic processes that produce the sequence of observed symbols. In our case, the observed stochastic process is the variance of the red color component of pixels and the hidden stochastic process is the change of states between fire and non-fire. HMMs learn time-varying patterns that distinguish between fire and non-fire. For further analysis, we calculate the mean value and the standard deviation value of fire and non-fire situations, respectively. Forty successive data items can generate one mean value and one standard deviation. The mean and standard deviation values of 1000 sequences in the fire situation are 206.56 and 34.58, while the mean and standard deviation values in the non-fire situation are 184.06 and 21.45. This indicates that the fire and non-fire situations are distinguishable. Thus, it is suitable to use HMMs in fire detection. Further details about HMMs are provided in Section 4.

3. FIRE PIXELS DETECTION

The input data of the system is streamed video, and hence it is necessary to convert the video into a series of frames at a fixed rate in order to facilitate the later processing steps. The candidate pixels for fire decision are obtained by detecting pixels in motion regions, checking fire-color pixels and clustering pixels based on a series of frames.

3.1. Moving pixels detection

We detect the pixels in motion regions in the first step by utilizing one of the most significant characteristics of fire, which is that it continues moving as time advances. This characteristic is exploited to filter most of the unwanted information.

Generally, moving pixels are extracted by subtracting the intensity values of two adjacent frames, h_{t+1} and h_t, as shown in equation (1). In our work, we define the function $h_t(X,Y)$ as shown in (2), which denotes the mean value of three adjacent frames for each pixel.

$$Diff_t(X,Y) = | h_{t+1}(X,Y) - h_t(X,Y) |, \quad (1)$$

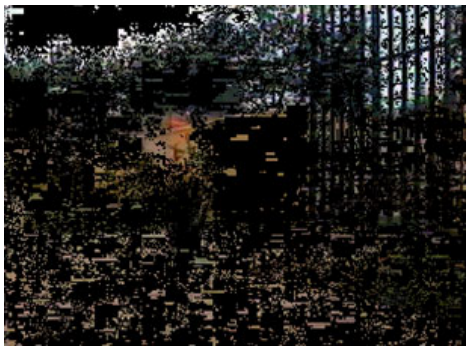
$$h_t(X,Y) = \frac{f_t(X,Y) + f_{t+1}(X,Y) + f_{t+2}(X,Y)}{3}. \quad (2)$$

This simple improvement can result in significant progress. In particular, relative to the two-frame method, useful information is not omitted while excessive and unnecessary information is not passed on to the next step (Fig. 2).

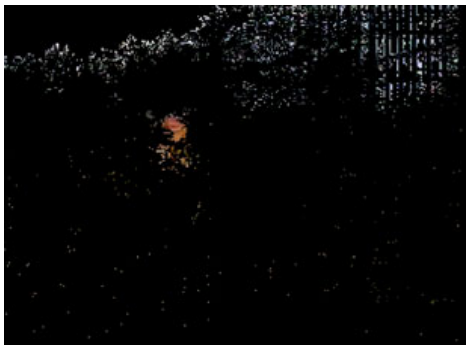
3.2. Fire-color pixels detection

Color is another important property of fire. In this step we deal with pixels generated in motion detection, that is, only moving pixels are checked in terms of whether they satisfy the fire-color conditions.

(a) Original image.



(b) Two frame difference method.



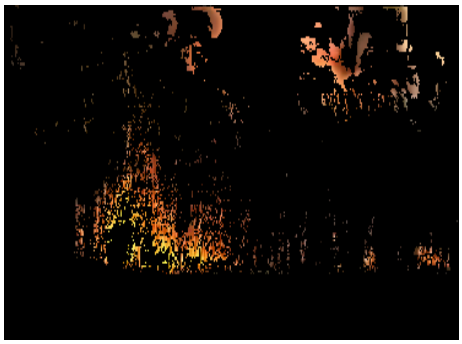
(c) Our method.

Fig. 2. Detecting motion pixels according to image difference. The upper part of the image results from the movement of the tree and bushes because of wind.

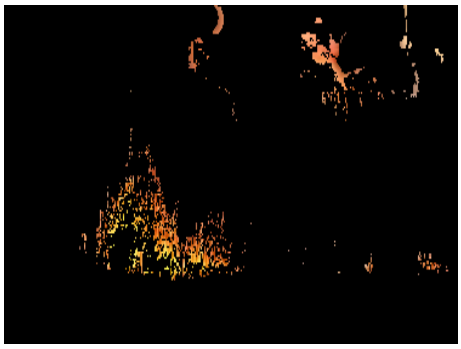
Many researchers have studied features of fire color, and they concluded that the color range of fire is red-to-yellow. This is denoted by $R \geq G$, $G > B$, and is the first of three fire-color conditions [3,7]. Moreover, since the red channel in an RGB image of fire flames is the major component and the red channel value of fire is larger than common objects, we must define what constitutes a strong R value in the captured fire image. Therefore, the second condition of fire color is defined as the value of R that exceeds a threshold R_T. The last property that should be considered is the effects of the background illumination. In order to remove any fire-like color, which may result in false fire detection, the saturation value of the extracted fire flame must exceed a given threshold [3].

According to these principles, the three decision rules used to extract fire color can be established as:

(a) Original image.



(b) Result of moving pixels detection.



(c) Result of fire-color pixels detection.

Fig. 3. Results of applying fire-color conditions.

Rule 1: $R \geq G, G > B$

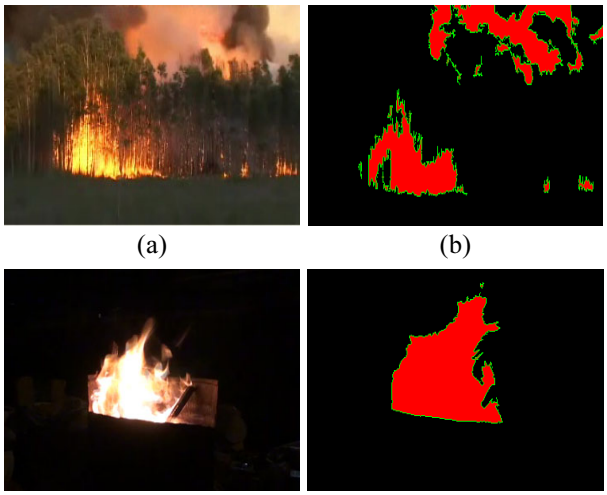
Rule 2: $R > R_T$

Rule 3: $S \geq (255 - R) \times S_T / R_T$.

In Rule 3, S_T denotes the corresponding saturation value when the value of the R term is R_T for the same pixel. The term of $(255 - R) \times S_T / R_T$ illustrates when the R component increases toward the upmost value 255 and then saturation will decrease downward to zero [1,3]. We set S_T and R_T to 60 and 170, respectively on the basis of our experiments, and the application of the above three rules to our algorithm generated the results shown in Fig. 3.

3.3. Pixels clustering

This step involves pixels that satisfy the color conditions. It is shown in Section 2 that fire is a dynamic process and the illumination of fire continuously changes. Consideration of only whether fire exists or not in a single frame leads to conclusions that are not robust. For example, when there is a situation where one pixel is labeled as a fire-color pixel at frame t, then it might be labeled as a non-fire-color pixel in the next few frames, and then the pixel could be labeled as a fire pixel again, because of the dynamic character of fire. This kind of pixel cannot be ignored, since it might be a fire pixel. The best means to determine the fire pixels is to make a cluster of a few continuous frames (See Fig. 4). In our work, one cluster is composed of forty frames. In the next step, we will make the final decision for every cluster. Fig. 5 presents a diagram of candidate fire pixels detection, where the Frame Number is the number of frames stored in the Queue Buffer. The Queue Buffer is a First-In-First-Out (FIFO) data structure with a fixed length of 40. That is, it stores 40 images. The first image added to the queue will be the first to be removed: when a new frame is updated, it will remove the oldest one and store the new one.



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

Fig. 4. Results of clustering during 40 frames. (a)&(c) Original image; (b)&(d) Clustering image.

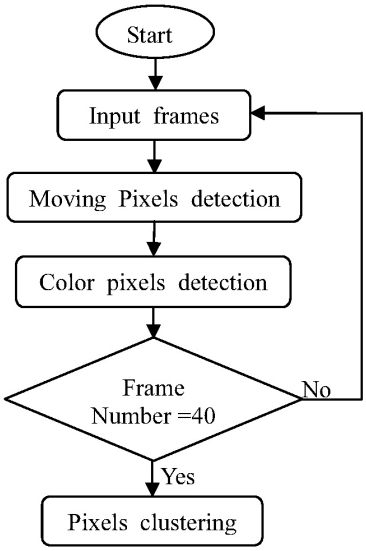


Fig. 5. Schematic diagram of candidate fire pixels detection.

4. FIRE PIXELS DECISION BASED ON HIDDEN MARKOV MODELS (HMMS)

It has been shown that the frequency of flame flicker is about 10 Hz, regardless of the fuel type or the burner size tested [8]. In particular, the boundaries of flame flicker frequently change every second, regardless of the cause of the fire. In addition, flame flicker has a random occurrence. This can also be identified from the images of the fire database in Section 2. In order for a sequence of random variables to be a Markov chain, the conditional probabilities must only be a function of the last random variable [9]; that is, the future state is independent of the past states. In contrast, in the case of fire, the variation in the flicker and oscillations of fire in the next frame depends on the state of the current frame, this satisfies the condition for a Markov chain. Hence, the fire case simply accords with the precondition for the Markov chain, and it is theoretically feasible to detect fire via a HMM.

In our case, fire and non-fire HMMs are constituted by model training via the fire and non-fire databases, which are extracted from a large number of videos. A test observation sequence is then compared with the two respective models; and based on the similarity between the test sequence and the model, the result with the greater similarity indicates the model to which the sequence most likely belongs.

4.1. HMM training

In HMM training, we aimed to find the parameters of HMMs associated with our two databases. The input of the HMM algorithm is states. We define three HMM states: F1, F2, and NF. NF expresses a non-fire state, and F1 and F2 indicate two fire states. Theoretically, the greater the number of HMM states, the better the HMM performance. However, it is difficult to perform experiments if too many states are defined. Therefore, we must reach a compromise between these two factors. According to the variation of the color of the fire flame, several states of fire can be defined. Based on the above analysis, we define two fire states. State F2 indicates a more considerable fluctuation of the red channel value in RGB space compared to F1.

The items in databases represent color values; in contrast, the input parameters of the HMM are denoted by states.

For instance, the database sequence and the input data of the HMM, respectively, can be expressed as:

{150 186 234 200 0 205 0 206 0 …}
{NF F1 F2 F1 NF F1 NF F1 NF…}.

Therefore it is necessary to use a converting algorithm to change color values to states. While a converting algorithm was proposed in [6], the method has some redundant calculations as the moving property of the fire is not considered in the algorithm. Based on this point, we improve the algorithm delineated in Algorithm 1.

Algorithm 1: Converting color value to state

Input: C(i), C(i-1), State(i-1).
Output: State(i).
if Current Pixel is not a moving pixel;
State(i)=NF;
else if State(i-1) is NF;
State(i)=F1;
else if |C(i)-C(i-1)| < th1;
State(i)= State(i-1);
else if |C(i)-C(i-1)| < th2;
State(i)= $\overline{State(i-1)}$;
else State(i)=NF;
End of Algorithm 1

In Algorithm 1, C(i) is defined as: if the pixel being tested is not a moving pixel, the color value is set to zero, otherwise the color value is equal to the red channel value of the pixel in the RGB color representation. If the pixel is not a moving pixel, then we do not consider it as a possible fire pixel, and hence we directly set the state to NF. State(i-1) represents the state of the previous pixel. The index i denotes a pixel of the current frame and $i-1$ denotes a pixel of the previous frame. $\overline{State(i-1)}$ denotes that if State(i-1) is F2, then $\overline{State(i-1)}$=F1, and if State(i-1) is F1, then $\overline{State(i-1)}$ is F2. If the difference of the current color value and the previous color value is not larger than th_1, which implies that the fluctuation is not drastic enough to evolve the states from F1 to F2, then the former state holds; if the difference of the current color value and previous color value is larger than th_2, which suggests that the variation between the current and previous color value is beyond the possible scope of fire variance, then the state is set to NF. We set the value of th_1 and th_2 to 10 and 30 according to our experiments.

Training data is expressed as states via the above algorithm, and data are then trained by the EM (Expectation-Maximization) algorithm [10,11]. The training structure is shown in Fig. 6. The number of

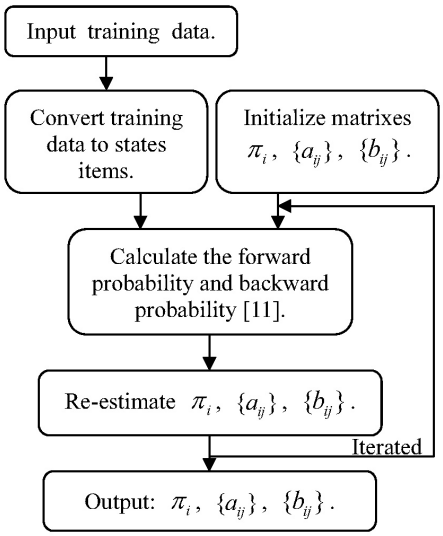


Fig. 6. Training through Expectation-Maximization.

iterations is 25 in our program. We used an open source with some improvements in order to facilitate application to our model (The general software implementation is available at http://people.cs.ubc.ca/~murphyk/software/HMM/hmm.html). By executing the training algorithm, the parameters of the HMM can be obtained.

In HMM training, the two databases are composed of a large number of sequences. Each sequence comprises 40 frames of the temporal history of the red color component of a pixel. We used 560,000 fire data items and 58,000 non-fire data items in the respective databases, where 200 items were randomly drawn as training data in our HMM training. Based on our experiments, it is determined that 200 items is a sufficient amount to optimize the parameters. Two models have been trained, and one uses the fire database, the other uses the non-fire database. π_i represents the initial state distribution, $\{a_{ij}\}$ represents state transition probability distribution, and $\{b_{ij}\}$ denotes the observation symbol probability distribution. The results of the training are as follows. In the fire state transition matrix $\{a_{ij}\}_{fire}$, 0.877465 is the possibility transferred from state F1 to F2 and 0.810050 denotes the possibility transferred from state NF to state NF; for comparison, in the non-fire model, 0.198413 and 0.318549 represent the possibility transferred from state F1 to F2 and NF to NF, respectively, which indicates that the fire and non-fire model have different tendencies. The matrixes $\{a_{ij}\}$ and $\{b_{ij}\}$ co-act on the calculation of *similarity*, as described in next section, which consequently affects the final decision.

Fire model:

$$\pi_{fire} = [0.3, 0.2, 0.5],$$

$$\{a_{ij}\}_{fire} = \begin{bmatrix} 0.000659 & 0.877465 & 0.121876 \\ 0.302857 & 0.292116 & 0.405027 \\ 0.107431 & 0.082519 & 0.810050 \end{bmatrix},$$

$$\{b_{ij}\}_{fire} = \begin{bmatrix} 0.952906 & 0.046640 & 0.000454 \\ 0.274963 & 0.461228 & 0.263809 \\ 0.199287 & 0.000001 & 0.800712 \end{bmatrix}.$$

Non-fire model:

$$\pi_{nonfire} = [0.3, 0.2, 0.5],$$

$$\{a_{ij}\}_{nonfire} = \begin{bmatrix} 0.491008 & 0.198413 & 0.310579 \\ 0.272130 & 0.727750 & 0.000120 \\ 0.281088 & 0.400363 & 0.318549 \end{bmatrix},$$

$$\{b_{ij}\}_{nonfire} = \begin{bmatrix} 0.733555 & 0.266039 & 0.000406 \\ 0.105474 & 0.000001 & 0.894525 \\ 0.280765 & 0.004744 & 0.714491 \end{bmatrix}.$$

4.2. HMM application

In HMM application, a sequence of color values is given. We wish to infer what this sequence means, that is, whether the sequence denotes fire or non-fire. A similarity result is obtained by comparing the given sequence of color values with the HMMs that are obtained in the training step. If the similarity between the sequence and the fire HMM is greater than that with the non-fire HMM, then the sequence denotes fire, and vice versa.

We define the similarity between the test sequence and the HMM model as (3):

$$\text{similarity} = \prod_{i=1}^{T} \sum_{j=1}^{N} \delta(i, j), \tag{3}$$

where $\delta(i, j)$ is obtained via a Viterbi-like algorithm that uses a Viterbi concept but only employs the introduction and recursion steps of the Viterbi algorithm (See Table 1).

In Table 1, N is the number of the states in the HMMs; our work uses three states, F1, F2, and NF. s(t) represents the given sequence, expressed as states. Note that the input sequence is expressed as color values. Hence, a converting algorithm, which is explained in Section 4.1 (HMM training), must be employed to

Table 1. Viterbi-like algorithm.

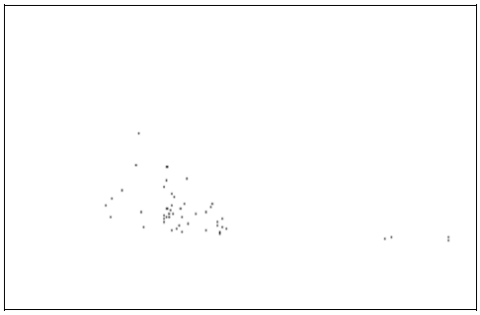
1) Introduction:

$$\delta(1, i) = \pi(i) \times b(i, s(1)) \qquad 1 \le i \le N$$

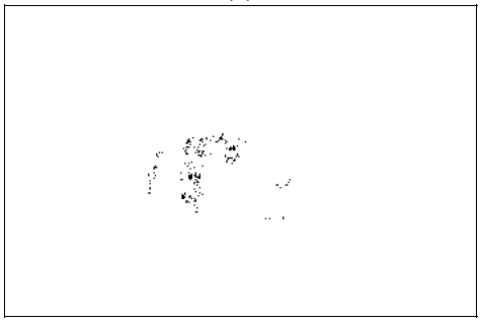
2) Recursion:

$$\delta(t, j) = \sum_{k=1}^{N} \delta(t-1, k) \times a(k, j) \times b(j, s(t))$$

$$2 \le t \le T \quad \text{and} \quad 1 \le j \le N$$



(a)



(b)

Fig. 7. Results after applying the HMM: (a)&(b) correspond to Fig. 4 (a)&(c), respectively.

change the color values to states in this step. T denotes the length of the given sequence.

$\delta(i, j)$ is the path probability, $\pi(i)$ is the initial state distribution probability, $a(i, j)$ is the state transition probability, and $b(i, j)$ is the observation probability. All of these probabilities are acquired in the HMM training stage.

After applying the HMM to the results described in Section 3, we obtain the results from the videos used in Section 3 (See Fig. 4) as shown in Fig. 7.

We can decide whether a pixel represents fire or not for every pixel after the HMM is applied to every candidate fire pixel generated via the steps described in Section 3. To summarize our algorithm explained in Section 3 and Section 4, we use Fig. 8 to show each step of the overall procedures of fire detection.

5. EXPERIMENTAL RESULTS

We implemented the proposed fire detection algorithm on an Intel(R) Core(TM)2 2.13GHz desktop computer with an image size of 360×240. The computational cost of the program is low, because most of the time-consuming steps such as HMM training are performed off-line, and therefore real-time operation can be achieved. We tested our program with multiple videos, some of which are presented in Fig. 9, including a comparison with [6] (The software of [6] was kindly provided by the authors of [6]). [6] also used a HMM model to detect fire, but the training method entails simply counting the state transition rather than the EM algorithm. In situations involving fire-color moving objects, false detection occurs. For example, the headlights are detected as fire in the second row of Fig. 9.

(a) (b) (c)

Fig. 8. Results for each step. The first row shows input video sequences, the second presents results after being processed by moving pixels detection, the third is the results of fire-color detection. The fourth is the clustering results and the last row is the final results after using HMM application. (a) A video shot of a road with head-lights of running cars as disturbance; (b) Fire with non-fire light with similar fire color; (c) Fire and a fire-like moving object.

(a) (b) (c)

Fig. 9. Comparative results for fire detection. (a) Original video sequences; (b) Results by using our method; (c) Results by using the method of [6]. The white boxes in Fig. 9(c) present the detected fires (with false detection in non-fire situation of 2nd row and fails to detect the small fires in 3rd row, respectively).

Table 2. Description and performance summary for some results.

Video	Number of frames showing fire/total frames	Number of frames for which fire was actually detected	Detection rate/ False alarm rate	Description
Video clip1	0/98	0	1.0/0.0	A smoking and moving person with a red bag
Video clip2	194/194	194	1.0/0.0	A moving person with a red bag behind fire
Video clip3	101/189	101	1.0/0.0	Running cars with car lamps on behind fire at night
Video clip4	122/122	122	1.0/0.0	Fire at night
Video clip5	0/134	0	1.0/0.0	Running cars with car lamps on at night
Video clip6	0/89	0	1.0/0.0	A person shaking a yellow book
Video clip7	96/96	96	1.0/0.0	Fire in the daytime
Video clip8	86/86	86	1.0/0.0	Fire at night with disturbance fire-color light

Meanwhile, in the third row of Fig. 9, there is a small fire in the image but the program of [6] fails to detect it. In contrast, our method successfully detected fire for various situations, even when there were many disturbances, such as moving fire-color objects, running cars with headlights on, lit cigarettes, etc.

We prepared 30 test video clips in total, with half of them showing fire. The proposed method successfully detected the 15 fire-video clips. All the situations in the test were accurately detected; that is, the false alarm rate was zero. Table 2 lists some of the tests.

6. CONCLUSIONS

In this paper, an approach to fire detection was provided. Decisions about fire pixels in a video sequence are made via HMMs, based on moving pixels detection, fire-color detection and pixels clustering. The reliability is tested via experiments involving multiple situations with many different disturbances, and we achieved promising results. The experimental results indicate that the rate of false alarms can be greatly reduced by HMMs and fire can be detected promptly and accurately. This

will be very useful for applications related to social security, commerce, the military, etc.

Although fire can be accurately detected by the proposed method, the fire pixels detected are not as many as the original images. The method of [6] also suffers from the same problem, that is, it only detects a small region as fire (shown in the first image of Fig. 9 (c)). In the future, we will continue our research on this aspect and utilize the proposed approach via coordination with hardware devices to monitor dangerous situations such as fire detection in tunnels.

REFERENCES

[1] I. Kopilovic, B. Vagvolgyi, and T. Sziranyi, "Application of panoramic annular lens fur motion analysis tasks: surveillance and smoke detection," *Proc. of IEEE Intemational Conference on Pattern Recognition*, pp. 714-717, 2000.

[2] G. Healey, D. Slater, T. Lin, B. Drda, and A. D. Goedeke, "A system for real-time fire detection," *Proc. of IEEE Computer Vision and Pattern Recognition Conference*, pp. 605-606. 1993.

[3] T. Chen, P. Wu, and Y. Chiou. "An early fire-detection method based on image processing," *Proc. of IEEE International on Image Processing*, pp. 1707-1710. 2004.

[4] B. U. Toreyin, Y. Dedeoglu, U. Gudukbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Lett*, vol. 27, pp. 49-58. 2006.

[5] T. C. Elik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Safety J*, doi:10.1016/j.firesaf.2008.05.005, 2008.

[6] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Flame detection in video using hidden markov models," *Proc. of ICIP '05*, pp. 1230-1233, 2005.

[7] T. H. Chen, C. L. Kao, and S. M. Chang, "An intelligent real-time fire-detection method based on video processing," *Proc. of the IEEE 37th Annual Inlernalional Carnahun Conference on Securiw Technology*, pp. 104-I 11, 2003.

[8] B. W. Albers and A. K. Agrawal, "Schlieren analysis of an oscillating gas-jet diffusion," *Combust. flame*, vol. 119, pp. 84-94. 1999.

[9] L. R. Welch, "Hidden Markov models and the baum-welch algorithm," *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, December 2003.

[10] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models," *Tech. Rep*., ICSI-TR-97-021, 1997.

[11] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of IEEE*, vol. 77, no. 2, pp. 257-285, Feb. 1989.

Zhu Teng received her B.S. degree in Automation from Central South University, China, in 2006. She is now a student of successive postgraduate and doctoral programs of study in Mechanical Engineering of Pusan National University, Korea. Her current research interests are visual surveillance, machine vision, robots and pattern recognition.

Jeong-Hyun Kim received his Master degree in Mechatronics Engineering from TongMyong University, Korea, in 2007. He is now a Ph.D. student at the department of Mechanical & Intelligent System Engineering of Pusan National University, Pusan, Korea. His current research interests are machine vision, visual surveillance and unmanned vehicle.

Dong-Joong Kang received his B.S. degree in Precision Engineering from Pusan National University, Pusan, Korea, in 1988 and an M.E. degree in Mechanical Engineering from KAIST (Korea Advanced Institute of Science and Technology), Seoul, Korea, in 1990. He also received his Ph.D. degree in Automation and Design Engineering at KAIST, in 1998. In 1997-1999, he was a research engineer at SAIT (Samsung Advanced Institute of Technology). During 2000.03~2006.02, he was an assistant professor at the department of mechatronics engineering in Tongmyong University. Since 2006, he has been an assistant professor at the school of mechanical engineering in Pusan National University. His current research interests are visual surveillance, intelligent vehicles/robotics, machine vision.