A Computer Vision-Based Method for Fire Detection in Color Videos

Xiaojun Qi[1] and Jessica Ebert[2]

[1]Department of Computer Science
Utah State University
Logan, UT 84322-4205, USA
Xiaojun.Qi@usu.edu

[2]Department of Computer Science
Connecticut College
New London, CT 06320-4196, USA
jrebe@conncoll.edu

ABSTRACT

This paper presents a computer vision-based approach for automatically detecting the presence of fire in video sequences. The algorithm not only uses the color and movement attributes of fire, but also analyzes the temporal variation of fire intensity, the spatial color variation of fire, and the tendency of fire to be grouped around a central point. A cumulative time derivative matrix is used to detect areas with a high frequency luminance flicker. The fire color of each frame is aggregated in a cumulative fire color matrix using a new color model which considers both pigmentation values of the RGB color and the saturation and the intensity properties in the HSV color space. A region merging algorithm is then applied to merge the nearby fire colored moving regions to eliminate the false positives. The spatial and temporal color variations are finally applied to detect fires. Our extensive experimental results demonstrate that the proposed system is effective in detecting all types of uncontrolled fire in various situations, lighting conditions, and environment. It also performs better than the peer system with higher true positives and true negatives and lower false positives and false negatives.

Keywords: video fire detection, cumulative time derivative matrix, cumulative fire color matrix, spatial color variation, temporal color variation*.*

2000 Mathematics Subject Classification: 68T45, 68W99.

1 INTRODUCTION

Smoke, fire, and flames are one of the leading hazards affecting everyday life around the world. Fires and burns are the second most common cause of death to children under 10 in the USA, second only to automobile crashes (Bowen, 2002). Smoke detectors have been widely used to an extent to raise alarms for saving people's life. However, faulty smoke detectors cause many of deaths resulting from smoke and fire. In addition, particle smoke detectors cannot be used reliably to detect fires in large spaces including auditoriums, warehouses, and outdoor spaces.

Other highly dependable fire detection systems use expensive infrared cameras. However, with the increased popularity and use of standard surveillance cameras, a fire detection system that is capable of utilizing these already installed cameras to reliably detect fires in large spaces could be invaluable. Computer vision algorithms for automatic video fire or smoke detection have been developed for applications in tunnels, aircraft hangars, ships, etc (Foo, 1996; Gottuk et al., 2006; Lloyd, 2000; Wieser and Brupbacher, 2001). They are dedicated to small, cluttered spaces by using various environment conditions that may occur in the specific applications to establish a range of flaming fire sources. But, none of these algorithms are robust and flexible to be applicable in standard surveillance cameras enabled automatic video fire detection. Therefore, a significant amount of research has been focused on developing reliable video fire detection systems in any large, open environment. Several related computer vision based fire detection systems are briefly reviewed here. Healey et al. (1993) propose to identify fire using only color clues. Based on the fact that fire flickers with a certain range of frequency, two independent systems (Liu and Ahuja, 2004; Marbach et al., 2006) add an analysis of temporal variations of fire to improve detection performance. In (Liu and Ahuja, 2004), fast Fourier transforms (FFT) of temporal object boundary pixels are computed to represent shapes of fire regions. In (Marbach et al., 2006), temporal variation of fire intensity is captured to select candidate fire regions and characteristic fire features are extracted to determine the presence of fire or non-fire patterns. Töreyin et al. apply hidden Markov models (HMM) (2005) and the wavelet transformation (2006) to extract spatial color variations to improve detection accuracy and reduce the false positives. These systems achieve promising results on their data sets. However, they cannot simultaneously handle the following two conditions:

- Complicated lighting conditions resulting from day and night, artificial lights, light reflection, or shadows.
- Complex scene with objects and/or people moving in velocities and sizes similar to fire.

The method presented in this paper addresses the above conditions and is flexible enough to be used in almost any condition in a multitude of environment. Specifically, our proposed method is reliable in all lighting conditions by incorporating both pigmentation values of the RGB color and the saturation and the intensity properties in the HSV color space. It is fairly apt at considering moving fire-colored objects as non-fire by using both spatial and temporal color variations in a novel way. It also analyzes the grouping of fire regions to further eliminate false alarms. In general, the proposed method works with a video stream by extracting a short clip, analyzing it for fire, and then extracting another clip for analysis. If a fire is detected within the clip, an alarm is issued. The remaining of the paper is organized as follows: Section 2 explains the proposed fire detection algorithm in detail. Section 3 demonstrates the effectiveness of our proposed system and shows the experimental results to compare our proposed system with a peer system (Töreyin et al., 2006). Section 4 concludes the paper and presents the future direction.

2 THE PROPOSED FIRE DETECTION ALGORITHM

The block diagram of our proposed fire detection algorithm is shown in Fig. 1. Specifically, it consists of the following six steps: (1) Flicker detection: Detect areas with a high frequency flicker using a cumulative time derivative matrix of luminance. (2) Color detection: Aggregate the fire color of each frame using a cumulative fire color matrix by considering both pigmentation values of the RGB color and the saturation and the intensity properties in the HSV color space. (3) Region merging: Merge the fire colored moving regions if they are closely located. (4) Spatial color variation: Measure spatial color variations on regions passing the first three steps to eliminate false-alarms raised by moving objects with a solid flame-color. (5) Temporal color variation: Measure temporal color variations on the same regions passing the first three steps to track the temporal color changes of each pixel in the fire candidate regions. (6) Final verdict: Decide the presence of fire in a video clip. Each step is explained in detail in the following subsections.

2.1 Flicker Detection

Two properties of fire (Hamins et al., 1992) are used to design the flicker detection algorithm to find the candidate fire regions. These two properties are:

- Fire tends to flicker with a frequency between one to ten Hz.
- Fire typically is the strongest source of light and therefore the luminance intensity near fire tends toward the maximal value.

In our system, we first use the time derivative of the luminance to track a moving object. The time derivative for a moving object is typically non-zero while the time derivative for a static environment is typically zero. We then use the cumulative time derivative (CTD) of the luminance to estimate the tendency for fire to periodically flicker around a region. This CTD incorporates both the cumulative strength and the luminance weight to accommodate the above two properties of fire, respectively. Specifically, the cumulative strength is proportional to the number of frames in the video clip. The luminance weight gives more weight to a bright pixel to improve the detection robustness. This CTD of the luminance is also normalized and processed to keep relatively large values. That is, we ensure that a fire border has a larger CTD response than a non-fire region.

The details of the flicker detection algorithm are:

1. Convert each frame F_i ($1 \le i \le N$) of a video clip from the RGB color space to the YUV color space, where N is the total number of frames in the video clip and $Y_i(x, y)$ represents the luminance color component at x-row and y-column of frame F_i.
2. For each position (x, y) of frame F_i ($2 \le i \le N$), perform the following operations:
   a. Compute the luminance time derivative by:

   $$D_i(x, y) = | Y_i(x, y) - Y_{i-1}(x, y) | \qquad (1)$$

   b. Compute the CTD by:

   $$CT_i(x, y) = \alpha CT_{i-1}(x, y) + (1 - \alpha) w_i(x, y) D_i(x, y) \qquad (2)$$

   where:

   - α represents the cumulative strength and is set to be (N-1)/N in our system. This setting always allows a larger cumulative strength to be set for a video with more frames. That is, the more frames in the video, the more accurate information the CTD carries, and the larger cumulative strength for each frames in the video.

- $w_i(x, y)$ is the luminance weight, which is proportional to the luminance that is brighter than the average luminance intensity δ of frame F_i. It is computed as:

$$w_i(x, y) = \begin{cases} Y_i(x, y) & \text{if } Y_i(x, y) \geq \delta \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

- $CT_1(x,y)$ is initially set to be all 0's.

c. Normalize $CT_i(x, y)$ to an 8-bit luminance range (i.e., $0 \leq CT_i(x, y) \leq 255$) and find its average *Ave*.

d. Reset the pixels in the normalized $CT_i(x, y)$, whose intensity values are below *Ave*, as 0's. This processed normalized $CT_i(x, y)$ is used by Equation (2) to compute the CTD in each iteration step.

3. Set all pixels in CT_N, whose values are less than the average of non-zero values in CT_N, to 0's to keep the strong moving regions.

Fig. 2 demonstrates two typical fire frames (e.g., one at night and one in the day) and the CTD of the two video clips containing these fire frames. It clearly shows that the borders of the flame regions have large CTD values. The static environment has small CTD values, as shown in dark in both CTD results. The center of night fire, a highly saturated luminance component, is dark due to less motion in center (i.e., the CTD at the center of fire is close to zero).

2.2 Color Detection

Other fire detection algorithms (Liu and Ahuja, 2004; Marbach et al. 2006; Töreyin et al., 2005; Töreyin et al., 2006; Chen et al., 2003; Phillips et al., 2000) exclusively use the pigmentation values of the RGB color to determine the potential fire regions. However, we notice that the saturation level and the intensity value in the HSV color space also play an important role in determining the potential fire regions in different lighting conditions and environment. Fig. 3 illustrates this observation using two different types of fire as shown in Fig. 2, i.e., fire at night and fire in the day. This figure shows that the average saturation levels of fire are relatively low if fire is the main source of light in the video. The average saturation levels of fire are relatively high if fire is not the main source of light in the video. The intensity values of fire tend to be relatively high in most cases regardless lighting conditions. Therefore, in our proposed system, we utilize both pigmentation values of the RGB color and the saturation and the intensity properties in the HSV color space to detect fire.

For each frame in a video clip, we perform the following operations to estimate fire colored regions. First, we consider fire potentially falling into a red to yellow range using the relationships among red, green, and blue components (Chen et al., 2003) shown in Fig. 4. Second, we use the saturation values of potential fire derived from the first step to measure the membership of fire (i.e., the possibility of being fire) based on the average saturation levels of potential fire. Third, we assign a larger membership to a brighter spot in the value (intensity) component of the HSV color space. Fourth, we estimate the membership of fire by multiplying the fire membership in saturation and intensity components. This additional consideration of saturation and intensity components effectively identifies all fire colored regions in different lighting conditions and environment by assigning pixels of strong intensity and saturation values with a high possibility of being a fire. Finally, we compute a cumulative fire color (CFC), which is akin to the CTD, to estimate the fire colored regions of the video clip.

The details of the color detection algorithm are:

1. Convert each frame F_i ($1 \le i \le N$) of a video clip from the RGB color space to the HSV color space, where N is the total number of frames in the video clip and $R_i(x, y)$, $G_i(x, y)$, $B_i(x, y)$, $S_i(x, y)$, and $V_i(x, y)$ represent the red, green, blue, saturation, and value components at x-row and y-column of frame F_i in RGB and HSV color spaces, respectively.
2. For each position (x, y) of frame F_i ($2 \le i \le N$), perform the following operations:
    a. Create a fire color mask, $FCM_i(x, y)$ by:
    $$FCM_i(x,y) = \begin{cases} 1 & \text{if } R_i(x,y) > \eta \,\&\, R_i(x,y) > G_i(x,y) \,\&\, G_i(x,y) > B_i(x,y) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$
    where η is experimentally decided to be 180 since the fire color tends to fall into a red to yellow range and the value of 180 in the red component represents more yellowish color.
    b. Create a new saturation matrix by:
    $$S_i^{'}(x,y) = \begin{cases} S_i(x,y) & \text{if } FCM_i(x,y) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$
    c. Compute the saturation level *SLevel* by averaging all the non-zero elements in $S_i^{'}$.
    d. Compute the fire membership in the saturation component by:
    $$FS_i(x,y) = \begin{cases} S_i(x,y) & \text{if } SLevel \ge 0.5 \\ 1 - S_i(x,y) & \text{if } SLevel < 0.5 \text{ and } R_i(x,y) > 180 \\ S_i(x,y) & \text{if } SLevel < 0.5 \text{ and } R_i(x,y) \le 180 \end{cases} \tag{6}$$
    e. Compute the fire membership in the value component based on the average intensity value *IntensityAve* of V_i:
    $$FV_i(x,y) = \begin{cases} V_i(x,y) & if\ V_i(x,y) > \max(0.51, IntensityAve) \\ 0 & otherwise \end{cases} \tag{7}$$
    f. Estimate the fire membership by:
    $$FCSV_i(x,y) = FS_i(x,y) \times FV_i(x,y) \tag{8}$$
    g. Set all the pixels in $FCSV_i(x, y)$ whose values are less than the average of non-zero values in $FCSV_i$ as 0's.
    h. Compute the cumulative fire color by:
    $$CC_i(x,y) = \alpha CC_{i-1}(x,y) + (1-\alpha) FCSV_i(x,y) \tag{9}$$
    where α represents the cumulative strength, which has the same meaning as the α used in the flicker detection algorithm and is set to be (N-1)/N in our system; and $CC_1(x, y)$ is initially set to be 0's.
3. Set all pixels in CC_N, whose values are less than the average of non-zero values in CC_N, to 0's to keep strong fire colored regions.

The highly likely fire regions as of this point by combining flicker and color detection are defined as:
$$Fire(x,y) = CT_N(x,y) \times CC_N(x,y) \tag{10}$$
Similarly, we maintain strong fire regions by keeping the pixels whose values are larger than the average of non-zero values in *Fire*(x,y). Fig. 5 shows the intermediate results by applying the color detection

algorithm on two video clips shown in Fig. 2. It clearly shows that the regions that satisfy both fire flicker and fire color properties are located as fire candidate regions.

2.3 Region Merging

We observe that strong fire regions in *Fire* are generally clustered together. As a result, we apply a series of morphological operations to merge nearby fire regions and eliminate non-fire outliers. The choice of the structuring elements is based on the spatial distance of nearby strong fire regions in a set of training images. Fig. 6(a) shows the results by applying the region merging algorithm on two video clips as shown in Fig. 2. This figure clearly shows that the strong fire regions are merged and other noises which lead to the small merged regions are eliminated. The details of the region merging algorithm are:

1. Apply the dilation operation with the square structuring element of 7×7 to *Fire* to enlarge each fire region. Store this dilated result in *DilatedFire*.
2. Apply the closing operation with the disk structuring element of 8×8 to *DilatedFire* to fill in the gaps between two enlarged fire regions. Store this closed result in *CandidateFire*.
3. Trace region boundaries in *CandidateFire* to evaluate the size of each candidate fire region. Discard any relatively small candidate fire regions as non-fire outliers. Store this result in *CleanCandidateFire*.
4. Apply the erosion operation with the disk structuring element of 2×2 to *CleanCandidateFire* to restore the cleaned candidate fire regions. Store this result in *RestoredCleanCandidateFire.*

2.4 Spatial Color Variation

We observe that fire does not remain a steady color and objects with a solid flame-color remain a steady color. This observation is demonstrated in Fig. 7. Consequently, we propose the spatial color variation algorithm to eliminate false-alarms raised by moving objects with a solid flame-color, such as fire colored shirts wore by dancing persons in a consistent lighting.

In order to speed-up the processing time, we exclusively measure the spatial color variations within the restored cleaned candidate fire regions in the first, middle, and last frames of a video clip. The candidate fire regions, i.e., *RestoredCleanCandidateFire*, are estimated by applying the first three algorithms (e.g., flicker detection, color detection, and region merging algorithms) on the original video clip. For each of the first, middle, and last frames of a video clip, the details of the spatial color variation algorithm are:

1. Respectively apply the range filter to each candidate region in *RestoredCleanCandidateFire* in red, green, and blue color components to measure the spatial variations. This range filter computes the spatial change of each candidate fire pixel as the difference between the maximum value and the minimum value within its 3×3 neighborhood.
2. Apply the normalization technique to scale the spatial color changes in red, green, and blue color components to [0, 1]. Here, we divide each of the spatial color changes in the three components by the maximum spatial color changes in all of the three components.

3. Keep the candidate fire pixels in each color component, which show larger spatial changes than the average spatial changes of three color components, as potential fire pixels. Discard the remaining candidate fire pixels as non-fire pixels.
4. Set the potential fire pixels in each color component, which show larger spatial changes than the average spatial changes in the green color component, as highly possible fire pixels. This setting is mainly based on the fact that the red component of fire has a limited range of changes and the green component of fire has a wide range of changes.

Let $R_{first}(x, y)$, $G_{first}(x, y)$, and $B_{first}(x, y)$ respectively denote the red, green, and blue color components of the first frame of a video clip with the highly possible fire pixels as 1's and non-fire pixels as 0's. Similarly, $R_{middle}(x, y)$, $G_{middle}(\mathrm{x, y})$, and $B_{middle}(\mathrm{x, y})$ respectively denote the red, green, and blue color components of the middle frame of a video clip with the highly possible fire pixels as 1's and non-fire pixels as 0's; $R_{last}(x, y)$, $G_{last}(\mathrm{x, y})$, and $B_{last}(\mathrm{x, y})$ respectively denote the red, green, and blue color components of the last frame of a video clip with the highly possible fire pixels as 1's and non-fire pixels as 0's. Apply the intersection operation on the highly possible fire regions of three frames in each color component. That is, $R_{all} = R_{first}(x, y)$ && $R_{middle}(x, y)$ && $R_{last}(x, y)$, $G_{all} = G_{first}(\mathrm{x, y})$ && $G_{middle}(\mathrm{x, y})$ && $G_{last}(\mathrm{x, y})$, and $B_{all} = B_{first}(\mathrm{x, y})$ && $B_{middle}(\mathrm{x, y})$ && $B_{last}(\mathrm{x, y})$ where && is the intersection operation. Apply the union operation, $R_{all} \,||B_{all} \,||\, G_{all}$, to get the restored clean candidate fire regions with large spatial color variations. This result is passed down to the next stage, temporal color variation, for further processing. Fig. 6(b) shows the results by applying the spatial color variation algorithm on two video clips as shown in Fig. 2. This figure clearly shows that the restored clean candidate fire regions with large spatial color variations are located.

2.5 Temporal Color Variation

We propose the temporal color variation algorithm to track the temporal color changes of each pixel in the fire candidate regions obtained up to this stage. The details of the temporal color variation algorithm are:

1. Compute the number of changes, *fireChanges*, that occur between fire colored pixels at the same location of the most nearby frames in the combined RGB channel.
2. Compute the number of changes, *Changes*, that occur between two pixels at the same location of adjacent frames in the combined RGB channel.
3. Use *Changes* as a stretching factor to multiply with *fireChanges* to augment the difference of temporal color variations. Store this augmented temporal color variations as *fireTemporalVariations*.
4. Threshold *fireTemporalVariations* to keep fire regions with the relatively large temporal color variations. The threshold is set to be the maximum of the value of $N/15$ and the average of *fireTemporalVariations*, where N is the total number of frames in a video clip.

Here, the average of *fireTemporalVariations* represents the overall average of the temporal color changes in the entire video. Since the temporal color changes tend to be larger in a video clip with more frames and in a video clip containing fire, we empirically decide the value of $N/15$ to measure the appropriate level of temporal color changes in a video without considering any fire information. This value is proportionally increased with the number of clips in a video. As a result, the maximum of $N/15$ and *fireTemporalVariations* is a better threshold to keep fire regions with the relatively large temporal color variations. Fig. 6(c) shows the results by applying the temporal color variation algorithm on two video clips as shown in Fig. 2. This figure clearly shows that the restored clean candidate fire regions with large spatial and temporal color variations are located.

2.6 Final Verdict

The final fire is determined by the ratio of the number of fire candidate pixels obtained up to this stage to the number of fire candidate pixels in the strong fire regions obtained by flick and color detection algorithms. If the ratio is larger than a reasonable number decided by the deployed application field of the fire detection system, fire is considered present and an alarm is issued. In the current system, fire is considered present as long as there is at least one pixel in the final fire candidate region up to this stage. Fig. 6(d) shows the final fire detection results of the two video clips shown in Fig. 2. Here, we mark the detected fire regions by red circles. These regions satisfy the fire flicker properties, the fire color properties, the fire group property, and the fire spatial variation and temporal variation properties.

3 EXPERIMENTAL RESULTS

The proposed method is effective for a large number of conditions and was tested on a large and varied database. Most clips were downloaded from the web (e.g., google video), some clips were provided by the Salt Lake City Fire Department, and some clips are taken by the authors. All of the clips within the database have a minimum resolution of 320×240 and the minimum viable frame rate is 25 frames per second. The database consists of 180 videos with a broad range of distinct situations and content. In total, the database has 60 fire videos ranging from campfire and fire in a grill in the day and night to burning candles, blue sofa, tan couches, cardboards, and houses at different lighting conditions, etc. The database also has 120 non-fire videos. Out of these non-fire videos, 60 videos have some characteristics that fire normally exhibits. For example, these non-fire videos range from a fountain with yellow lighting and arching water, a running cheetah, a spotted bird walking, dead trees in the wind at sunset, a man doing disco in a red shirt in the spotlight, a fire truck with its lights on driving towards the camera, an American flag and a firemen flag waving in the wind, several people wearing red or pink dancing the flamenco, cars driving on an icy road with their taillights on, a girl spinning on a unicycle with a red tunic, to a man talking in front of a black background, etc. They were deliberately selected to test the robustness of our fire detection system in terms of false positives and false negatives. The remaining 60 videos out of the non-fire videos do not have any characteristics that fire normally exhibits.

Table 1 compares the fire detection results by applying our proposed system and the system proposed in (Töreyin et al., 2006) on 16 sample videos (eight fire videos and eight non-fire videos) of our database. It clearly shows that our system is more reliable than the other system. Specifically, our system detects the candle (the last video in the fire videos) as fire while the other system fails to detect it as fire. Our system also correctly recognizes three non-fire videos (persons walking on pink floors, a boy dancing with a neon orange blanket and red clothing, and fire truck with lights on), which have some of the characteristics of fire, as non-fire. However, the other system is fooled to consider these three videos as having fire inside of them. Both systems mistakenly consider the last video in the non-fire videos to have fire inside of it. This video is of a yellow lit fountain at night with several arches of water and a bit of yellow lit mist. The fountain exhibits all the characteristics of fire and could only be conceivably eliminated through a thorough shape analysis.

Table 2 summarizes the fire detection results by applying our proposed system and the system proposed in (Töreyin et al., 2006) on the 180-video database. It clearly shows that our system is able to correctly detect all 60 kinds of fire while the other system fails to detect three kinds of fire. Among the 60 non-fire videos, which have the characteristics of fire, our system can correctly identify 50 videos as non-fires while the other system can only correctly identify 30 of them. Among the remaining 60 non-fire videos, which do not have the characteristics of fire, both our system and the other system can correctly identify them as non-fire. Fig. 8 shows a few sample frames in video clips where our system induces false positives.

4 CONCLUSIONS

This paper presents a novel system for automatically detecting the presence of fire in color video sequences. The algorithm not only uses the color and movement attributes of fire, but also analyzes the temporal variation of fire intensity, the spatial color variation of fire, and the tendency of fire to be grouped around a central point. Our system also uses an enhanced fire color model, which incorporates both the pigmentation values of the RGB color and the saturation and the intensity properties in the HSV color space, to identify all fire candidate regions in different lighting conditions and environment. Extensive experimental results on 180 videos (60 fire videos and 120 non-fire videos with half of them containing fire characteristics) show that our proposed system is consistent and reliable at detecting all forms of fire and performs better than the peer system with higher true positives and true negatives and lower false positives and false negatives.

The entire system has been implemented using Matlab 7.6 on a Pentium IV 3.16 Ghz PC running Windows XP operating system. In average, 13.02 seconds are needed to detect fire in a video clip of around 100 frames. This can be easily reduced to at least one-twentieth if the entire system is implemented in C language. We will further improve the run time efficiency in Matlab and convert the algorithms in C to make our proposed system practical as a preventative fire detection system. We will also study the shape of fire to further eliminate the false positives.

References

Bowen, J., 2002, *The Do's and Don'ts of Teaching Home Fire Safety*. National Safety Council.

Chen, T. H., Kao, C. L., Chang, S. M., 2003, *An Intelligent Real-Time Fire-Detection Method Based on Video Processing*, Proc. of IEEE, 104-111.

Foo, S. Y., 1996, *A Rule-Based Machine Vision System for Fire Detection in Aircraft Dry Bays and Engine Components*, Knowledge-Based System, 9, 531-541.

Gottuk, D. T., Lynch, J. A., Rose-Pehrsson, S. L., Owrutsky, J. C., Williams F. W., 2006, *Video Image Fire Detection for Shipboard Use*, Fire Safety Journal, 41, 4, 321-326.

Hamins, A., Yang, J. C., Kashiwagi, T., 1992, *An Experimental Investigation of the Pulsation Frequency of Flames*, Proc. of the 24th International Symposium on Combustion, 1695-1702.

Healey, G., Slater, D., Lin, T., Drda, B., and Goedeke, D., 1993, *A System for Real-Time Fire Detection*, Computer Vision and Pattern Recognition, 605-606.

Liu, C., Ahuja, N., 2004, *Vision Based Fire Detection*, Proc. of the 17th International Conference on Pattern Recognition, 134-137.

Lloyd, D., 2000, *Video Smoke Detection (VSD-8)*, Fire Safety Engineering, pp. 26.

Marbach, G., Loepfe, M., Brupbacher, T., 2006, *An Image Processing Technique for Fire Detection in Video Images*, Fire Safety Journal, 4, 285-289.

Phillips, W., Shah, M., Lobo N. d. V., 2000, *Flame Recognition in Video*, Proc. of Fifth IEEE Workshop on Applications of Computer Vision, 224-229.

Töreyin, B. U., Dedeoğlu, Y., Çetin, A. E., 2005, *Computer Vision Based Method for Real-Time Fire and Flame Detection*. Bilkent University.

Töreyin, B. U., Dedeoğlu, Y., Gudukbay, U., Çetin, A. E., 2006, *Computer Vision Based Method for Real-Time Fire and Flame Detection*, Pattern Recognition Letters, 27, 49-58.

Wieser, D., Brupbacher, T., 2001, *Smoke Detection in Tunnels Using Video Images*, Proc. of the 12th Int. Conf. on Automatic Fire Detection, 79-90.

List of figures and tables:

Fig. 1: Block diagram of the proposed fire detection algorithm.

Fig. 2: Results of the flicker detection algorithm. (a) A typical night fire frame in a video clip of 93 frames and the CTD after the flicker detection step. (b) A typical daytime fire frame in a video clip of 142 frames and the CTD after the flicker detection step.

Fig. 3: Saturation levels and intensity values for two types of fire. (a) Left: Fire at night; Middle: Its saturation values in S component of the HSV color space; Right: Its intensity values in V component of the HSV color space. (b) Left: Fire in the day; Middle: Its saturation values in S component of the HSV color space; Right: Its intensity values in V component of the HSV color space.

Fig. 4: Fire colors in the range of red to yellow.

Fig. 5: Intermediate results of the color detection algorithm on two video clips (the upper row is for the night fire video and the lower row is for the daytime fire video). (a) Typical fire frames of two video clips. (b)The CFC CC_N of the video clip obtained after applying the first two steps. (c) The CTD CT_N of the video clip obtained after the flicker detection algorithm. (d) The strong fire regions (marked as white) obtained by keeping large pixel values in highly likely fire regions obtained by Equation (10).

Fig. 6: Results of the next four algorithms, namely, region merging, spatial color variation, temporal color variation, and final verdict (the upper row is for the night fire video and the lower row is for the daytime fire video). (a) The restored clean candidate fire regions, e.g., *RestoredCleanCandidateFire,* obtained after the region merging algorithm. (b) The restored clean candidate fire regions with large spatial color variations, e.g., $R_{all} || B_{all} || G_{all}$, obtained after the spatial color variation algorithm. (c) The restored clean candidate fire regions with large spatial and temporal color variations, e.g., thresholded *fireTemporalVariations*, obtained after the temporal color variation algorithm. (d) Final fire regions marked by red circles after the final verdict operation.

Fig. 7: Spatial variations in fire and a typical fire colored object and their blow-up views.

Fig. 8: Three sample frames resulting in the false positives. The red rectangles indicate our misclassified fire regions.

Table 1: Comparison of the fire detection results of our proposed system and the other system (Töreyin et al., 2006) on 16 videos.

Table 2: Comparison of the fire detection results of our proposed system and the other system (Töreyin et al., 2006) on the entire database (**our results**, the results of the other system)

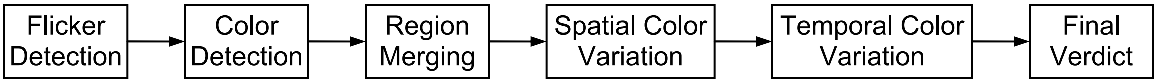
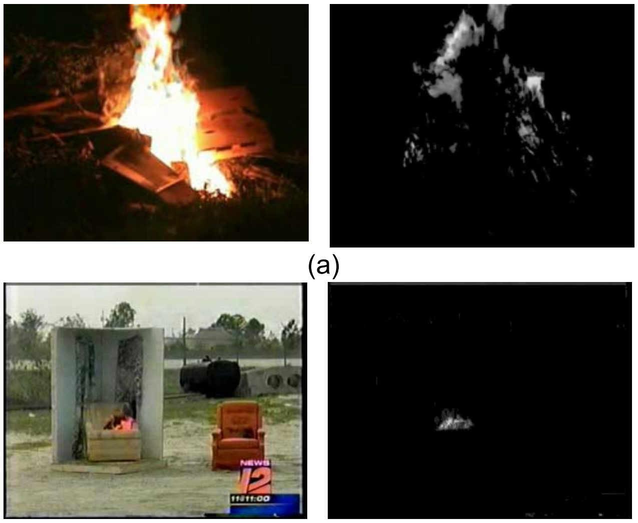
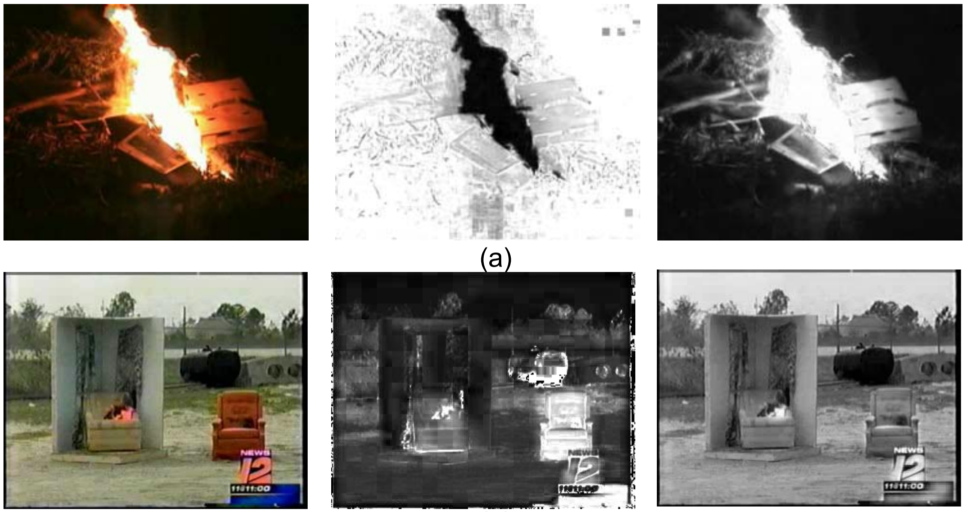


Fig. 1: Block diagram of the proposed fire detection algorithm.


(a)

(b)

Fig. 2: Results of the flicker detection algorithm. (a) A typical night fire frame in a video clip of 93 frames and the CTD after the flicker detection step. (b) A typical daytime fire frame in a video clip of 142 frames and the CTD after the flicker detection step.



(a)

(b)

Fig. 3: Saturation levels and intensity values for two types of fire. (a) Left: Fire at night; Middle: Its saturation values in S component of the HSV color space; Right: Its intensity values in V component of the HSV color space. (b) Left: Fire in the day; Middle: Its saturation values in S component of the HSV color space; Right: Its intensity values in V component of the HSV color space.

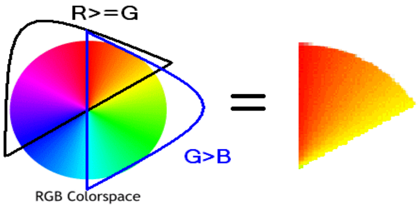


Fig. 4: Fire colors in the range of red to yellow.

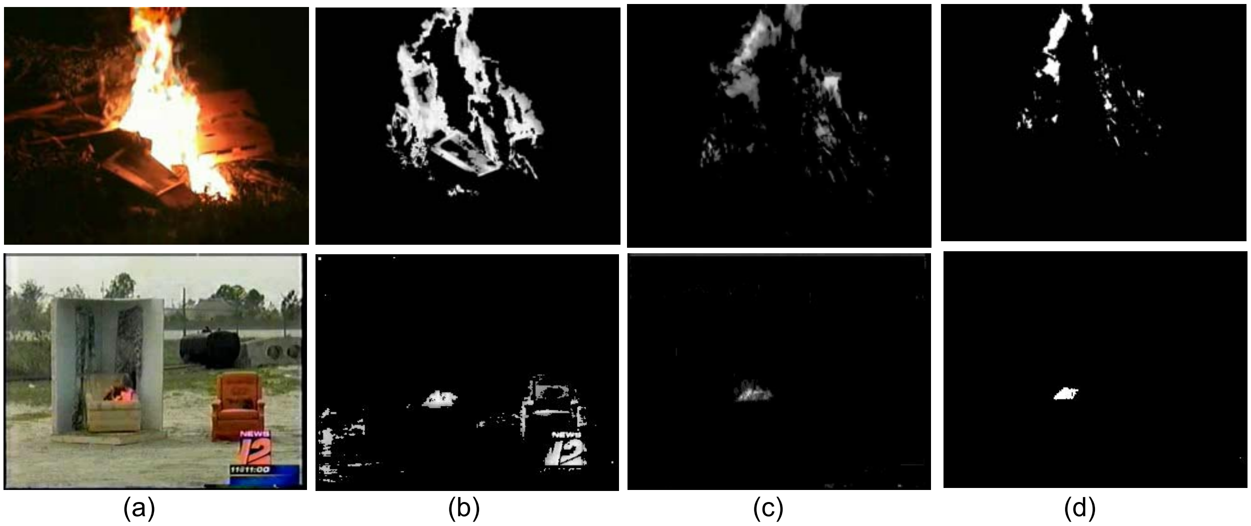


Fig. 5: Intermediate results of the color detection algorithm on two video clips (the upper row is for the night fire video and the lower row is for the daytime fire video). (a) Typical fire frames of two video clips. (b)The CFC CC_N of the video clip obtained after applying the first two steps. (c) The CTD CT_N of the video clip obtained after the flicker detection algorithm. (d) The strong fire regions (marked as white) obtained by keeping large pixel values in highly likely fire regions obtained by Equation (10).

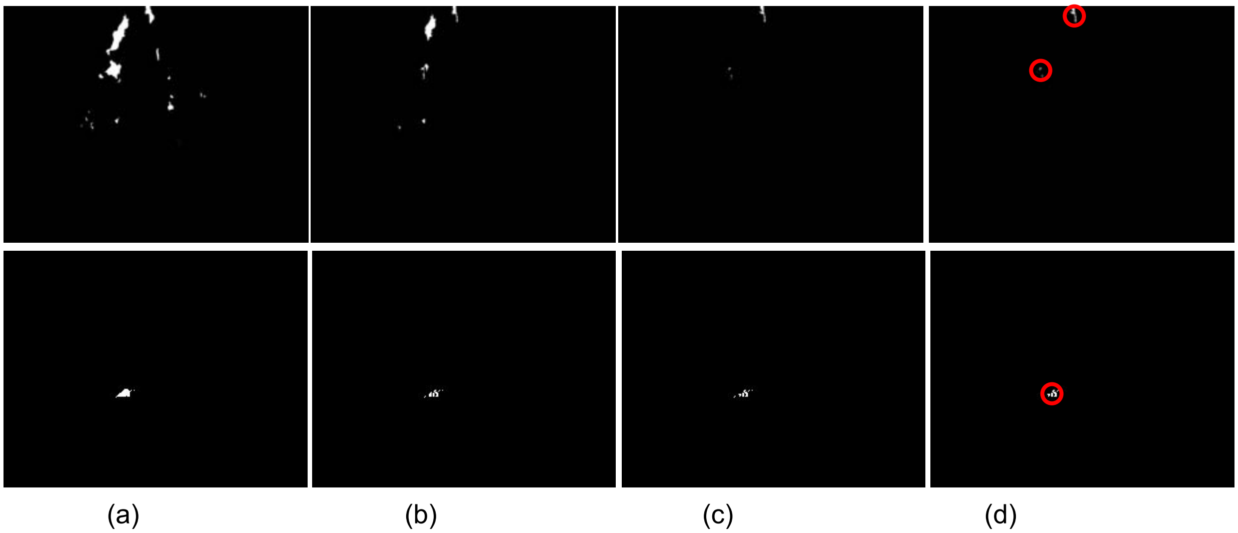


Fig. 6: Results of the next four algorithms, namely, region merging, spatial color variation, temporal color variation, and final verdict (the upper row is for the night fire video and the lower row is for the daytime fire video). (a) The restored clean candidate fire regions, e.g., *RestoredCleanCandidateFire,* obtained after the region merging algorithm. (b) The restored clean candidate fire regions with large spatial color variations, e.g., R_{all} ||B_{all} || G_{all}, obtained after the spatial color variation algorithm. (c) The restored clean candidate fire regions with large spatial and temporal color variations, e.g., thresholded *fireTemporalVariations*, obtained after the temporal color variation algorithm. (d) Final fire regions marked by red circles after the final verdict operation.

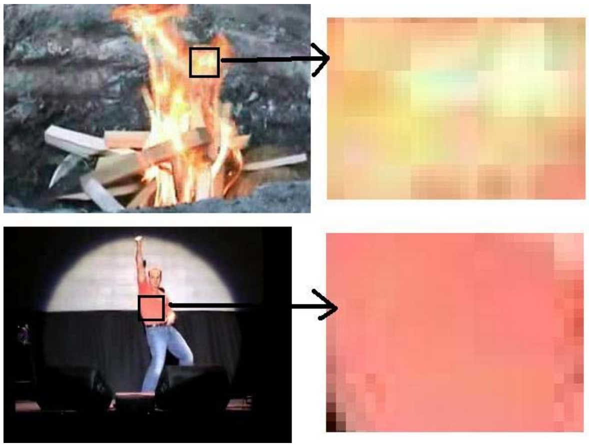


Fig. 7: Spatial variations in fire and a typical fire colored object and their blow-up views.

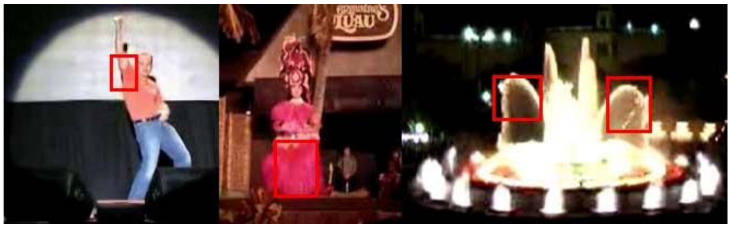


Fig. 8: Three sample frames resulting in the false positives. The red rectangles indicate our misclassified fire regions.

Table 1: Comparison of the fire detection results of our proposed system and the other system (Töreyin et al., 2006) on 16 videos.

	Number of Frames	***Frames with Fire***	***Fire Detected (ours, other)***	***Description***
Fire Videos	618	401	(Yes, Yes)	Tan couch on fire in a clear day
	49	49	(Yes, Yes)	People holding candles at night
	98	98	(Yes, Yes)	Fire in a grill in a cloudy day
	108	108	(Yes, Yes)	Cardboard fire at late evening
	93	93	(Yes, Yes)	Campfire at dark night
	142	142	(Yes, Yes)	Armchair burning in a clear day
	104	104	(Yes, Yes)	Three dog houses burning
	98	98	(Yes, No)	Candle with both ends burning that is balanced and teeters
Non-Fire Videos	37	0	(No, No)	White birds taking off
	48	0	(No, No)	Street intersection in a raining day
	32	0	(No, No)	American flag waving in the wind in a clear day
	48	0	(No, No)	A swinging light bulb at night
	14	0	(No, Yes)	Fire truck with lights on driving towards the camera
	199	0	(No, Yes)	A mall with people walking on pink floors with light reflection
	34	0	(No, Yes)	A boy dancing in front of a window with a neon orange blanket and red clothing.
	41	0	(Yes, Yes)	Yellow lit fountain and arching water at night

Table 2: Comparison of the fire detection results of our proposed system and the other system (Töreyin et al., 2006) on the entire database (**our results**, the results of the other system)

		Actual Condition	
		Fire	Non-Fire
Test Results	Fire	(**60**, 57)	(**10**, 30)
	Non-Fire	(**0**, 3)	(**110**, 90)