

Real Time based Fire & Smoke Detection without Sensor by Image Processing

Prerna B. Pagar & A. N. Shaikh

Dept. of Electronics & Telecommunication, L.E.C.T.'s Savitribai Phule Women's Engineering College,
Aurangabad, Maharashtra, India

E-mail : prerna.pagar@gmail.com, aamer_be2005@gmail.com

Abstract – This paper gives a literature review on a novel method to detect fire and smoke in real time by processing video data generated by an ordinary CCTV camera monitoring a scene. Fire and smoke is a disaster that can strike anywhere and can be very destructive. A method to detect smoke and fires would allow the authorities to detect and put out the fires before it becomes out of control by giving the alarm. For fire detection the proposed method uses RGB and YCbCr colour space as well as for smoke detection uses Gaussian mixture model (GMM) in HSV colour space. Detection of fire and smoke pixels is at first achieved by means of a motion detection algorithm. In addition, separation of smoke and fire pixels using colour information (within appropriate spaces, specifically chosen in order to enhance specific chromatic features) is performed. In parallel, a pixel selection based on the dynamics of the area is carried out in order to reduce false detection. The outputs of the three parallel algorithms are eventually fused by means of a Multi-Layers Perceptron (MLP). This paper presents a brief survey on real time based fire & smoke detection without sensor by image processing.

Keywords – Image processing, Fire detection, Smoke detection, Generic colour model, Gaussian mixture model, Smoke dynamic features, Multi-Layers Perceptron

I. INTRODUCTION

Fire detection systems are one of the most important components in surveillance systems used to monitor buildings and environment as part of an early warning mechanism that reports preferably the start of fire. Currently, almost all fire detection systems use built-in sensors that primarily depend on the reliability and the positional distribution of the sensors. The sensors should be distributed densely for a high precision fire detector system. In a sensor-based fire detection system, coverage of large areas in outdoor applications is impractical due to the requirement of regular distribution of sensors in close proximity. Due to

the rapid developments in digital camera technology and video processing techniques, there is a big trend to replace conventional fire detection techniques with computer vision-based systems.

In general computer vision-based fire detection systems employ three major stages [1–4]. First stage is the flame pixel classification; the second stage is the moving object segmentation, and the last part is the analysis of candidate regions. This analysis is usually based on two figures of merit; shape of the region and the temporal changes of the region.

II. FIRE DETECTION

A. Introduction

The fire detection performance depends critically on the performance of the flame pixel classifier which generates seed areas on which the rest of the system operates. The flame pixel classifier is thus required to have a very high detection rate and preferably a low false alarm rate. There exist few algorithms which directly deal with the flame pixel classification in the literature. The flame pixel classification can be considered both in grayscale and colour video sequences. Krull et al. [5] used low-cost CCD cameras to detect fire in the cargo bay of long range passenger aircraft. The method uses statistical features, based on grayscale video frames, including mean pixel intensity, standard deviation, and second-order moments, along with non-image features such as humidity and temperature to detect fire in the cargo compartment. The system is commercially used in parallel to standard smoke detectors to reduce the false alarms caused by the smoke detectors.

The system also provides visual inspection capability which helps the aircraft crew to confirm the presence or absence of fire. However, the statistical image features are not considered to be used as part of a

standalone fire detection system. Most of the works on flame pixel classification in colour video sequences are rule based. Chen et al. [1] used raw R, G, and B information and developed a set of rules to classify the flame pixels. Instead of using the rule-based colour model as in Chen et al., To "reyin et al. [2] used a mixture of Gaussians in RGB space which is obtained from a training set of flame pixels. In a recent paper, the authors employed Chen's flame pixel classification method along with a motion information and Markov field modeling of the flame flicker process [3]. Marbach et al. [6] used YUV colour model for the representation of video data, where time derivative of luminance component Y was used to declare the candidate fire pixels and the Chrominance components U and V were used to classify the candidate pixels to be in the fire sector or not. In addition to luminance and chrominance they have incorporated motion into their work. They report that their algorithm detects less than one false alarm per week; however, they do not mention the number of tests conducted. Homg et al.[7] used HSI colour model to roughly segment the fire-like regions for brighter and darker environments. Initial segmentation is followed by removing lower intensity and lower saturation pixels. In order to get rid of the spurious fire-like regions such as smoke.

They also introduced a metric based on binary contour difference images to measure the burning degree of fire flames into classes such as "no fire", "small", "medium" and "big" fires. They report 96.94% detection rate, together with results including false positives and false negatives for their algorithms. However, there is no attempt to reduce the false positives and false negatives by changing their threshold values. Celik et al. [4] used normalized RGB (rgb) values for a generic colour model for the flame. The normalized RGB is proposed in order to alleviate the effects of changing illumination. The generic model is obtained using statistical analysis carried out in r-g, r-b, and g-b planes. Due to the distribution nature of the sample fire pixels in each plane, three lines are used to specify a triangular region representing the region of interest for the fire pixels. Therefore, triangular regions in respective r-g, r-b, and g-b planes are used to classify a pixel. A pixel is declared to be a fire pixel if it falls into three of the triangular regions in r-g, r-b, and g-b planes. Even though normalized RGB colour space overcomes to some extent the effects of variations in illumination, further improvement can be achieved if one uses YCbCr colour space which makes it possible to separate luminance/illumination from chrominance.

In this paper we propose to use the YCbCr colour space to construct a generic chrominance model for flame pixel classification. In addition to translating the

rules developed in the RGB and normalized rgb to YCbCr colour space, new rules are developed in YCbCr colour space which further alleviate the harmful effects of changing illumination and improves detection performance... This is a significant improvement over other methods used in the literature.

B. Classification of flame pixels

Each digital colour image is composed of three colour planes: red, green, and blue (R, G, and B). Each colour plane represents a colour-receptor in human eye working on different wavelength. The combination of RGB colour planes gives ability to devices to represent a colour in digital environment. Each colour plane is quantized into discrete levels. Generally 256 (8 bits per colour plane) quantization levels are used for each plane, for instance white is represented by (R, G, B) $\frac{1}{4}$ (255, 255, 255) and black is represented by (R, G, B) $\frac{1}{4}$ (0, 0, 0). A colour image consists of pixels, where each pixel is represented by spatial location in rectangular grid (x, y), and a colour vector (R(x, y), G(x, y), B(x, y)) corresponding to spatial location (x, y). Each pixel in a colour image containing a fire blob (region containing fire), the value of Red channel is greater than the Green channel, and the value of Green channel is greater than the value of Blue channel for the spatial location. Furthermore, the flame colour has high saturation in Red channel [1, 4]. For instance in Fig. 1 (a) shows samples of digital colour

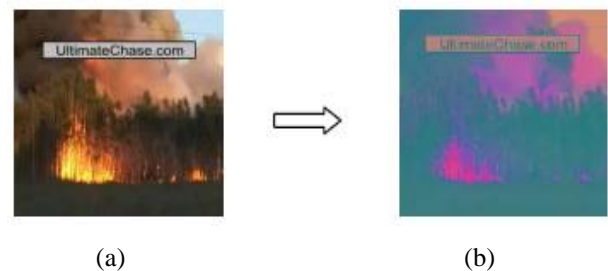


Fig. 1(a) shows input sample of digital colour video image

Fig. 1(b) shows R, G, and B colour planes (channels), Respectively

Image and fig. 1 (b) show R, G, and B colour planes (channels), respectively. It can be noticed from Fig. 1 that for the fire regions, R channel has higher intensity values than the G channel, and G channel has higher intensity values than the B channel.

In order to explain this idea better, we picked sample images from Fig. 1(a), and segmented its fire pixels as shown in Fig. 1(b) with RGB colour. Then we calculate mean values of R, G, and B planes in the segmented fire regions of the original images. It is clear that, on the average, the fire pixels show the

characteristics that their R intensity value is greater than G and G intensity value is greater than the B.

Even though RGB colour space can be used for pixel classification, it has disadvantages of illumination dependence. It means that if the illumination of image changes, the fire pixel classification rules can not perform well. Furthermore, it is not possible to separate a pixel's value into intensity and chrominance. The chrominance can be used in modeling colour of fire rather than modeling its intensity. This gives more robust representation for fire pixels. So it is needed to transform RGB colour space to one of the colour spaces where the separation between intensity and chrominance is more discriminate. Because of the linear conversion between RGB and YCbCr colour spaces, we use YCbCr colour space to model fire pixels. The conversion from RGB to YCbCr colour space is formulated as follows [8]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

Where Y is luminance, Cb and Cr are Chrominance Blue and Chrominance Red components, respectively. The range of Y is [16 235], Cb and Cr are equal to [16 240]. For a given image, one can define the mean values of the three components in YCbCr colour space as

$$\begin{aligned} Y_{\text{mean}} &= \frac{1}{K} \sum_{i=1}^K Y(x_i, y_i) \\ Cb_{\text{mean}} &= \frac{1}{K} \sum_{i=1}^K Cb(x_i, y_i) \\ Cr_{\text{mean}} &= \frac{1}{K} \sum_{i=1}^K Cr(x_i, y_i) \end{aligned} \quad (2)$$

Where (x_i, y_i) is the spatial location of the pixel, Y_{mean} , Cb_{mean} , and Cr_{mean} are the mean values of luminance, Chrominance Blue, and Chrominance Red channels of pixels, and K is the total number of pixels in image.

The rules defined for RGB colour space, i.e. $R \geq G \geq B$ and $R \geq R_{\text{mean}}$ [4, 1], can be translated into YCbCr space as

$$Y(x, y) > Cb(x, y) \quad (3)$$

$$Cr(x, y) > Cb(x, y) \quad (4)$$

Where $Y(x, y)$, $Cb(x, y)$, and $Cr(x, y)$ are luminance, Chrominance Blue and Chrominance Red values at the spatial location (x, y) Eq. (3) and (4) imply, respectively, that flame luminance should be greater than Chrominance Blue and Chrominance Red should be greater than the Chrominance Blue. Eq. (3) and (4) can be interpreted to be a consequence of the fact that the flame has saturation in red colour channel (R).

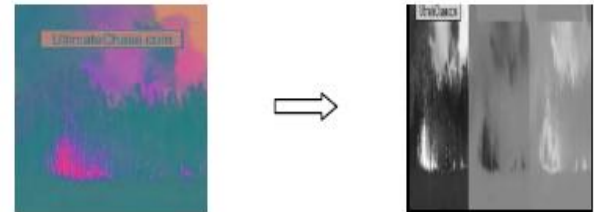


Fig. 2, we show the RGB images and its corresponding Y, Cb, and Cr channel responses for the images shown in Fig. 1(a).

In Fig. 2, we show the RGB images and its corresponding Y, Cb, and Cr channel responses for the images shown in Fig. 1(a). The validity of Eq. (3) and (4) can easily be observed for fire regions. Similar to Table 1, we picked sample images from Fig. 1(a), and segmented its fire pixels as shown in Fig. 1(b). Then we calculate mean values of Y, Cb, and Cr planes in the segmented fire regions of the original images. It is clear that, on the average, the fire pixels shows the characteristics that their Y colour value is greater than Cb colour value and Cr colour value is greater than the Cb colour value. Besides these two rules (Eq. (3) and (4)), since the flame region is generally the brightest region in the observed scene, the mean values of the three channels, in the overall image Y_{mean} , Cb_{mean} , and Cr_{mean} contain valuable information. For the flame region the value of the Y component is bigger than the mean Y component of the overall image while the value of Cb component is in general smaller than the mean Cb value of the overall image. Furthermore, the Cr component of the flame region is bigger than the mean Cr component. These observations which are verified over countless experiments with images containing fire regions are formulated as the following rule:

$$F(x, y) = \begin{cases} 1, & \text{if } Y(x, y) > Y_{\text{mean}}, Cb(x, y) < Cb_{\text{mean}}, Cr(x, y) > Cr_{\text{mean}} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $F(x, y)$ indicates that any pixel which satisfies condition given in Eq. (5) is labeled as fire pixel.

Fig. 2 shows the three channels for a representative image containing fire in more detail. The rule in (5) can be easily verified. It can easily be observed from the representative fire image (Fig. 2) that there is a significant difference between the Cb and Cr components of the flame pixels. The Cb component is predominantly “black” while the Cr component is predominantly “white”. This fact is formulated as a rule as follows:

$$F_t = \begin{cases} 1, & \text{if } |Cb(x, y) - Cr(x, y)| \geq \tau \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Where τ is a constant.

The value of τ is determined using a receiver operating characteristics (ROC) [9] analysis of Eq. (6) on an image set Consisting of 1000 images. Fig.1 shows a sample of digital colour video which has variety of images including ones with changing illumination and lighting. Furthermore, the images are selected so that fire-like coloured objects are also included in the set. For instance the Sun in the image that produces fire-like colour. There are some images in the set which do not contain any fire. The image set consists of random images collected from the internet. Images are from both indoor and outdoor environments. The ROC curve for the image set is given in Fig. 3 where hand segmented fire images are used in order to create the ROC curve. The rules (2) through (6) are applied to hand segmented fire

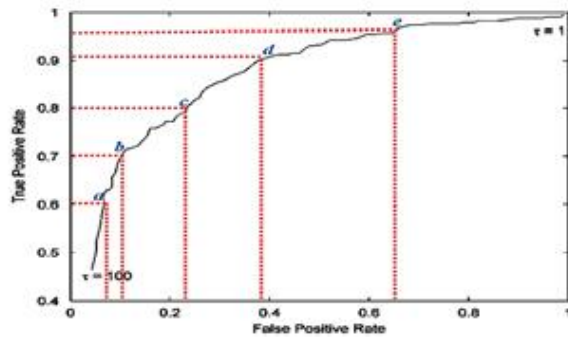


Fig. 3 : Receiver operating characteristics for τ .

images with different values of τ changing from 1 to 100. For each value of τ , we calculate corresponding true and false positive rates on the image set and tabulate it. The true positive is defined as the decision when an image contains a fire, and false positive is defined as the decision when an image contains no fire but classified as having fire. The ROC curve consists of 100 data points corresponding to different τ values and some of them are labeled in Fig. 3 with blue letters, i.e. a–e. For each point in the ROC curve there are three values; true positive rate, false positive rate, and τ . For instance, for the point labeled with a, the true positive rate is 60%, false positive rate is 6% and corresponding τ is 96. Using the ROC curve, different values of τ can be selected with respect to required true positive and false positive rates. Since fire detection systems should not miss any fire alarm, the value of τ should be selected so that systems true positive rate is high enough. It is clear from Fig. 6 that, high positive rate means that high false positive rate. Using this tradeoff, in our experiments the

value of τ is picked such that the detection rate is over 90% and false alarm rate is less than 40% (point d) which corresponds to $\tau \approx 40$.

In addition to the above rules a statistical analysis of chrominance information in flame pixels over a larger set of images is performed. For this purpose a set of 1000 images, containing fire at different resolutions are collected from the Internet. Samples from this set are shown in Fig. 7. The collected set of images has a wide range of illumination and camera effects. The fire regions in the 1000 images are manually segmented and the histogram of a total of 16,309,070 pixels is created in the Cb–Cr chrominance plane. Fig. 8 shows the distribution of flame pixels in Cb–Cr plane. The area containing flame pixels in Cb–Cr plane can be modeled using intersections of three polynomials denoted by $f_u(Cr)$, $f_l(Cr)$, and $f_d(Cr)$. The equations for the polynomials are derived using a least-square estimation technique [10]:

$$\begin{aligned} f_u(Cr) &= -2.6 \times 10^{-10} Cr^7 + 3.3 \times 10^{-7} Cr^6 - 1.7 \times 10^{-4} Cr^5 \\ &\quad + 5.16 \times 10^{-2} Cr^4 - 9.10 \times Cr^3 \\ &\quad + 9.60 \times 10^2 Cr^2 + 1.40 \times 10^6 \\ f_l(Cr) &= -6.77 \times 10^{-8} Cr^5 + 5.50 \times 10^{-5} Cr^4 \\ &\quad - 1.76 \times 10^{-2} Cr^3 + 2.78 Cr^2 \\ &\quad - 2.15 \times Cr + 10^2 Cr + 6.62 \times 10^3 \\ f_d(Cr) &= -1.81 \times 10^{-4} Cr^4 + 1.02 \times 10^{-1} Cr^3 \\ &\quad + 2.17 \times 10 Cr^2 + 2.05 \times Cr - 2.15 \times 10^3 Cr \\ &\quad + 7.29 \times 10^4 \end{aligned} \quad (7)$$

The region bounded by the three polynomials is depicted in Fig. 4.

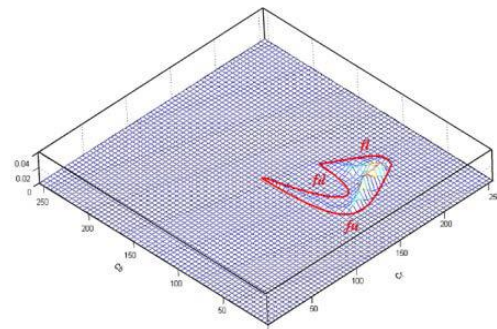


Fig.4: 3-D distribution of hand labeled flame pixels in Cb–Cr colour plane and three polynomials, $f_u(Cr)$, $f_l(Cr)$, and $f_d(Cr)$, bounding the flame region.

The boundaries of the region which correspond to the polynomials are shown in red. Once this region is obtained, it is easy to define another rule for classifying the flame pixel. We formulate this in Eq. (8) as follows:

$$F_{chr}(x, y) = \begin{cases} 1, & \text{if } Cb(x, y) \geq f_u(Cr(x, y)) \cap Cb(x, y) \leq f_d(Cr(x, y)) \cap Cb(x, y) \leq f_l(Cr(x, y)) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $F_{cbcr}(x, y)$ shows whether corresponding pixel at spatial location (x, y) falls into region defined by boundaries formulated in Eq. (7) with 1 indicating that it is included in this region and 0 indicating that it is not included and \wedge is the binary AND operator. With the derived set of rules in the YCbCr colour space given in Eq. (3)–(6) and (8), one can classify whether a given pixel is a flame pixel or not. The overall segmentation process is illustrated in Fig. 9 in a step-by-step manner. As can be seen from Fig. 9, each rule produces false alarms, but their overall combination produces the result which is convenient in identifying fire regions in corresponding colour image.

III. SMOKE DETECTION

A. Introduction

Each year, a large number of people die in fires throughout the world, and the real killer is not just the fire. When a fire disaster strikes, smoke will always come with the fire. The number of deaths caused by smoke is 50%-80% of the total number of deaths in the fire. Therefore, smoke is the main cause of deaths when fire occurs in buildings. In buildings, there are oftentimes a large number of plastic decorations, chemical fiber carpets, foam filler furniture, and similar objects. These things will, in the combustion process, produce large amounts of toxic gases and consume a large amount of oxygen. The main components of the smoke in the fire are carbon monoxide, carbon dioxide, hydrogen sulfide and so on, all being toxic gases. Because of these gases, smoke can kill both by poisoning and by suffocation. Therefore, smoke detection is very important for fire prevention, to avoid people from dying in the fires; therefore it is important for the world. Based on the characteristics of smoke, we want to learn how to detect it – this can help us prevent fires from occurring and avoiding people dying.

What is smoke? Smoke is a physical and chemical phenomenon. When substances burn, chemical changes will suddenly occur and particles as well as gases will be generated by transformation or decomposition of the substances. These particles and gases are what constitute the smoke. Smoke can have different colours, because of the different compositions it can have. Because of the impact of air, smoke has both irregular and diffuse features. Irregular, in that its shape is always changing, and diffuse, in that its area will grow larger and larger. As the area increases, the density will decrease more and more. As you see in figure 1, the smoke's edge is fuzzy, and the area of the smoke is becoming bigger and bigger. The colour of the smoke in figure 1 is gray-black – however, as the area increases, the colour of the smoke becomes lighter, since the density decreases. The

traditional methods of detecting smoke use techniques such as particle sampling, temperature sampling and relative humidity sampling – however, these techniques have proven inefficient and unreliable in many settings. Chen et al. state that this is due to many reasons, such as the fact that the sensors needed for the detection equipment has to be close to the source of the smoke – otherwise the sensors cannot detect the smoke effectively.

The traditional smoke detection methods are oftentimes hard to use in large places, such as warehouses and tunnels. Furthermore, they are difficult to use in areas with strong airflow, such as offshore drilling platforms [4, 12]. In addition, sensors always being in harsh environments risk being damaged or failing.

To avoid these situations, an alternative way, which has grown more popular, is video smoke detection. These methods use computer programs to analyze videos in more advanced ways, giving more flexibility and better responsiveness in detecting smoke. Because the smoke has both static feature (colour) as well as dynamic features (Irregularity, Diffuse feature and Direction), in this thesis, we propose to use these features to detect smoke. Smoke can have different colours depending on the substances combusted, but in this thesis, we focus particularly on black and gray smoke. As for the dynamic features, we use three characteristics of smoke: irregularity, diffusion and direction. Smoke is always moving around and cannot maintain its shape instead, it is changing all the time, seemingly without following any clear pattern. Smoke always keeps diffusion. The area of smoke will become larger and larger. In the easier case of smoke indoors, which is the main focus of this thesis and where wind is seldom a factor, the direction of the smoke is from the bottom, going upwards. Using these features, we hope to detect smoke effectively. Smoke detection can provide an early warning of fire, reducing economic losses and casualties.

B. HSV colour domain

We all know that different smoke colours will be produced when different fuels are burned. We can use this property to distinguish the real smoke. In this case, we need to choose a suitable way to detect the colours. After some comparisons, we decided to use HSV colour domain. HSV means hue saturation value. As can be seen in fig.5 the H parameter represents the colour information, and shows the position of the spectral colours; red, green and blue are separated by 120°. Complementary colour differ 180°. The S parameter represents saturation. This parameter ranges from 0 to 1, and it indicates the ratio between the selected colour

purity and the maximum purity of the colour. When $S = 0$, it is only gray. Finally, the V parameter represents the brightness of the colour, and also ranges from 0 to 1, with 1 being the brightest. HSV is an intuitive colour model to the user, and is much easier than the RGB colour space. We specify the colour angle H and let $V = S = 1$. Then, we can add black and white in different amounts to get the colour which we need. Adding black, the V parameter will decrease while the S parameter stays unchanged. Adding white, S will decrease and V will not change.

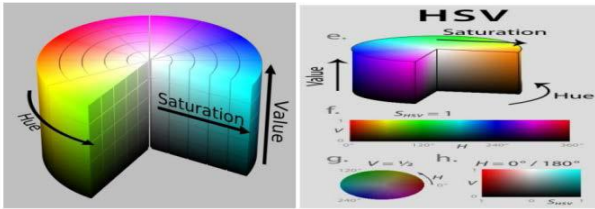


Fig.5. HSV the HSV colour model mapped to a cylinder

C. Gaussian Mixture Model (GMM)

The colour is an important feature of the smoke. Detecting colours which are similar with smoke colours, we can extract areas which might be smoke. In our thesis, we will choose to create a Gaussian Mixture Model (GMM) to detect the smoke colour pixel in HSV colour space. Moving target detection is very important. In the extraction of moving target detection, background is important for target recognition. Furthermore, modeling is an important part of background object extraction. However, due to illumination changes and other environmental effects, after the general method of modeling, the background is not very clear. But the Gaussian mixture model is one of the most successful methods for modeling. As shown in Figure 3, that is why we have chosen this method.

First, we must mention two things: background and foreground. Assuming the background is static, any moving object can be considered as foreground. Moving target detection is about taking out the foreground objects from the background in the image sequence. Moving target detection method is commonly accomplished in three ways: Optical flow method, inter-frame difference method and background subtraction method.

Problems of moving target detection are divided into two categories: the stationary camera and the moving camera. In the case of moving camera, the optical flow method is a more well-known solution of doing moving target detection. However, due to the complexity of optical flow, it is always difficult to calculate in real time. Because of this, we will use a Gaussian background model. Through background

modeling, we will separate the foreground and the background in a given image. In general, the foreground is the moving object, so as to achieve the purpose of moving target detection

This method uses the Gaussian mixture model to describe pixel processes. Based on the delay and variance of the Gaussian mixture model, we can determine which Gaussian distribution is corresponding to the background color. The pixels which do not fit the background distribution can be considered as foreground. Gaussian distribution is also called normal distribution. It was developed by the German mathematician Gauss in 1809. There are two important parameters for Gaussian distribution: the variance, denoted by μ^2 , and the mean, denoted by μ . Depending on a pixel value x , $p(x)$ is determined. Also, when $\mu=0$ $\sigma^2=1$ and , the distribution of X is a standard normal distribution. See equation

$$p(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (9)$$

Means probability density. x is a random variable μ . Is the mean of the Gaussian distribution, while σ is the variance? If a set of data matches the Gaussian distribution, then most of this data will be concentrated in the center of p within an interval of $[-2\sigma, 2\sigma]$, as shown in Fig.6

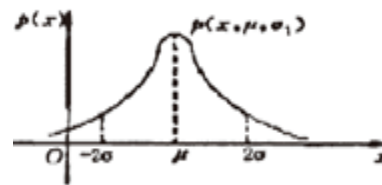


Fig. 6 : A sketch of Gaussian distribution

Usually, if there are many factors affecting the detection, a Gaussian distribution cannot be used to fully describe the actual background. We need multiple Gaussian models to describe the dynamic background. This means we need to create different Gaussian models for different situations. 6

In the Gaussian Mixture Model, we use multiple Gaussian models to represent each pixel. For a point (i,j) in a frame image, the observation value at time t is written as X_t . At a given point (i,j) a series of observation values are $\{X_1, X_2, \dots, X_t\}$. It can be seen as a statistical random process. We use the number of K Gaussian Mixture Model to simulate. With the probability distribution for the point (i,j) we can use the following equation to estimate.

$$P(X_t) = \sum_{k=1}^K w_{k,t} N(X_t, \mu_{k,t}, \Sigma_{k,t}) \quad (10)$$

$w_{k,t}$ is the weight of the k :th Gaussian mixture distribution at time t . N is the Gaussian probability density function $\mu_{k,t}$ is the mean value for the k :th Gaussian mixture distribution $\Sigma_{k,t}$ is the variance for the k :th Gaussian mixture distribution. This section will show the basic idea about Gaussian distribution Match. For a given point (i,j) , we use the value X_t to match with K Gaussian distributions. K is constant, taking a value from 3 to 5. We set one of the K Gaussian distributions to be N_k . If this Gaussian distribution N_k matches X_t , then we use the value of to update the parameters of this N_k . If none of the K Gaussian distributions match, we use new Gaussian distributions to replace the old one. The definition of Match can be shown like this. We arrange Gaussian distributions from big to small by the ratio of weight and variance (the ratio comes from w/σ). Then we choose a Gaussian distribution which has a value similar with X_t and $\mu_{j,t-1}$ as the matched Gaussian distribution, as you can see in equation 11:

$$M = (|X_t - \mu_{j,t-1}|^2) < \lambda^2 \sigma^2 \quad (11)$$

λ is a constant always set to 2.5. If we cannot find any Gaussian distribution match with the current pixel, then the smallest Gaussian distribution will be replaced by a new Gaussian distribution. In this new Gaussian distribution, the mean value is the current pixel value. The new Gaussian distribution will have larger variance and smaller weight. The formula for adjusting the weight is shown below:

$$w_{k,t} = (1 - \alpha) + \alpha M_{k,t} \quad (12)$$

α is the learning rate. If a Gaussian distribution is matched, the value of $M_{k,t}$ is 1. Otherwise, the value of $M_{k,t}$ is 0. For the Gaussian distribution which is not matched, α and μ are not changed. For the Gaussian distribution which is matched, the values are updated as follows:

$$\begin{aligned} \mu_t &= (1 - \beta)\mu_{t-1} + \beta X_t \\ \sigma_t^2 &= (1 - \beta)\sigma_{t-1}^2 + \beta(X_t - \mu_t)^T(X_t - \mu_t) \\ \beta &= \alpha N(X_t | \mu_t, \sigma_k) \end{aligned} \quad (13)$$

β is a learning rate which is used to adjust the current Gaussian distribution. If β is large, the degree of matching Gaussian distribution is better. α is also a learning rate, and it reflects the speed of current pixel merge into the background model. The basic idea of modeling is to extract the foreground from the current video frame. Let us summarize the modeling process Gaussian mixture model. First, we initialize several Gaussian models. Then, we initialize the parameters of the Gaussian model, followed by calculating the new

parameters that will be used later. After that, we process each pixel in each video frame, to see whether it matches a Gaussian model. If a pixel matches, then this pixel will be included in this Gaussian model, and then we will update the model under the new pixel value. If the pixel does not match, we build a new Gaussian model depending on the values of the pixel and the initialization parameters, and replace the most unlikely model in the original model. Finally, select the most likely models as the background model. We establish the Gaussian model for every pixel in the video frame, and then do Gaussian model match with these pixels – this is used to extract the foreground (which should be the smoke area). According to the background, we use the Gaussian mixture model to remove the background pixels (background pixel should be static target pixel) to get the smoke pixels. The Gaussian mixture model uses $K - a$ value ranging from 3 to 5 – Gaussian models to represent the characteristics of each pixel in video frame. After getting a new video frame, the Gaussian mixture model will be updated. The current video frame with each pixel point matches with the Gaussian mixture model. If successful, this point will be regarded as a background point. Otherwise, it is a foreground point. After that, we can find which pixels belong to the foreground. Last, we extract the foreground – and the foreground extracted can be considered as smoke area.

D. Dynamic Analysis

The attention of the human eye is easily caught by moving Objects. Smoke is gas, so it cannot maintain its shape over time rather; it is always changing and moving. As time passes, the smoke area will become larger and larger. The dynamic features of smoke include irregular shape, diffusion and the direction of movement. Therefore, we use the dynamic features of smoke in order to detect it. For the smoke color areas which are acquired from the GMM method, we determine whether they have these dynamic features. If these areas have the dynamic features of smoke, they can be considered as smoke. Otherwise, they will not. This will filter out false smoke areas. The false smoke areas are the areas which have color similar with smoke, but still s not real smoke. Therefore, these dynamic features of smoke can be useful for detecting smoke, and avoiding detection of false smoke. Using these dynamic features will greatly improve the accuracy of the detection method. For the irregularity, we set a threshold as standard. Then, we compare the area of the extracted part with its circumference. After that, we will get a ratio between circumference and area. Then, we compare this ratio with the threshold value to determine whether or not the extracted area is irregular. The rule can be shown like this: If ratio threshold, then it might be smoke, Else, it is not smoke. For the diffusion

feature, we use the growth rate to determine whether the extracted area has this feature. Due to the smoke diffusion process, the smoke area will always increase in an image sequence. The growth rate is how much the extracted area grows during a given period of time. If the growth rate is larger than a certain threshold, we can consider that extracted area to have the diffusion feature. Otherwise, it does not. The rule can be shown like this:

If growth rate \geq threshold, then it might be smoke,
Else, it is not smoke.

As for the direction of movement, in the case of no wind, the direction of the smoke is from bottom to top. We can use this feature to detect the smoke areas. In this way, we can filter away the areas which have the smoke color, but actually is not smoke. We use the max value of the binary image for each frame to detect if the direction is going up. The rule can be shown like this: If the value of the second frame $>$ the value of the first frame, then the direction is going up. Otherwise, it is not.

E. Smoke features

Smoke will be produced when different materials are burning; this smoke is toxic if the combustion is incomplete. Different materials will generate different colors of smoke. Furthermore, smoke has some dynamic features. Smoke always moves around, and in the beginning, it is difficult to say what direction the smoke has, but as time passes, the final direction of the smoke will be upward. The shape of the smoke changes without any clear rules or patterns every moment this, along with the fact that the edge of the smoke is fuzzy, tells us that the shape of smoke is irregular. The area of the smoke is not very big at the source of the smoke, but as time passes, the smoke will diffuse, and the area of the smoke will grow larger, and at the same time, the density of the smoke will decrease.

F. Static Detection

In our thesis, we create a Gaussian mixture model to detect smoke color pixels. Removing the background and extracting the smoke is a very important part in detecting smoke. Our approach contains two steps in static detection. The prerequisites for understanding are described in the background section. This section will show how we use these methods.

Step 1: Build Gaussian mixture model:

First, we will initialize a Gaussian mixture model $u_i(x_i|\mu_i,\Sigma_i)$, (i stands for the number of the Gaussian model). x_i is the observation value μ_i is the mean value. Σ_i is the variance. Then, we use the first pixel value of the object which is being observed as the mean value.

Initialize a large variance and a small weight. Then we use these three parameters mean value, variance and weight to get an initialized Gaussian model. After that, we use the next pixel to match with the Gaussian model which has been initialized. If the existing Gaussian model does not match, we add a new Gaussian model. Repeat the above steps.

Step 2 : Match the Gaussian distribution:

We arranged Gaussian distributions from big to small by the ratio of the weight and variance. Then we updated the priority of the Gaussian model, through the updated weights w_i and variance σ_i . The update priority formula is shown in eq.14.

$$p_i = w_i/\sigma_i \quad (14)$$

The color of the current pixel and the i :th Gaussian distribution will match by the update priority until it finds a matching distribution. The matching condition for the i :th distribution is:

$$|X_t - \mu_{j,t-1}| < 2.5\sigma_i \quad (15)$$

If there are matching models, then we will update the mean value μ and variance σ^2 , as described by equation 16, 17 and 18:

$$\mu = (1 - \beta)\mu + \beta x \quad (16)$$

$$\sigma^2 = (1 - \beta)\sigma^2 + \beta(X_t - \mu_t)^T(X_t - \mu_t) \quad (17)$$

$$\beta = \alpha\eta(x|\mu, \sigma) \quad (18)$$

Then we will update the weight, as by equation 19:

$$w_i = (1 - \alpha)w_i + \alpha M_i \quad (19)$$

In eq.19, M_i means the i :th Gaussian distribution is matched, If a Gaussian distribution is matched, the value of M_i is 1. Otherwise, the value of is 0. The formula (19) also means that the weight of the Gaussian model which matches with the current pixel should increase, otherwise the weight will decrease. When the Gaussian model reaches the upper limit, if it still has not found any Gaussian models matching the current pixel, then the Gaussian model which has the east priority will be removed. We will make a new Gaussian model. In the new Gaussian distribution, a smaller weight and larger variance will be given, and then update all the weight of Gaussian model. When we create Gaussian mixture model, we need some samples to test. This test can be used to detect whether the pixels in the image of the Gaussian mixture model have been created to extract the image area with smoke color, see Figure 6. During this step, we can remove some things which might disturb the smoke in the background. Some objects have different colors which, when compared with the smoke, can be eliminated, e.g. the wall, the metal pail, the floor, the flame. But sometimes the color of disruptors are

similar with smoke, for example, the people's moving shadow on the wall is similar with smoke, it will still be distinguishable from the smoke. There are some features of the smoke which can be used to distinguish the things which are similar with the smoke. The next method will show how to do it.

G. Dynamic analysis

After the previous methods, we can remove the things which do not have the smoke color in the background. But if there are some things which might be confused with the smoke, such as shadows, people who are moving and so on, and then we can use the following method to analyze these things. We will focus on three dynamic characteristics of smoke: irregularity, diffusion and direction.

Step 1 : Irregularity of smoke:

Because of the air flowing, smoke is constantly changing shape. Determining the shape of the smoke is a difficult task. Therefore, we use two parameters which are the perimeter and the smoke area. We compare the perimeter with the smoke area to get a ratio. Then, we compare this ratio with a certain threshold. If the ratio is greater than this threshold, this area could be the smoke area. Otherwise, it is discarded. The equation is shown below. In equation 14, the smoke area is defined as the total pixel area of suspected smoke :

$$\frac{\text{The perimeter of the smoke area}}{\text{The smoke area}} \geq \text{Threshold} \quad (20)$$

The parameter perimeter means the sums of circumferences of smoke regions segmented. The parameter smoke area means the sum of smoke pixel extraction. Threshold is used to distinguish the other similar smoke area.

Step 2 : Diffusion of smoke:

Due to the diffusion of smoke, the size of smoke will continue to increase. Therefore, we calculate the period of time for the growth rate of the extracted area in order to determine the diffusion of the smoke. In digital images, the area of the smoke (p) can be represented by the number of pixels. The time interval can be expressed as the number of frames. This is used in eq. 21:

$$\nabla A_{di} = \frac{d_p}{d_t} = \frac{A_{i-k} - A_i}{t_{i,k} - t_i} \quad (21)$$

A_i represents the smoke area at time i. In video processing, the smoke area pixel quality and time intervals can be replaced by the interval number of frames. As you can see in eq. 22:

$$\nabla A_{di} = \frac{d_p}{d_t} = \frac{p_{i-k} - p_i}{(i+k) - i} \quad (22)$$

p_i is the total number of pixels in the smoke area in the i:th frame of the image sequence. $(i+k) - i$ is d_t which represent the k:th frame. ∇A_{di} represents the rate of change in number of pixels extracted from the smoke area, from the i:th frame to the $(i+k)$:th frame. To avoid temporary inconsistencies, due to factors such as airflow, we calculate the average growth rate during 10 frame period. This can improve the detection accuracy. As can be seen in eq. 23:

$$\overline{\nabla A_{di}} = \frac{1}{n} \sum \nabla A_{di} \quad (23)$$

$\overline{\nabla A_{di}}$ is the average growth rate. If the average growth rate is greater than the threshold value, this area is considered smoke area. Otherwise, it is not. The threshold will be decided by the experimental data. After this step, the dynamic things which have color similar with real smoke can be removed.

Step 3: Direction of movement of the smoke:

For the human eye, moving objects are very clear, and easily attract attention. For the extracted areas which we get after applying the GMM method, we will judge its movement direction. We use the max value of binary image for each frame to detect if the direction is going up. According to the maximum value of binary image, we use the value of the first frame of the area which has been detected as the standard. Then we let the next frame value of this area to compare with this standard. If the next value is larger than the first value, we can consider this area to be moving from bottom to up and regard it as smoke area. The processing will continue. But if the next value is equal to the first value, then the extracted area is moving sideways; and if the next value is smaller than the first value, then the extracted area is moving downwards. In either of these two cases, the processing will stop. The rule can be show like this:

$$\begin{aligned} f2 > f1, & \text{then continue} \\ f2 \leq f1, & \text{stop.} \end{aligned}$$

$f1$ means the value of first frame, $f2$ means the value of next frame. After these three steps, most of the interferences can be eliminated. But there is still an interference which cannot be eliminated. That is the shadow of the smoke. It has the same dynamic features with the smoke and it can be shown after these steps. If this happens, we will use symmetry to remove the shadow. If we detect two similar areas then we will remove the left one, since the left side of the screen always displays the shadow of the smoke in our video. So we set the default that smoke area on the left is

smoke shadow. This area will be removed and not be shown.

IV. CONCLUSION

In this paper, we propose a method for smoke detection for indoor as well as for outdoor video sequences. It works well in videos captured by surveillance CCTV cameras monitoring any type of fires. Our proposed method is composed of three steps. The first step is to Detection of fire and smoke pixels is at first achieved by means of a motion detection algorithm. The second step is separation of smoke and fire pixels using colour information (within appropriate spaces, specifically chosen in order to enhance specific chromatic feature). The final step is to a pixel selection based on the dynamics of the area is carried out in order to reduce false detection. The outputs of the three parallel algorithms are eventually fused by means of a Multi-Layers Perceptron (MLP).

V. REFERENCES

- [1] T. Chen, P. Wu, Y. Chiou, An early fire-detection method based on image processing, in: *Proceedings of IEEE International on Image Processing*, 2004, pp. 1707–1710.
- [2] B.U. To˘reyin, Y. Dedeog˘lu, A.E. Cetin, Flame detection in video using hidden Markov models, in: *Proceedings of IEEE International Conference on ImageProcessing*, 2005, pp. 1230–1233.
- [3] B.U. To˘reyin, Y. Dedeog˘lu, U. Gu˘ndu˘kbay, A.E. Cetin, Computer vision based method for real-time fire and flame detection, *Pattern Recognition Lett.* 27 (1) (2006) 49–58.
- [4] T. Celik, H. Demirel, H. Ozkaramanli, Automatic fire detection in video sequences, in: *Proceedings of European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, September 2006.
- [5] W. Krull, I. Willms, R.R. Zakrzewski, M. Sadok, J. Shirer, B. Zeliff, Design and test methods for a video-based cargo fire verification system for commercial aircraft, *Fire Saf. J.* 41 (4) (2006) 290–300.
- [6] G. Marbach, M. Loepfe, T. Brupbacher, An image processing technique for fire detection in video images, *Fire Saf. J.* 41 (4) (2006) 285–289.
- [7] Wen-Bing Hong, Jim-Wen Peng, Chih-Yuan Chen, A new image-based real-time flame detection method using colour analysis, in: *Proceedings of IEEE Networking, Sensing and Control, ICNSC*, 2005, pp. 100–105.
- [8] C.A. Poynton, *A Technical Introduction to Digital Video*, Wiley, New York, 1996.
- [9] D.M. Green, J.M. Swets, *Signal Detection Theory and Psychophysics*, Wiley, New York, 1966. [
- [10] J.H. Mathews, K.D. Fink, *Numerical Methods using MATLAB*, Prentice-Hall, Englewood Cliffs, NJ, 1999.
- [11] Wildland Fire Operations Research Group, Retrieved August 11, 2006, from <http://fire.feric.ca/index.htm>

