

# QuickBlaze: Early Fire Detection Using a Combined Video Processing Approach

*Waqar S. Qureshi, Mongkol Ekpanyapong\*, Matthew N. Dailey and Suchet Rinsurongkawong, Asian Institute of Technology, Klong Luang, Pathumthani 12120, Thailand*

*Anton Malenichev and Olga Krasotkina, Tula State University Cybernetics Faculty, Tula, Russia*

**Received:** 11 October 2014/**Accepted:** 16 April 2015

**Abstract.** Optical fire sensors, sometimes called “volumetric” sensors, are complementary to conventional point sensors such as smoke and heat detectors in providing people with early warnings of fire incidents. Cameras combined with image processing software hold the promise of detecting fire incidents more quickly than point sensors and can also provide size, growth, and direction information more readily than their conventional counterparts. In this paper, we present QuickBlaze, a flame and smoke detection system based on vision sensors aimed at early detection of fire incidents for open or closed indoor and outdoor environments. We use simple image and video processing techniques to compute motion and color cues, enabling segmentation of flame and smoke candidates from the background in real time. We begin with color balancing, then separate smoke and flame detection streams operate on the image. Both streams identify candidate regions based on color information then perform morphological image processing on the candidates. The smoke detection stream then filters candidate regions based on turbulence flow rate analysis, and the flame detection stream filters based on growth and flow rate information. QuickBlaze does not require any offline training, although manual adjustment of parameters during a calibration phase is required to cater to the particular camera’s depth of view and the surrounding environment. In an extensive empirical evaluation benchmarking QuickBlaze against commercial fire detection software, we find that it has a better response time, is 2.66 times faster, and better localizes fire incidents. Detection of fire using our real-time video processing approach early on in the burning process holds the potential to decrease the length of the critical period from combustion to human response in the event of a fire.

**Keywords:** Optical flow, Machine vision, Fire detection, Turbulence analysis

## 1. Introduction

Fire poses a major risk to human safety, health, and property. Traditional “point sensor” fire detection technologies based on particle sampling, temperature sampling, and smoke analysis have slow response time, usually measured in minutes,

---

\* Correspondence should be addressed to: Mongkol Ekpanyapong, E-mail: mongkol@ait.asia



and have little applicability in outdoor environments. To avoid false alarms, smoke and heat detectors only trigger once a sufficient amount of smoke particles flow into the device or until the temperature has increased substantially. However, time is a major factor in minimizing the damage caused by a fire. Decreasing the response time can greatly increase our chances to extinguish a fire and reduce the damaged caused by the incident.

The slow response time of point sensors has led researchers to consider, as an alternative, volumetric sensor based computing for automatic fire detection systems. Detecting smoke and flame in images or video frames grabbed from a vision sensor is one such technique [1]. Vision sensor based methods hold the promise of decreasing the response time, increasing the probability of detection, and providing coverage of large areas including open areas. Vision sensors can provide information about the direction, size, and growth of fire and smoke [2]. Existing CCTV surveillance systems in factories and public sites can in principle be upgraded at low cost to provide early warnings of a fire using video processing and machine vision techniques to detect flame and smoke.

Çetin et al. [3] provide a recent review of short-range video-based fire detection systems and summarize the existing approaches in terms of the underlying techniques they use. They note that the current vision sensor based approaches focus on both flame and smoke detection, while earlier work only investigated flame detection. From the early research of the late 1990's, significant improvement has been made, leading to commercial products such as VFDS [4], VisiFire [5], and SIGNIFIRE [6], which are now widespread in buildings and outdoor wild-fire warning systems. However, these fire detection systems still require human intelligence to distinguish between real fires and false alarms.

In general, vision sensor based fire detection systems consist of three components: motion detection, candidate object segmentation, and flame or smoke blob detection. As the camera is normally assumed to be fixed in such applications, most of the attempts use some variation on background modeling or image subtraction methods [7] to detect initial motion. Motion detection is followed by candidate object segmentation using either color information [8, 9], texture cues [10], pixel variability measures [11], or by optical flow field distribution [12]. Flame and smoke blobs may be detected either by modeling flame or smoke in mathematical terms [13–15] or using image separation approaches [16, 17]. After motion detection, morphological image processing techniques are often used as a preprocessing step for region detection. Millan-Garcia [18] present an early fire alarm system using IP cameras focusing only on smoke detection. The motion and smoke detection is carried out in Discrete Cosine Transform (DCT) domain instead of the time domain. Yu Cui et al. [10] and Truong et al. [19] also focus on the detection of smoke for early fire alarms and use color and texture cues respectively for smoke segmentation and machine learning classifiers for the final region classification decision.

Classifiers able to identify regions of interest can be used at a number of stages in the process. A classifier may involve simple image processing based segmentation criteria [20, 21] or more complex approaches involving machine learning [22, 23], Bayesian classification [24], classifiers using a mixture of Gaussian (MoG) model [25], support vector machines [26], Markov random fields [27–30], or neural

networks [31, 32]. Classifiers may incorporate features characterizing color, texture of flame or smoke, and spatial or temporal frequency analysis.

Video evidence of a fire can be characterized by either flame or smoke regions. In either case, the system's ability to detect fire and smoke will depend on the specific scene depth and camera field of view. But flame and smoke have different physical properties and dynamic behavior. A system that can detect both smoke and flame regions would have a greater probability of detecting fire earlier than one that only detects one or the other. Yu et al. [33] present a real-time flame and smoke detection algorithm. Initial candidate moving pixels are computed using differential background subtraction and then flame and smoke color models are used to obtain a decision rule to segment out the flame and smoke regions. Foreground images for flame and smoke are accumulated. Candidate flame regions are declared if a block (an  $8 \times 8$  group of pixels) of accumulated foreground image value is greater than a threshold. Smoke features are extracted using optical flow of blocks of pixels of the accumulated foreground image. Smoke candidate regions are classified using a neural network classifier. The authors report a processing time of 25 frames per second for a video at  $320 \times 240$  resolution, but they do not mention the specification of the processor, the response time (the time from the point the fire started to the time the fire is detected), or the false alarm rate for their method.

Some authors have reported empirical evaluation results more thoroughly. Wang et al. [34] present a fire flame detection algorithm that extracts initial flame candidates using motion, texture, and color cues. They then use flame area variation to eliminate false detections. Finally, the authors compare their algorithm's performance with that of Chen et al. [35] in terms of response time. The authors claim to process a  $320 \times 240$  video at 24 frames per second. However, they do not detect smoke, which results in a slow response time for incidents where smoke is the better early indicator of fire.

In this paper we focus on the use of an RGB camera to detect flame and smoke as an early indicator of fire incidents occurring within a range of one meter to 20 meters from the camera. Our system, called QuickBlaze, is an extension of Rinsurongkawong et al.'s [20] and Malenichev et al.'s [21] methods. We combined the techniques into two parallel streams to achieve better response time for early fire detection. Flame is detected by a combination of growth rate analysis and Lucas-Kanade pyramidal optical flow analysis [36] on candidate regions segmented out by background subtraction and a RGB color model. To detect smoke, we use turbulence analysis on candidate regions detected after motion and color cues.

We present a comprehensive empirical evaluation of QuickBlaze in terms of response time, processing speed in frames per second (FPS), and frame error rate (FER, for videos that do not contain fire) on a wide variety of videos including a new set of videos of fire incidents and non-fire events as well as a set of videos available online. For comparison, we choose the commercial-grade software Visi-Fire [5], which is a real-time vision-based flame and smoke detection application available for commercial use. It is based on a series of academic research contributions [3, 27, 28, 37, 38] in the area of fire incident detection. Out of 20 test video sequences that contain flame or smoke regions, QuickBlaze's time is better

for 18 of the sequences. QuickBlaze is able to detect fire incidents in cases (three video sequences) where VisiFire failed to detect the fire incident. The algorithm runs 2.66 times faster than VisiFire on the same hardware. The name QuickBlaze is not a trademark; it is simply used for ease of reading the manuscript. All the videos used for testing in the empirical evaluation described in this paper are available online at the AIT Computer Vision Wiki [39].

In summary, the detailed experimental evaluation reported upon in this paper shows that QuickBlaze is faster and has a lower response time to fire incidents than state of the art commercial software. More generally, simultaneous detection of flame and smoke in video holds the promise to increase the probability of detection, decrease false alarm rates, and decrease the response time of fire detection systems.

In the rest of this paper, we first describe our methods (Sect. 2). Then, in Sect. 3, we present and discuss the empirical evaluation. Finally, in Sect. 4, we conclude.

## 2. Methodology

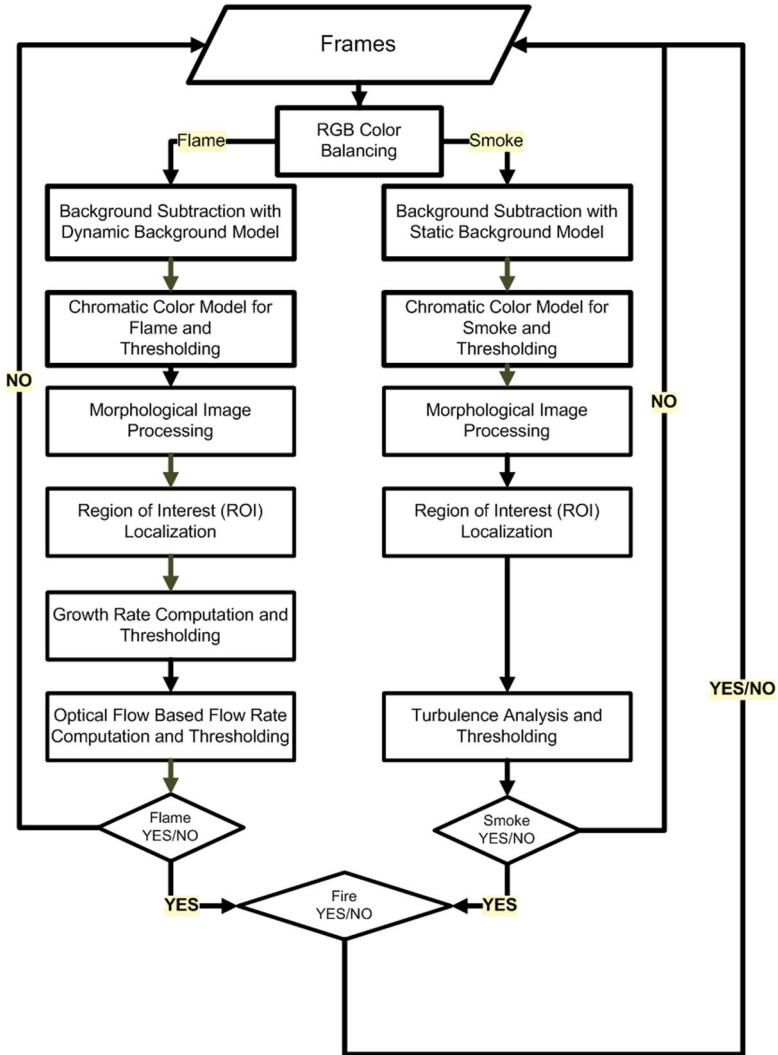
The general framework of QuickBlaze is shown in Figure 1. We provide details on each block in the following subsections.

### 2.1. Color Balancing

An essential preprocessing step of our video processing pipeline is color balancing. Color balancing is essential when object segmentation is based on a color model and the system may be deployed under different illumination conditions. Color balancing consists of two steps, estimation of the illumination and chromatic normalization by a scaling factor [40]. We assume that there is a good color distribution in each frame and therefore use a “gray-world” algorithm, one of the simplest illuminant estimation methods. We compute the average of the image intensities in the R, G, and B planes over each frame. The resulting vector of three intensities is known as the “gray-value” for the image. Each R, G, and B plane is then scaled independently using a multiplication factor that normalizes the gray-value to the average intensity of the frame in the R, G, and B planes. If a scaled value is greater than the maximum possible intensity, it is clamped to the maximum. After color balancing, the frames are fed to the independent flame and smoke detection pipelines as shown in Figure 1.

### 2.2. Motion Detection

Generally, vision sensors for fire detection are assumed to be fixed at a certain location and orientation, keeping the field of view and the background scene fixed. Motion regions are detected and segmented by extracting the foreground objects using a background model. Let the intensity of pixel  $(x, y)$  in frame  $t$  be represented by  $I(x, y; t)$ , and let the estimated background intensity of pixel  $(x, y)$  in frame  $t$  be denoted by  $B(x, y; t)$ . To determine whether the pixel positioned at  $(x, y)$  is moving, we first compute  $D_1(x, y; t)$ ,  $D_2(x, y; t)$ , and  $D_3(x, y; t)$ , given as



**Figure 1. QuickBlaze framework. Blocks shown in parallel can be executed in parallel on separate cores.**

$$D_1(x, y; t) \triangleq I(x, y; t) - I(x, y; t - 1),$$

$$D_2(x, y; t) \triangleq I(x, y; t) - I(x, y; t - 2),$$

$$D_3(x, y; t) \triangleq I(x, y; t) - B(x, y; t).$$

Let  $F(x, y; t)$  be a binary image that specifies whether pixel  $(x, y)$  is apparently moving in frame  $t$ .  $F(x, y; t)$  can be defined by

$$F(x, y; t) = \begin{cases} 1 & \text{if } |D_1(x, y; t)| > T(x, y; t) \wedge \\ & |D_2(x, y; t)| > T(x, y; t) \wedge \\ & |D_3(x, y; t)| > T(x, y; t) \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where  $\alpha$ , an update parameter, is a small positive real number close to zero. We use the same procedure to obtain  $F(x, y; t)$  for flame and smoke independently, but the background model  $B(x, y; t)$  and the threshold  $T(x, y; t)$  are different for flame and smoke.

*2.2.1. Dynamic Background Model and Adaptive Motion Threshold for Flame* For flame, the background is dynamically updated for pixels considered stationary using the update rule

$$B(x, y; t + 1) = \begin{cases} \alpha B(x, y; t) + (1 - \alpha)I(x, y; t) & \text{if } F(x, y, t) = 1 \\ B(x, y; t) & \text{otherwise.} \end{cases} \quad (2)$$

The threshold  $T(x, y; t)$  for flame is adaptive, updated according to two cases:

$$T(x, y; t + 1) = \begin{cases} \alpha T(x, y; t) + (1 - \alpha)|I(x, y; t) - B(x, y, t + 1)| & \text{if } F(x, y, t) \geq 1 \\ T(x, y; t) & \text{otherwise.} \end{cases} \quad (3)$$

We use the same  $\alpha$  for the adaptive threshold and the background model update.

*2.2.2. Static Background Model and Static Threshold for Smoke* Smoke pixels are generally more sparse than fire pixels, and the rate of change of the intensity of smoke pixels per frame is slower than that of flame. Regularly updating the background mode suppresses detection of slow-moving smoke pixels as foreground. To address this problem, we use a static background model  $B(x, y; t)$  for smoke detection that is computed at the start of video processing and an empirical constant  $T(x, y; n) = \tau$  for the threshold. The parameter  $\tau$  can be determined through experimental evaluation on training videos.

### 2.3. Chromatic Filtering

The color of flame is generally in the red-yellow or reddish range, while smoke, at least at the start of a fire, has a color that varies from bluish white to white. We therefore use separate color models to detect flame and smoke regions. Pixels are only considered as candidate members of potential flame or smoke regions if they are marked as being in motion according to the motion map  $F(x, y; t)$  described in Sect. 2.2 and then survive the chromatic filtering step for flame or smoke.

**2.3.1. Flame Color Model** To find the candidate fire pixels we follow the method of Chen et al. [9]. The first filter selects pixels that have an appropriate hue in the HSI color space. The method then eliminates from consideration any pixels outside the red-yellow range of  $0^\circ$  to  $60^\circ$  for H. Let the intensity of a pixel under consideration in the red, green, and blue channel be R, G, and B respectively. The hue filter happens to correspond to the RGB region where,  $R \geq G$  and  $G > B$ . The consequence of this filter is that for all selected pixels, in the RGB color space,

$$R \geq G > B.$$

To prevent selection of reddish pixels with low brightness and low saturation, we further filter out pixels with low red levels and low saturation levels. For the red level, we apply a simple threshold:

$$R > R_T,$$

where  $R$  is the red level of the pixel in question and  $R_T$  is an empirical threshold for the red level that is determined by calibrating on training videos. For saturation, we apply a slightly more complicated rule:

$$S \geq ((255 - R) \times \frac{S_T}{R_T})$$

The set of valid saturation and red levels according to this rule is shown in Figure 2. As a final step, following Chen et al., we filter out any candidate fire pixels with intensity below a threshold  $I_T$ . As with all other parameters, we set the values of  $R_T$ ,  $S_T$ , and  $I_T$  empirically using training videos.

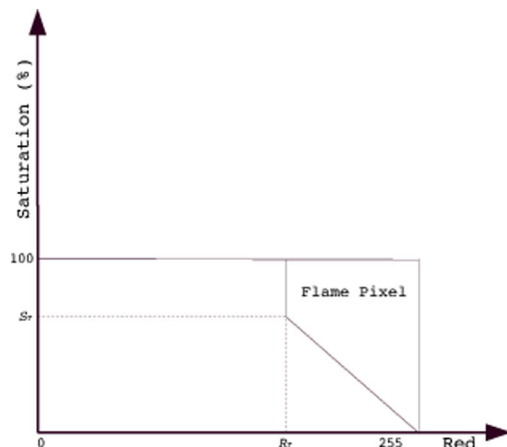
**2.3.2. Smoke Color Model** For the smoke color model, we used the rule presented by Celik et al. [41]:

$$|R - G| < T; |G - B| < T; |R - B| < T,$$

where threshold  $T$  is adjusted for good performance on the training video set.

## 2.4. Morphological Image Processing

After basic filtering according to motion and color cues, the resulting candidate regions are typically noisy. We perform morphological image processing on both the flame and smoke pixel maps to remove noise and extract final candidate regions. We first perform basic operations to close holes and remove small noise elements. We then obtain connected components and eliminate small components according to the length of the region's perimeter. Finally, we get a chain code for each surviving candidate fire and smoke region separately.



**Figure 2. Filtering for candidate flame pixels based on red level and saturation level.**

*2.4.1. Candidate Flame Regions* On the potential flame pixels, we perform morphological closing and then opening to eliminate small holes and connect nearby regions. Since fire normally moves and spreads vertically, we use a  $5 \times 5$  specifically-tuned vertical structuring element for closing and a standard  $3 \times 3$  rectangular structuring element for opening.

*2.4.2. Candidate Smoke Regions* Candidate smoke pixel regions tend to be more sparse than candidate flame pixel regions. Instead of using opening and closing, we used dilation and erosion (using  $3 \times 3$  rectangular structuring element) on the probable smoke pixels to remove the noise.

## 2.5. Localization of Candidate Regions

A frame may contain multiple candidate flame and smoke regions. Each candidate region should be tracked and localized separately to analyze growth rate and flow rate in case of flame regions or turbulence in case of smoke regions. When tracking regions from frame to frame, each region can be classified as either newly born or a previously existing region. To perform this classification, we logically AND the masks (obtained after filtering for motion and color morphology) for the current and previous frame. If all the values for a given foreground region evaluate to 0, the region is classified as newly born. If, on the other hand, some values for the given foreground region are non zero, the region is associated with the previously detected region. For both smoke and flame, a candidate region must survive (be associated with a region in the previous frame) for some number of frames before being declared evidence of an actual fire (see the next subsection for details).



## 2.6. Smoke Detection

For smoke, we use simplistic motion, color, and connectivity rules to finalize the set of detected regions, as detailed below.

**2.6.1. Absolute Color Change** When smoke appears in a scene, it tends to cause fading of the color of the object behind it, until it completely covers that object. This color change or increase in smoke opacity does not happen instantly, but happens gradually as the smoke density increases. When a smoke candidate region is detected, due to morphological processing in the previous step, the region may include not only actual smoke pixels but also neighboring background region or object pixels. This leads us to define a criterion for the presence of smoke: let  $(\delta R, \delta G, \delta B)$  be the absolute change in the color of a pixel from the previous frame to the current frame. We define a threshold  $C_{\max}$  such that to be considered part of a smoke region, the pixel must satisfy

$$\delta R < C_{\max} \wedge \delta G < C_{\max} \wedge \delta B < C_{\max}.$$

As with all other parameters, we fix  $C_{\max}$  for good performance on a set of training videos.

**2.6.2. Turbulence Rate Analysis** Smoke emanating from a fire is warmer and lighter than surrounding air and therefore tends to move upwards. The shape of a smoke region's projection onto the image plane is therefore complex; furthermore, its perimeter will change more abruptly than its area. This property can be modeled mathematically to more accurately detect possible smoke regions.

Let the perimeter of a candidate smoke blob be  $P(t)$ , and let the area of the blob be  $A(t)$ . We define a ratio  $\Omega(t)$  to represent the irregularity of the blob at time  $t$ :

$$\Omega(t) = \frac{P(t)}{2\sqrt{\pi A(t)}}.$$

We can characterize the *turbulence* of the blob [42], as the derivative of the irregularity with respect to time:

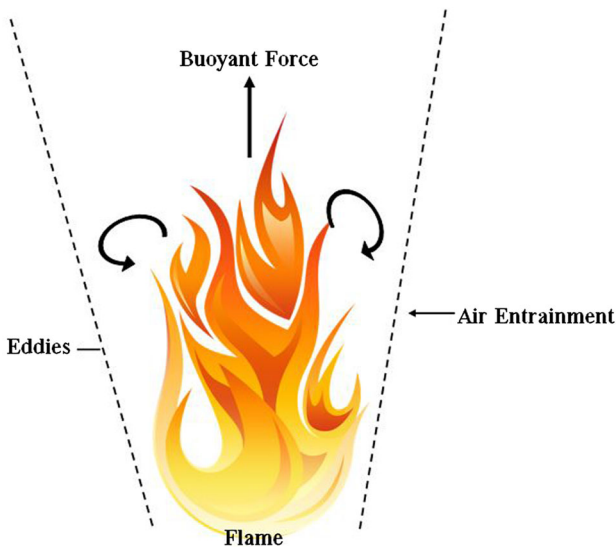
$$\frac{d\Omega}{dt} = \frac{2\frac{dP}{dt}A(t) - P(t)\frac{dA}{dt}}{4\sqrt{\pi A^3(t)}}.$$

$\frac{d\Omega}{dt}$  increases with the complexity of the shape. To increase our level of confidence in declaring a candidate region as smoke, we maintain a cumulative sum of  $\frac{d\Omega}{dt}$  and  $A(t)$  for  $\Delta_t$  frames. If the cumulative sum of  $\frac{d\Omega}{dt}$  for a region is greater than  $\Omega_{\text{threshold}}$ , and  $A(t)$  is greater than  $\Delta_A$ , we classify the region as a smoke region. The thresholds are empirically calculated on the training dataset.

## 2.7. Flame Detection Algorithm

Similar to the way in which we use pixel color change and turbulence analysis to eliminate false positive smoke regions, we can eliminate false positive flame regions when their behavior is inconsistent with the physics of flame. Fire is characterized by turbulent flames. As shown in Figure 3, as air is heated by a fire, since the density of a gas is inversely proportional to its temperature, a plume will rise above the burning object, causing upward motion. As air in the hot plume is cooled by the surrounding air, its buoyancy decreases, causing it to cease rising and start falling. As cool air is induced to flow into the fire plume (a process called *entrainment*), eddies form, creating rising vortices and turbulence. We use two heuristics, growth rate analysis and flow rate analysis, to characterize the dynamics typical of flame regions and to filter out regions unlikely to contain flame.

**2.7.1. Growth Rate Analysis** As explained in the previous section, generally, a burning fire will expand upward and outward, depending on the air flow and fuel type. Our growth rate analysis method exploits this feature of growing fires. When a candidate fire connected component is initially detected, we record its bounding box for the frame in which it is first detected. We then extract growth rate information from the next  $n$  frames. We measure growth separately in the left, right, upward, and downward directions. Only candidate regions that grow more in the upward direction than the other three directions are finally considered fire regions. Our method of measuring growth in each frame is as follows. In the first frame subsequent to detection of the region, we initialize a search region obtained from the original bounding box by expanding the bounding box by one pixel in all four directions and then removing the pixels corresponding to the original bounding



**Figure 3. Buoyancy and fire plumes.**

box. On each subsequent frame, we expand the search region by one pixel. In every frame, we count candidate fire pixels separately for the left, right, upward, and downward directions. We take the summation of the number of candidate fire pixels in the region over all  $n$  frames as a measure of how much the candidate fire blob has grown in the respective direction from the frame in which it was originally detected. Finally, we only retain as candidate fire regions those components whose growth in the upward direction is greater than its growth in the left, right, and downward directions at frame  $n$ .

**2.7.2. Flow Rate Analysis** Flow rate analysis is based on the hypothesis that image regions containing turbulent flames generate a large amount of flow compared to more rigid moving objects.

We use the intensity-based pyramidal Lucas-Kanade optical flow method with Shi and Tomasi keypoint features [43] to estimate motion in a candidate flame region from frame to frame. Lucas-Kanade first solves for the optical flow at the top of the pyramid, then at each layer below, it uses the motion estimates from the previous layer as a starting point. The method continues down the pyramid until it reaches the lowest level. Pyramidal Lucas-Kanade estimates a motion velocity vector for each feature point in the region of interest. The results at time  $t$  are

$$P^{(t)} = \{(p_{xi}^{(t)}, p_{yi}^{(t)})\}_{i \in 1, \dots, m^{(t)}}$$

$$Q^{(t)} = \{(q_{xi}^{(t)}, q_{yi}^{(t)})\}_{i \in 1, \dots, m^{(t)}}$$

where each  $\mathbf{p}_i^{(t)}$  and  $\mathbf{q}_i^{(t)}$  denote the starting and ending point of a feature point, respectively.  $m^{(t)}$  is the total number of feature points tracked from frame  $t - 1$  to frame  $t$ . We will use the velocity vectors  $\mathbf{q}_i^{(t)} - \mathbf{p}_i^{(t)}$  to calculate two features of the overall flow rate. The first feature is the average flow from frame 0 to frame 1:

$$F_0 = \frac{1}{m^{(1)}} \sum_{i=1}^{m^{(1)}} \sqrt{(p_{yi}^{(1)} - q_{yi}^{(1)})^2 + (p_{xi}^{(1)} - q_{xi}^{(1)})^2}. \tag{4}$$

$F_0$  is used as a reference value for the subsequent  $n - 1$  frames:

$$F(t) = \frac{1}{m^{(t)}} \sum_{i=1}^{m^{(t)}} \sqrt{(p_{yi}^{(t)} - q_{yi}^{(t)})^2 + (p_{xi}^{(t)} - q_{xi}^{(t)})^2}. \tag{5}$$

$$F_v = \frac{1}{n - 1} \sum_{t=2}^n (F(t) - F_0), \tag{6}$$

$F_v$  is the total growth in the average flow rate over a window of  $n$  frames. We expect the flow rate of turbulent fire regions to grow faster than that of other

moving objects. We fix a threshold for the flow rate growth  $F_T$ . We require  $F_v > F_T$  to declare a candidate flame region as a fire region.  $F_T$  is computed using training video data.

### 3. Experimental Results

We implemented the algorithm pipeline defined in Sect. 2 for fire detection using an Intel Core 2 Quad CPU Q9550 with four cores, running at 2.83GHz with 4 GB RAM and Ubuntu Linux as the operating system. The current implementation is sequential, but since the two processing streams are in fact independent, they can in principle easily be run on separate cores after the color balancing step. The software was written in C and C++ with the help of the OpenCV 2.4 library. The focus of our experiments is to analyze the ability of the system to detect fires as early as possible after the burning process has begun, so as key indicators of the algorithm's effectiveness, we measure the response time, the frame processing rate, and the false error detection rate. We manually adjusted the threshold parameters (Table 1) to enable the system to perform well on a set of three training videos, which contains two videos with flame and smoke regions and one distracter video containing an orange balloon.

After training, the system was tested on the videos listed in Table 2 without any modification to the parameters. Of course, some further calibration might be necessary to detect flames or smoke with characteristics very different from those in the videos tested, but the manually adjusted parameter settings proved suitable without modification for this set of test videos, including those with orange distracter objects, moving people and cars, and so on. Sample frames from each of the training videos are shown in Figure 4.

For testing we collected a total of 30 videos, 20 of which contain flame and/or smoke and 10 of which are distracters containing no flame or smoke. M-1, M-2, M-3, M-4, M-8, and M-22 were created by the authors [20]. M-5, M-6, M-9, M-10, M-11, M-12, M-13, M-14, M-21, M-23, M-24, and M-29 are from the Visi-Fire website [5]. M-7 was taken from NIST [44]. M-16, M-17, M-18, M-19, M-25, M-26, M-27, and M-28 were taken from Malenichev et al. [21]. To verify that our algorithm runs in real time, we created videos M-20 and M-30 using an IQEye IP



**Figure 4. Sample frames from training videos used for adjusting thresholds for QuickBlaze. T1 has 7343 frames, T2 has 1607 frames, and T3 has 3383 frames. T1 and T3 contain both fire and smoke regions, while T2 contains no fire or smoke but does contain an orange balloon as a distracter (Color figure online).**

**Table 1**  
**Threshold Parameters Adjusted for Good Performance on Training Videos**

No.	Parameter	Description	Value
1	$T(x, y, 0)$	Initial flame motion threshold	40
2	$\tau$	Smoke intensity change threshold	40
3	$\alpha$	Background averaging update ratio	0.02
4	$R_T$	Red flame intensity threshold	230
5	$S_T$	Flame color saturation threshold	0.7
6	$T_T$	Flame intensity threshold	150
7	$F_T$	Flame optical flow growth rate threshold	10
8	$T$	Smoke color compatibility threshold	10
9	$C_{\max}$	Smoke color difference threshold	60
10	$\Omega_{\text{threshold}}$	Smoke region area growth threshold	2.0
11	$\Delta_t$	Smoke region area observation window	3
12	$\Delta_A$	Smoke region area	100

camera with a resolution of  $720 \times 480$  pixels at 15 frames per second and ran our algorithm on the live streams as the videos were being recorded. M-1 through M-15 and M-20 contain both flame and smoke, while M-16 to M-19 only contain smoke. M-21, M-25, M-27, and M-28 contain distracter objects likely to be classified as smoke. M-22, M-24, M-29, and M-30 contain distracter objects likely to be classified as flame. M-23 has both flame-like and smoke-like objects. M-1, M-3, M-4, and M-22 are similar to the training videos, but they were created at different times with different target objects. Detailed descriptions of each video are given in Table 2, and a sample frame from each video that contains flame and smoke is shown in Figure 5. We selected the first frame containing flame or smoke according to human observation of the incident. Sample frames from the test videos that do not contain flame or smoke are shown in Figure 6.

In our implementation, to speed up total compute time, every frame is down-sampled to  $320 \times 240$  as a preprocessing step. For the ground truth, the first author manually identified the first frame of each video in which evidence of fire (flame or smoke) was evident. This manual identification process is tedious, and the precise localization of the first frame would no doubt differ for different people, especially for videos that contain faint smoke regions before flame appears. However, as we shall see, the ground truth frame numbers are always lower than those identified by the machine vision algorithms. Since any human error affects both of the tested algorithms equally, it can be ignored.

To demonstrate the effectiveness of combining the two methods in parallel streams, we evaluate the performance of our combined approach with the individual methods as described in [20] and [21]. In Table 3, we compare the combined approach to the individual methods in terms of fire detection time, and the data clearly show that the combined approach is better in terms of detection time than either of the individual methods alone. In Table 4, we compare the combined approach to the individual methods in terms of false alarm rates, and the data

**Table 2**  
**Video Sequences Selected for Testing**

Seq. no.	Description	Duration (frames)	Frames/s	Fire/smoke
M-1	Outdoor plain box	2208	24	Fire, smoke
M-2	Multiple boxes-1	4301	24	Fire, smoke
M-3	Printed box	5027	24	Fire, smoke
M-4	Multiple boxes-2	4500	24	Fire, smoke
M-5	Barbeque stand	439	10	Fire
M-6	Plant pot	707	15	Fire, smoke
M-7	Christmas tree room	1464	30	Fire
M-8	Warehouse fire	4389	30	Fire
M-9	Dark room flame	410	24	Fire
M-10	Indoor night 20 m heptane	1657	24	Fire
M-11	Outdoor daytime 10 m gasoline	3490	24	Fire
M-12	Outdoor daytime 10 m heptane	4547	24	Fire
M-13	Outdoor night 10 m gasoline	1207	24	Fire
M-14	Outdoor night 10 m heptane	3274	24	Fire
M-15	Hard paper flame on grass	3436	25	Fire, smoke
M-16	Wastebin smoke	900	10	Smoke
M-17	Smoke from car	155	30	Smoke
M-18	Behind the fence	629	10	Smoke
M-19	Ballastic smoke	347	30	Smoke
M-20	Box on fire outdoor	1160	15	None
M-21	Tunnel accident	190	15	None
M-22	Moving orange balloon	1183	24	None
M-23	Car lights at night	155	15	None
M-24	Moving man with red jacket	55	24	None
M-25	Black and white cat	87	25	None
M-26	Man with white shirt	223	25	None
M-27	Lady moving	794	25	None
M-28	Girl moving with white shirt	178	15	None
M-29	Flame-like neon sign	529	24	None
M-30	Flame-like CD cover	587	15	None

clearly show that the improvement in terms of response time comes at the cost of a small increase in false alarm rates, as the false alarms for the combined system will necessarily be the union of the false alarms for the individual systems.

For a comparative evaluation of our machine vision detection algorithm, we contrast our results with those obtained from VisiFire [5], a Windows-based program that is available for commercial use with a fee and for free limited academic use on request. VisiFire contains several different fire analyzers, including a fire and smoke detector, a forest fire detector, and an infrared sensor-based fire detector. We chose VisiFire because it runs in real time and has evolved through a series of academic research contributions [3, 27, 28, 37, 38]. In the program, we selected fire and smoke detection and used the default settings for the parameters available. The detailed settings are shown in Figure 7. We set the processing size for each video frame to  $320 \times 240$ , the same as used by our algorithm. The  $S\_T$  parameter shown in Figure 7 represents the threshold number of neighboring blocks with evidence of fire necessary to trigger a fire alarm. The software marks



**Figure 5. Sample frames from test videos that contain flame or smoke. In each case, the first frame containing flame or smoke according to human observations is shown. Although M-1, M-3, and M-4 are similar to videos in the training set, they were acquired at different times with different target objects.**

likely flame blocks with red squares and likely smoke blocks with dark blue squares. Although the VisiFire executable was run on Windows and our method was run on Ubuntu, the test video sequences were identical, and the runs were performed on the same hardware. VisiFire displays the frame number and number of frames currently being processed per second on its display, so we could easily record the first frame in which fire was detected and the overall frame processing rate by observing this display.

Comparisons of VisiFire and the QuickBlaze for response time, frames per second, and false alarms are shown in Tables 5 and 6.

In terms of speed of processing, although the frame rate varies from video to video based on the number of candidate fire and smoke regions, the proposed



**Figure 6. Arbitrary sample frames from test video sequences without any flame or smoke regions. Although M-22 is similar to one of the training videos, it was acquired at a different time with different target objects.**

method runs faster than VisiFire on every video, with a total overall speedup of 2.66.

In terms of accuracy, we observe that on 18 of the 20 videos that contain flame and/or smoke, fire detection using our proposed method results in better response time than using VisiFire.

In general, we observe that the fire incidents can be characterized by the visibility of any flame and/or smoke and depends upon the surrounding environment, the illumination of the scene, and the camera's field of view. We now take a closer look at results on specific videos.

Figure 8 compares detection of fire based on smoke regions by our method and by VisiFire. The first row shows the first frame of video sequence M-1, M-3, M-4, and M-15 respectively, in which a fire incident was first detected by our method, and the second row shows the first frame in the corresponding video in which VisiFire detected fire, if it did. VisiFire detected smoke regions in M-1 and M-2 (albeit much later than our method), a flame region in M-4, and did not detect any fire incident in M-15 (see Table 5 for details for all fire videos).

Figure 9 shows a similar comparison of fire detection based on flame regions. The first row shows the first frame of video sequence M-7, M-8, M-12, and M-14 respectively, in which a fire incident was first detected by our method, and the second row shows first frame in the corresponding video in which VisiFire detected fire, if it did. VisiFire detected smoke in M-7, flame in M-8, and did not detect a fire incidents in M-12 or M-14. Clearly, our method detects flame regions earlier than VisiFire in these videos. QuickBlaze tends to work very well when illumina-



**Table 3**  
**Experimental Results for Videos Containing Fire and/or Smoke**

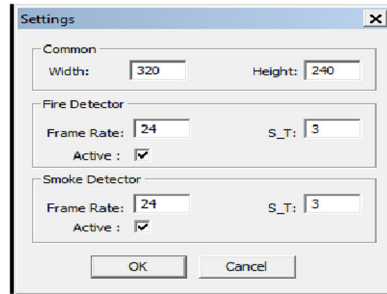
Seq. no.	First fire frame (Ground truth)	First fire frame (Method [20])	First fire frame (QuickBlaze)	First fire frame (Method [21])
M-1	1037	1213	<b>1158</b>	<b>1158</b>
M-2	246	2304	<b>694</b>	<b>694</b>
M-3	990	1665	<b>1001</b>	<b>1001</b>
M-4	1	873	<b>440</b>	<b>440</b>
M-5	1	11	<b>11</b>	Not-detected
M-6	50	401	<b>323</b>	<b>323</b>
M-7	251	<b>305</b>	<b>305</b>	761
M-8	1685	<b>1755</b>	<b>1755</b>	Not-detected
M-9	1	<b>12</b>	<b>12</b>	Not-detected
M-10	120	<b>133</b>	<b>133</b>	156
M-11	90	<b>105</b>	<b>105</b>	426
M-12	120	<b>207</b>	<b>207</b>	Not-detected
M-13	70	<b>104</b>	<b>104</b>	Not-detected
M-14	92	<b>168</b>	<b>168</b>	231
M-15	1	1985	<b>594</b>	<b>594</b>
M-16	19	Not-detected	<b>87</b>	<b>87</b>
M-17	19	Not-detected	<b>28</b>	<b>28</b>
M-18	54	Not-detected	<b>233</b>	<b>233</b>
M-19	59	Not-detected	<b>71</b>	<b>71</b>
M-20	1	<b>35</b>	<b>35</b>	<b>201</b>

Column two shows the ground truth frame number in which the fire is first visible. Columns three, four, and five show the frame number in which Method [20], Method [21], and QuickBlaze detect fire in, respectively. Bold text indicates the best result

**Table 4**  
**Experimental Results for Distracter Videos (Videos without any Fire or Smoke)**

Seq. no.	FER (Method [20], %)	FER (QuickBlaze, %)	FER (Method [21], %)
M-21	0	6	6
M-22	0	0.2	0.2
M-23	0	0	0
M-24	0	0	0
M-25	0	0	0
M-26	0	0	0
M-27	0	0	0
M-28	0	0	0
M-29	1	1	0
M-30	0	0	0

FER (false error rate) is the percentage of frames in the video classified as fire using Method [20], QuickBlaze and Method [21]



**Figure 7. Settings used for VisiFire fire and smoke analyzer in the experimental comparison.**

**Table 5  
Experimental Results for Videos Containing Fire and/or Smoke**

Seq. No.	First fire frame (ground truth)	First fire frame (VisiFire)	First fire frame (QuickBlaze)	Processing speed (FPS, VisiFire)	Processing speed (FPS, QuickBlaze)
M-1	1037	1165	<b>1158</b>	4.86	<b>14.1</b>
M-2	246	1427	<b>694</b>	8.76	<b>14.1</b>
M-3	990	1599	<b>1001</b>	8.09	<b>14.0</b>
M-4	1	1460	<b>440</b>	11.82	<b>13.6</b>
M-5	1	37	<b>11</b>	11.40	<b>36.11</b>
M-6	50	<b>187</b>	323	11.79	<b>13.8</b>
M-7	251	370	<b>305</b>	8.81	<b>29.2</b>
M-8	1685	1858	<b>1755</b>	8.31	<b>28.6</b>
M-9	1	70	<b>12</b>	11.97	<b>34.1</b>
M-10	120	179	<b>133</b>	16.02	<b>32.7</b>
M-11	90	171	<b>105</b>	13.64	<b>32.6</b>
M-12	120	Not detected	<b>207</b>	12.5	<b>32.2</b>
M-13	70	454	<b>104</b>	12.1	<b>33.3</b>
M-14	92	Not detected	<b>168</b>	12.39	<b>32.5</b>
M-15	1	Not detected	<b>594</b>	11.54	<b>33.3</b>
M-16	19	<b>37</b>	87	13.94	<b>34.0</b>
M-17	19	135	<b>28</b>	11.23	<b>37.9</b>
M-18	54	241	<b>233</b>	13.22	<b>32.0</b>
M-19	59	100	<b>71</b>	11.73	<b>34.8</b>
M-20	1	39	<b>35</b>	6.77	<b>29.8</b>

Column two shows the ground truth frame number in which the fire is first visible. Columns three and four show the frame number in which VisiFire and QuickBlaze detect fire in, respectively. Columns five and six show the processing speed of VisiFire and QuickBlaze, in frames per second. Bold text indicates the best result

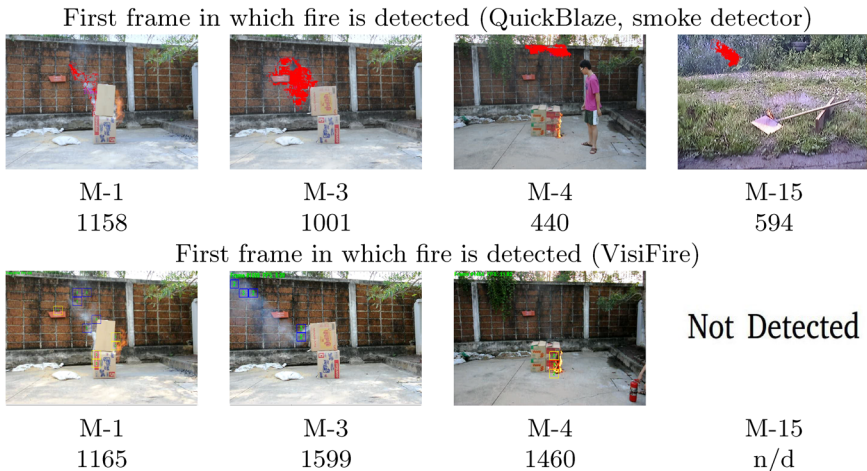
tion is dim relative to any flame and when clutter is low with little background contrast.

In two cases, VisiFire detected flame or smoke regions earlier than the proposed method. See Figure 10 for a visual comparison of the frames in which fire was first detected by the proposed method and VisiFire. In video M-6, the flame is

**Table 6**  
**Experimental Results for Distracter Videos (Videos Without any Fire or Smoke)**

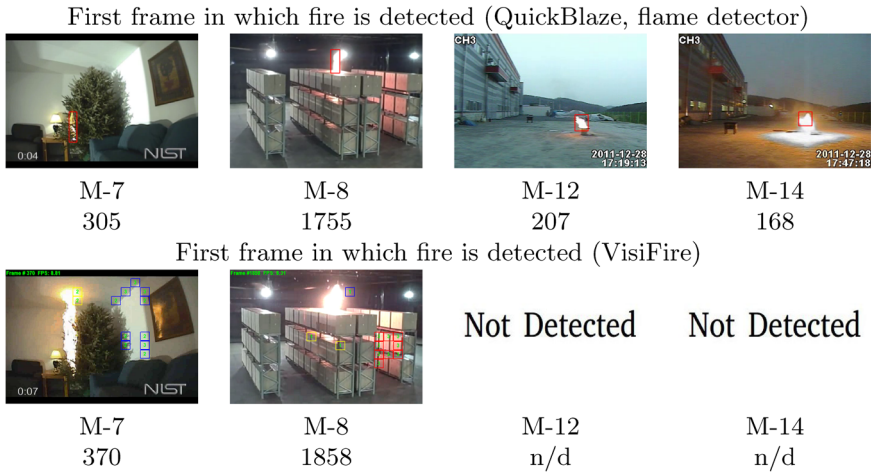
Seq. no.	FER (VisiFire, %)	FER (QuickBlaze, %)	Processing speed (FPS, VisiFire)	Processing speed (FPS, QuickBlaze)
M-21	5	6	13.46	<b>40.9</b>
M-22	0.2	0.2	11.75	<b>37.7</b>
M-23	8	0	13.49	<b>43.1</b>
M-24	0	0	15.32	<b>45.0</b>
M-25	0	0	16.6	<b>43.3</b>
M-26	0	0	16.5	<b>45.2</b>
M-27	0	0	15.6	<b>41.6</b>
M-28	4	0	9.88	<b>43.5</b>
M-29	1	1	16.62	<b>51.8</b>
M-30	3	0	13.14	<b>13.7</b>

FER (false error rate) is the percentage of frames in the video classified as fire using VisiFire and QuickBlaze

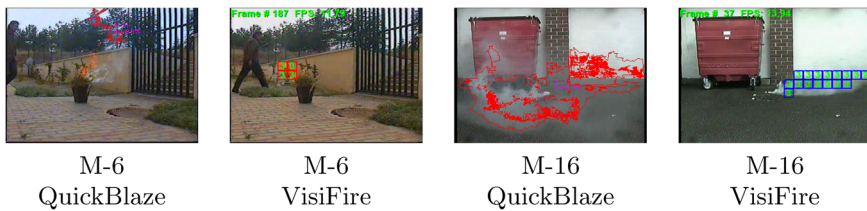


**Figure 8. First frame in which fire is detected through smoke detection by QuickBlaze. The first row shows the first frame in video sequences M-1, M-3, M-4, and M-15 in which the fire incident was detected by the proposed method’s smoke region detector. The second row shows the first frame in the same sequences in which the fire incident was detected by VisiFire. VisiFire failed to detect any fire incident in video M-15.**

quite transparent, so the background shows through, reducing our method’s ability to detect it. The method does detect the smoke given off by the fire, but later than VisiFire, which is able to detect the flame early on. In the second case, VisiFire rapidly detects the thick, opaque smoke initially coming from the smoke



**Figure 9. First frame in which fire is detected through flame detection by QuickBlaze. The first row shows the first frame in video M-7, M-8, M-12, and M-14 in which the fire incident was detected by the proposed method’s flame region detector. The second row shows the first frame in the same sequences in which the fire incident was detected by VisiFire. VisiFire failed to detect any fire incident in videos M-12 and M-14.**

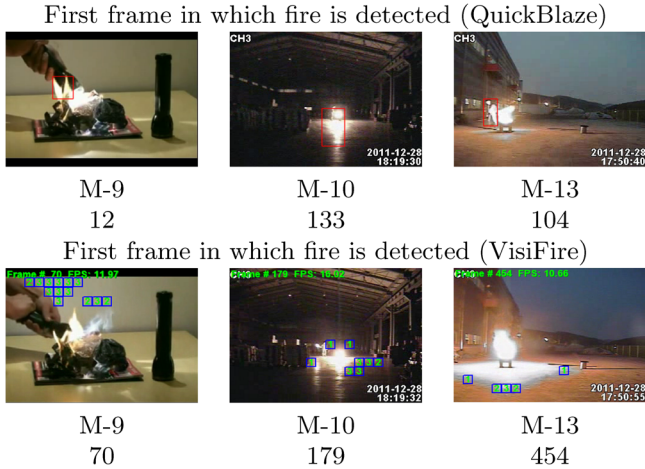


**Figure 10. First frame in which fire was detected by QuickBlaze and VisiFire on two videos for which the QuickBlaze’s response time was slow than VisiFire’s.**

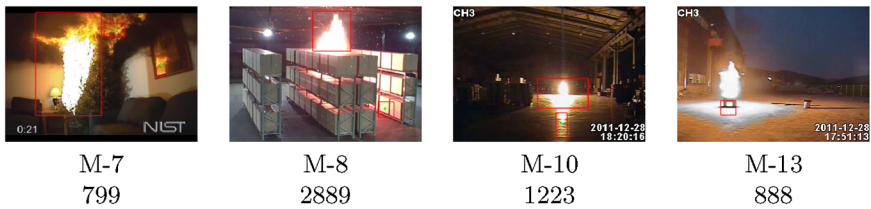
bomb, whereas our method performs better on thin, wispy smoke, resulting in later detection once the smoke is more diffuse.

We also observe that our algorithm precisely localizes fire incidents in all of the test video sequences, both in daytime and nighttime. This is shown in Figure 11. The first row shows the first frame in which a fire incident is detected by our method in nighttime videos M-9, M-10, and M-13, while the second row shows the VisiFire result for the corresponding sequences. Visual comparison clearly shows that the proposed method better localizes the fire incident than does VisiFire.

Although the best possible threshold parameters may vary with the type of fire or smoke, we observe that the threshold parameters we obtained manually using



**Figure 11. The first frame in which fire incidents were detected in nighttime video sequences M-9, M-10, and M-13. QuickBlaze detected flame regions, while VisiFire detected smoke regions. Note the better localization of the fire by the proposed method.**

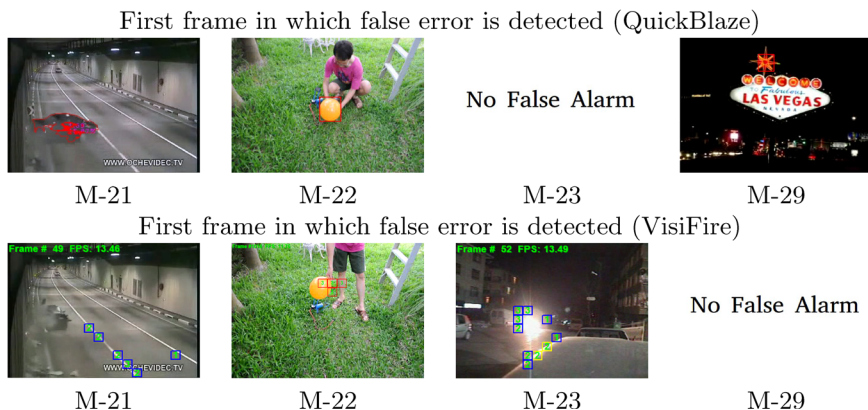


**Figure 12. Frame in which fire was detected from an indirect surface that reflected the radiation emitted by fire (QuickBlaze).**

the training videos provide very good performance across a wide variety early fire detection scenarios. For instance, in video sequences M-7, M-8, M-10, and M-13, actual fires were detected by their indirect reflections from other surfaces. These results can be seen in Figure 12.

As a final comparison, Figure 13 shows sample false detections in M-21, M-22, and M-23. M-21 contains a cloud of dust produced by a car accident, which triggers a false alarm for our method. Our method was distracted by the partly orange neon sign in M-29 whereas VisiFire was not; VisiFire, on the other hand, was distracted by the headlights in M-23, whereas our method was not.

This extensive evaluation in comparison with state of the art commercial software shows that our method responds to fire incidents sooner, processes the images faster, has a slightly lower false alarm rate, and better localizes fire incidents, especially at night-time.



**Figure 13. Sample frames from videos M-21, M-22, M-23, and M-29 in which false alarms were raised by the QuickBlaze or VisiFire. No false alarm was raised for M-23 by QuickBlaze. No false alarm was raised for M-29 by VisiFire.**

#### 4. Conclusion

In this paper, we have proposed QuickBlaze, a real-time early fire incident detection system that detects both flame and smoke regions in video images. We use parallel image processing streams to detect flame and smoke regions at high speed, with low response times. Our method does not require any offline training, although manual adjustment of parameters during a calibration phase is required to cater to the particular camera's depth of view and surrounding environment. We compare the combined approach to the individual methods in terms of false alarm rates, and the data clearly show that the improvement in terms of response time comes at the cost of a small increase in false alarm rates, as the false alarms for the combined system will necessarily be the union of the false alarms for the individual systems. We evaluate the algorithm on videos of a variety of real-world fires, and we have also performed a live test. Our focus in this paper is on detection of bright red-to-orange flame and white smoke typical of early fires [45, 46]. Detecting different kinds of smoke and flame more typical of mature fires is out of the scope of this paper. For comparison, we benchmark our method against commercial real-time fire and smoke detection software that evolved through a series of academic research contributions. We find that QuickBlaze has a better response time, is faster, and provides better fire localization than the commercial system. QuickBlaze could be deployed in any environment, and would likely have a faster response time than smoke detectors, as long as the fire or smoke is in view of the camera sensor. This does mean however that it may be most practical for large open spaces (commercial or industrial spaces) and less practical for areas with many small rooms such as residences. Furthermore, one sensor system might not be the best possible detector for all situations. In indoor environments, it

would of course be possible to augment the video based approach reported upon here with traditional point based sensors such as demonstrated by [47–50].

## Acknowledgments

WSQ was supported by fellowships from the faculty development program at the National University of Science and Technology (NUST), Pakistan, and the Asian Institute of Technology. We thank Prof. A. Enis Çetin for the demo license for VisiFire software.

## References

1. Mahdipour E, Dadkhah C (2014) Automatic fire detection based on soft computing techniques: review from 2000 to 2010. *Artif Intell Rev* 42(4):895–934
2. Töreyn BU, Dedeoğlu Y, Güdükbay U, Çetin AE (2006) Computer vision based method for real-time fire and flame detection. *Pattern Recogn Lett* 27(1):49–58
3. Çetin AE, Dimitropoulos K, Gouverneur B, Grammalidis N, Günay O, Habiboglu YH, Töreyn BU, Verstockt S (2013) Video fire detection—review. *Digit Signal Proc* 23(6):1827–1843
4. Security G (2014) Video fire detection system (VFDS) solution. URL [http://www.gkbsecurity.com/VFDS/vfds\\_solution.php](http://www.gkbsecurity.com/VFDS/vfds_solution.php)
5. VisiFire (2009) Video fire detection system (VFDS) solution. URL <http://signal.ee.bilkent.edu.tr/VisiFire/>
6. FIKE (2014) SIGNIFIRE video flame, smoke and Intrusion detection system. URL <http://www.fike.com/products/signifire-video-flame-smoke-intrusion-detection-system/>
7. Piccardi M (2004) Background subtraction techniques: a review. In: 2004 IEEE international conference on systems, man and cybernetics, vol 4, pp 3099–3104
8. Chen TH, Wu PH, Chiou YC (2004) An early fire-detection method based on image processing. *International Conference on Image Processing, ICIP* 3:1707–1710
9. Chen T, Yin Y, Huang S, Ye Y (2006) The smoke detection for early fire-alarming system base on video processing. In: *International conference on intelligent information hiding and multimedia signal processing*, pp 427–430 (2006)
10. Cui Y, Dong H, Zhou E (2008) An early fire detection method based on smoke texture analysis and discrimination. In: *Congress on image and signal processing, 2008. CISP'08*, vol 3, pp 95–99. IEEE (2008)
11. Chang SK, Chao HT, Chu HS, Huang KL, Lu CH, Wang CW (2012) Method and system for detecting flame. US Patent 8,311,345
12. Kopilovic I, Vagvolgyi B, Szirányi T (2000) Application of panoramic annular lens for motion analysis tasks: surveillance and smoke detection. In: *15th international conference on pattern recognition, 2000. Proceedings*, vol 4, pp 714–717. IEEE (2000)
13. Celik T, Demirel H (2009) Fire detection in video sequences using a generic color model. *Fire Saf J* 44(2):147–158
14. Ko BC, Ham SJ, Nam JY (2011) Modeling and formalization of fuzzy finite automata for detection of irregular fire flames. *IEEE Trans Circuits Syst Video Technol* 21(12):1903–1912
15. Nguyen-Ti T, Nguyen-Phuc, T, Do-Hong, T.: Fire detection based on video processing method. In: *2013 International Conference on Advanced Technologies for Communications (ATC)*, , pp. 106–110 (2013)

16. Hongda T, Wanqing L, Lei W, Ogunbona P.: "A Novel Video-Based Smoke Detection Method Using Image Separation". In: Multimedia and Expo (ICME), 2012 IEEE International Conference on, pp. 532–537 (2012)
17. Tian H, Li W, Wang L, Ogunbona P (2014) Smoke Detection in Video: An Image Separation Approach. *Int J Comput Vision* 106(2):192–209
18. Millan-Garcia L, Sanchez-Perez G, Nakano M, Toscano-Medina K, Perez-Meana H, Rojas-Cardenas L (2012) An early fire detection algorithm using IP cameras. *Sensors* 12(5):5670–5686
19. Tung TX, Kim JM (2011) An effective four-stage smoke-detection algorithm using video images for early fire-alarm systems. *Fire Saf J* 46(5):276–282
20. Rinsurongkawong S, Ekpanyapong M, Dailey M (2012) Fire detection for early fire alarm based on optical flow video processing. In: 2012 9th international conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON), pp 1–4
21. Malenichev A, Krasotkina O (2013) "Real-time smoke detection in video sequences: combined approach. In: Pattern recognition and machine intelligence. Springer, pp 445–450
22. Yang J, Chen F, Zhang W (2008) Visual-based smoke detection using support vector machine. In: Fourth International Conference on natural computation, 2008. ICNC '08, vol 4, pp 301–305
23. Günay O, Taşdemir K, Töreyn BU, Çetin AE (2010) Fire detection in video using LMS based active learning. *Fire Technol* 46(3):551–577. doi:10.1007/s10694-009-0106-8
24. Grech-Cini H (2005) Smoke detection. URL <https://www.google.com/patents/US6844818>. US Patent 6,844,818
25. Calderara S, Piccinini P, Cucchiara R (2011) Vision based smoke detection system using image energy and color information. *Mach Vis Appl* 22(4):705–719
26. Ko BC, Cheong KH, Nam JY (2009) Fire detection based on vision sensor and support vector machines. *Fire Saf J* 44(3):322–329
27. Toreyin BU, Dedeoglu Y, Cetin AE (2005) Flame detection in video using hidden markov models. In: IEEE International Conference on image processing, 2005. ICIP 2005, vol 2, IEEE, pp II-1230
28. Toreyin BU, Dedeoglu Y, Cetin AE (2006) Contour based smoke detection in video using wavelets. In: European signal processing conference, pp 123–128
29. Wang L, Ye M, Zhu Y (2010) A hybrid fire detection using Hidden Markov Model and luminance map. In: 2010 international conference on medical image analysis and clinical applications (MIACA), pp 118–122
30. Teng Z, Kim JH, Kang DJ (2010) Fire detection based on hidden Markov models. *Int J Control Autom Syst* 8(4):822–830
31. Tipsuwanporn V, Krongratana V, Gulpanich S, Thongnopakun K.: Fire detection using neural network. In: SICE-ICASE, 2006. International Joint Conference. IEEE, pp 5474–5477
32. Chunyu Y, Jun F, Jinjun W, Yongming Z (2010) Video fire smoke detection using motion and color features. *Fire Technol* 46(3):651–663. doi:10.1007/s10694-009-0110-z
33. Yu C, Mei Z, Zhang X (2013) A real-time video fire flame and smoke detection algorithm. *Procedia Eng* 62:891–898
34. Wang S, He Y, Zou J, Duan B, Wang J (2014) A flame detection synthesis algorithm. *Fire Technol* 50(4):959–975. doi:10.1007/s10694-012-0321-6
35. Chen J, He Y, Wang J (2010) Multi-feature fusion based fast video flame detection. *Build Environ* 45(5):1113–1122



36. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: DARPA Imaging Understanding Workshop, pp 121–130
37. Toreyin BU, Cetin AE (2007) Online detection of fire in video. 2013 IEEE conference on computer vision and pattern recognition 0, 1–5 (2007)
38. Toreyin BU, Dedeoğlu Y, Güdükbay U, Cetin AE (2006) Computer vision based method for real-time fire and flame detection. *Pattern Recognit Lett* 27(1):49–58
39. Qureshi WS (2015) QuickBlaze. URL <http://vgl-ait.org/cvwiki/doku.php?id=quickblaze:main>
40. Iyengar P (2013) Color constancy: gray world algorithm. URL <http://pi-virtualworld.blogspot.com/2013/09/color-constancy-gray-world-algorithm.html>
41. Çelik T, Özkaramanli H, Demirel H (2007) Fire and smoke detection without sensors: image processing based approach. In: 15th European signal processing conference, EUSIPCO, pp 147–158
42. Catrakis HJ, Dimotakis PE (1998) Shape complexity in turbulence. *Phys Rev Lett* 80(5):968
43. Shi J, Tomasi C (1994) Good features to track. In: 1994 IEEE computer society conference on computer vision and pattern recognition, 1994 CVPR'94. Proceedings, pp 593–600. IEEE (1994)
44. McGrattan K, Klein B, Hostikka S, Floyd J. *Fire Dynamics Simulator (Version 6) User's Guide*
45. Charles E, Baukal JR, Robert ES (2001) *The John Zink combustion handbook*. John Zink Company LLC, Tulsa
46. David WD (2007) Where there's smoke: the fire officer's guide to reading smoke. *Fire Rescue* 9 (2007)
47. Owrutsky JC, Steinhurst DA, Minor CP, Rose-Pehrsson SL, Williams FW, Gottuk DT (2006) Long wavelength video detection of fire in ship compartments. *Fire Saf J* 41(4):315–320
48. Gottuk DT, Lynch JA, Rose-Pehrsson SL, Owrutsky JC, Williams FW (2006) Video image fire detection for shipboard use. *Fire Saf J* 41(4):321–326. 13th International Conference on Automatic Fire Detection, Duisburg, Germany AUBE '04 13th International Conference on Automatic Fire Detection, Duisburg, Germany
49. Rose-Pehrsson SL, Minor CP, Steinhurst DA, Owrutsky JC, Lynch JA, Gottuk DT, Wales SC, Farley JP, Williams FW (2006) Volume sensor for damage assessment and situational awareness. *Fire Saf J* 41(4):301–310
50. Minor CP, Johnson KJ, Rose-Pehrsson SL, Owrutsky JC, Wales SC, Steinhurst DA, Gottuk DT (2010) A full-scale prototype multisensor system for damage control and situational awareness. *Fire Technol* 46(2):437–469. doi:10.1007/s10694-009-0103-y