Fire Detection Using Image Processing

V.N.S. Sheshma Kumari

Abstract: Conventional fire detection systems use physical sensors to detect fire. Chemical properties of particles in the air are acquired by sensors and are used by conventional fire detection systems to raise an alarm. However, this can also cause false alarms; for example, a person smoking in a room may trigger a typical fire alarm system.

In parallel to this the availability of sensors is low and the maintenance of these sensors is high. And also to get more efficient output the density of sensors should be high. By increasing density of sensors over a particular area the establishment cost of fire detection system should increases though the cost of sensors is high. In order to manage false alarms of conventional fire detection systems, a new computer vision-based fire detection algorithm is proposed.

The proposed fire detection algorithm consists of two main parts: fire color modeling and motion detection. The algorithm can be used in parallel with conventional fire detection systems to reduce false alarms. It can also be deployed as a stand-alone system to detect fire by using video frames acquired through a video acquisition device.

This fire detection system can be used in determining intensity levels in boiler section in industries in order to regulate the flow of fuel gases into the boilers so as to control the heat in the boilers. It also can be used in home security fire alarms, sort circuit alarms in trains and also to monitor fire accidents in thick forests by enhancing the proposed fire detection system.

Keywords: Computer- vision, Fire Accidents, Fire color modeling, Fire Detection, Motion detection, Sensors, Video frames**.**

1. Introduction

Fire detection systems are among the most important components in surveillance systems used to monitor buildings and the environment. As part of an early warning mechanism, it is preferable that the system has the capacity to report the earliest stage of a fire. Currently, almost all fire detection systems use built-in sensors that depend primarily on the reliability and the positional distribution of the sensors. It is essential that these sensors are distributed densely for a high precision fire detection system. In a sensor-based fire detection system for an outdoor environment, coverage of large areas is impractical due to the necessity of a regular distribution of sensors in close proximity.

Due to rapid developments in digital camera technology and video processing techniques, there is a major trend to replace conventional fire detection methods with computer vision based systems. In general, computer vision-based fire detection systems employ three major stages: fire pixel classification, moving object segmentation, and analysis of the candidate regions. This analysis is usually based on two figures: the shape of the region and the temporal changes of the region. The performance depends critically on the effectiveness of the fire pixel classifier which generates seed areas that the rest of the system will exercise. The fire pixel classifier is thus required to have a very high detection rate and preferably, allow false alarm rate. There exist few algorithms which directly deal with the fire pixel classification in the literature.

A good color model for fire modeling and robust moving pixel segmentation are essential because of their critical role in computer vision-based fire detection systems. In this paper, we propose an algorithm that models the fire pixels using the CIE *L*a*b** color space.

The motivation for using CIE *L*a*b**color space is because it is perceptually uniform color space, thus making it possible to represent color information of fire better than other color spaces. The moving pixels are detected by applying a background subtraction algorithm together with a frame differencing algorithm on the frame buffer filled with consecutive frames of input video to separate the moving pixels from non-moving pixels. The moving pixels which are also detected as a fire pixel are further analyzed in consecutive frames to raise a fire alarm.

2. Introduction to CIE Lab Colour Model

Color space defined by the CIE, based on one channel for Luminance (lightness) (L) and two color channels (a and b). CIE XYZ is an absolute color space (not device dependent). Each visible color has non-negative coordinates X,Y,Z.

CIE xyz, the horseshoe diagram as shown below, is a perspective projection of XYZ coordinates onto a plane xy. The luminance is missing.CIE Lab is a nonlinear transformation of XYZ into coordinates L*,a*,b*.

One problem with the XYZ color system, is that colorimetric distances between the individual colors do not correspond to perceived color differences. In this model, the color differences which you perceive correspond to distances when measured color diametrically.

This color space is better suited to many digital image manipulations than the RGB space, which is typically used in image editing programs. For example, the Lab space is useful for sharpening images and the removing artifacts in JPEG images or in images from digital cameras and scanners.

CIE XYZ is an absolute color space (not device dependent). Each visible color has non-negative coordinates X,Y,Z. CIE xyY, the horseshoe diagram as shown below, is a perspective projection of XYZ coordinates onto a plane xy. The luminance is missing. CIE Lab is a nonlinear transformation of XYZ into coordinates L*,a*,b*

The CIE Lab color space was intended for equal perceptual differences for equal changes in the coordinates L*,a* and b*. Color differences delta E are defined as Euclidian distances in CIE Lab. This document shows color charts in CIE Lab for several RGB color spaces

The gamut triangle in xy Y has to be replaced by a representation of a color cube with

corners R,G,B and Y(yellow),C,M. The gray axis is at a*=b*=0. Therefore the area is confined by a distorted hexagon.

2.1 RGB TO CIE L*a*b* Color Space Conversion

The first stage in our algorithm is the conversion from RGB to CIE L*a*b* color space. Most of the existing CCTV video cameras provide output in RGB color space, but there are also other color spaces used for data output representation. The conversion from any color space representation to CIE L*a*b*color space is straightforward [10]. Given RGB

data, the conversion to CIE L*a*b* color space is formulated as follows

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

$$L^* = \begin{cases} 116 \times (Y/Y_n)^{1/3} - 16, & \text{if } (Y/Y_n) > 0.008856, \\ 903.3 \times (Y/Y_n), & \text{otherwise,} \end{cases}$$

$$a^* = 500 \times \left(f(X/X_n) - f(Y/Y_n) \right),$$

$$b^* = 200 \times \left(f(Y/Y_n) - f(Z/Z_n) \right),$$

$$f(t) = \begin{cases} t^{1/3}, & \text{if } t > 0.008856, \\ 7.787 \times t + 16/116, & \text{otherwise,} \end{cases}$$

Where Xn, Yn, and Zn are the tri-stimulus values of the reference color white. The data range of RGB color channels is between0 and 255 for 8-bit data representation. Meanwhile, the data ranges of L*, a*, and b* components are [0, 100], [–110, 110], and [–110, 110], respectively where *Xn*, *Yn*, and *Zn* are the tri-stimulus values of the reference color white. The data range of RGB color channels is between 0 and 255 for 8-bit data representation. Meanwhile, the data ranges of *L**, *a**, and *b** components are [0, 100], [–110, 110], and [–110, 110], respectively.

2.2 Color Modelling for Fire Detection

A fire in an image can be described by using its visual properties. These visual properties can be expressed using simple mathematical formulations. In Fig. 2, we show sample images which contain fire and their CIE *L*a*b** color channels (*L**, *a**, *b**). Figure 2 gives some clues about the way CIE *L*a*b** color channel values characterize fire pixels. Using such visual properties, we develop rules to detect fire using CIE *L*a*b** color space. The range of fire color can be defined as an interval of color values between red and yellow. Since the color of fire is generally close to red and has high illumination, we can use this property to define measures to detect the existence of fire in an image.

For a given image in CIE *L*a*b** color space, the following statistical measures for each color channel are defined as

$$L_m^* = \frac{1}{N} \sum_x \sum_y L^*(x, y),$$

$$a_m^* = \frac{1}{N} \sum_x \sum_y a^*(x, y),$$

$$b_m^* = \frac{1}{N} \sum_x \sum_y b^*(x, y), \qquad (1)$$

where **Lm* , **ma* , and **mb* are a collection of average values of the *L**, *a**, and *b** color channels, respectively; *N* is the total number of pixels in the image; and (*x, y*) is spatial pixel location in an imaging grid. The numeric color responses *L**, *a**, and *b** are normalized to [0, 1]. It is assumed that the fire in an image has the brightest image region and is near to the color red. Thus, the following rules can be used to define a fire pixel:

$$R1(x, y) = \begin{cases} 1, & \text{if } L^*(x, y) \geq L_m^*, \\ 0, & \text{otherwise,} \end{cases} \qquad (3)$$

$$R2(x, y) = \begin{cases} 1, & \text{if } a^*(x, y) \geq a_m^*, \\ 0, & \text{otherwise,} \end{cases} \qquad (4)$$

$$R3(x, y) = \begin{cases} 1, & \text{if } b^*(x, y) \geq b_m^*, \\ 0, & \text{otherwise,} \end{cases} \qquad (5)$$

$$R4(x, y) = \begin{cases} 1, & \text{if } b^*(x, y) \geq a^*(x, y), \\ 0, & \text{otherwise,} \end{cases} \qquad (6)$$

where *R*1, *R*2, *R*3, and *R*4 are binary images which represent the existence of fire in a spatial pixel location (*x, y*) by 1 and the non-existence of fire by 0. *R*1(*x, y*), *R*2(*x, y*), and *R*3(*x, y*) are calculated from global properties of the input image. $R4(x, y)$ represents the color information of fire; for example, fire has a reddish color. Figure 3 shows sample images from Fig. 2(a), and binary images created using (3)-(6). Figure 3(f) shows a combination of these binary images with the binary AND operator. Figure 3(g) displays the segmented fire image. In order to find the correlation between L^*, a^*, and b^* values of fire pixels, the following strategy was applied. A set of 500 RGB images was collected from the Internet. Then, each image was manually segmented to identify all fire regions. Segmented fire regions are converted to L^*, a^*, and b^* color space. A histogram of fire pixels is created for each of the 3 different color planes, that is, (L^*-a^*), (L^*-b^*), and (a^*-b^*).c

Figure 2.1: Sample RGB images containing fire and their CIE $L^*a^*b^*$ color channels

For visualization purposes, responses in different color channels are normalized into interval [0, 1].

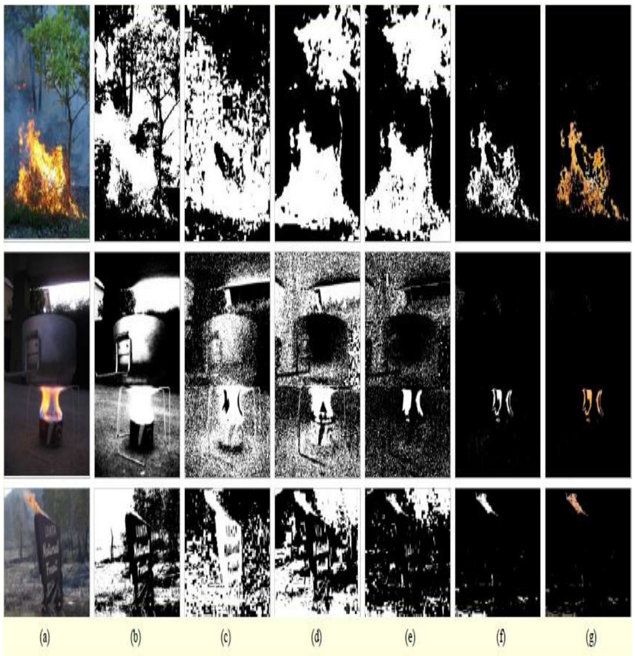

Figure 2.2: Processing images

3. Image Segmentation

Image segmentation is the first step in image analysis and pattern recognition. It is a critical and essential component of image analysis system, is one of the most difficult tasks in image processing, and determines the quality of the final result of analysis. Image segmentation is the process of dividing an image into different regions such that each region is homogeneous.

Image segmentation methods can be categorized as follows (this is not an exhaustive list):

- **Histogram thresholding:** assumes that images are composed of regions with different gray (or color) ranges, and separates it into a number of peaks, each corresponding to one region.
- **Edge-based approaches:** use edge detection operators such as Sable, Laplacian for example. Resulting regions may not be connected, hence edges need to be joined.

- **Region-based approaches:** based on similarity of regional image data. Some of the more widely used approaches in this category are: Thresholding, Clustering, Region growing, Splitting and merging.
- **Hybrid:** consider both edges and regions.

The project is done using Image Segmentation by Clustering. It is based on Color image segmentation using Mahalanobis distance. Euclidean distance is also used for comparing between the quality of segmentation between the Mahalanobis and Euclidean distance.

3.1 Image Segmentation by Clustering

Clustering is a classification technique. Given a vector of N measurements describing each pixel or group of pixels (i.e., region) in an image, a similarity of the measurement vectors and therefore their clustering in the N-dimensional measurement space implies similarity of the corresponding pixels or pixel groups. Therefore, clustering in measurement space may be an indicator of similarity of image regions, and may be used for segmentation purposes.

The vector of measurements describes some useful image feature and thus is also known as a feature vector. Similarity between image regions or pixels implies clustering (small separation distances) in the feature space. Clustering methods were some of the earliest data segmentation techniques to be developed.

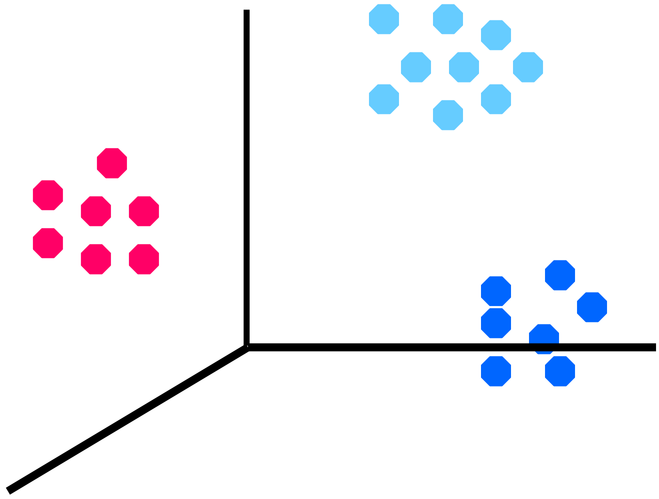

Figure 3.1 : Similar data points grouped together into clusters

4. Background Subtraction

In many applications, objects appear on a background which is very largely stable. The standard example is detecting parts on a conveyor belt. Another example is counting motor cars in an overhead view of a road the road itself is pretty stable in appearance. Another, less obvious, example is in human computer interaction. Quite commonly, a camera is fixed (say, on top of a monitor) and views a room. Pretty much anything in the view that doesn't look like the room is interesting. In these kinds of applications, a useful segmentation can often be obtained by subtracting an estimate of the appearance of the background from the image, and looking for large absolute values in the result. The main issue is obtaining a good estimate of the background. One method is simply to take a picture. This approach works rather poorly, because the background typically changes slowly over time. For example, the road may get more shiny as it rains and less when the weather driesup; people may move books and furniture around in the room, etc.

An alternative which usually works quite well is to

estimate the value of back-
ground pixels using a moving average. In this approach, we estimate the value of a particular background pixel as a weighted average of the previous values. Typically, pixels in the very distant past should be weighted at zero, and the weights increase smoothly. Ideally, the moving average should track the changes in the background, meaning that if the weather changes very quickly (or the book mover is frenetic) relatively few pixels should have non-zero weights, and if changes are slow, the number of past pixels with non-zero weights should increase.

This yields algorithm for those who have read the filters chapter, this is a filter that smooths a function of time, and we would like it to suppress frequencies that are larger than the typical frequency of change in the background and pass those that are at or below that frequency.

5. Proposed System

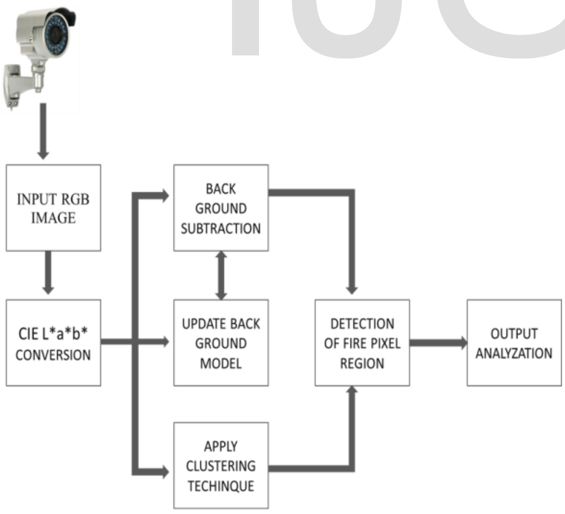

Figure 5.1: Block Diagram of Fire Detection System Using Image Processing

6. Conclusion

In this paper, a new image-based real-time fire detection method was proposed which is based on computer vision techniques. The proposed method consists of three main stages: fire pixel detection using color, moving pixel detection, and analyzing fire-colored moving pixels in consecutive frames to raise an alarm. The proposed fire color model achieves a detection rate of 99.88% on the ten tested video sequences with diverse imaging conditions. Furthermore, the experiments on benchmark fire video databases show that the proposed method achieves comparable performance with respect to the state-of-the-art fire detection method.

References

[1] T. Chen, P. Wu, and Y. Chiou, “An Early Fire-Detection Method Based on Image Processing,” Proc. IEEE Int. Image Process.,2004, pp. 1707-1710.

[2] B.U. Toreyin, Y. Dedeoglu, and A.E. Cetin, “Flame Detection inVideo Using Hidden Markov Models,” Proc. IEEE Int. Conf. Image Process., 2005, pp. 1230-1233, 2005.

[3] B.U. Toreyin, Y. Dedeoglu, and A.E. Cetin, “Computer Vision Based Method for Real-Time Fire and Flame Detection,” *Pattern Recognition Lett., vol. 27, no. 1, 2006, pp. 49-58.*

[4] T. Celik et al., “Fire Detection Using Statistical Color Model in Video Sequences,” J. Visual Commun. Image Representation, vol.18, no. 2, Apr 2007, pp. 176-185.

[5] T. Celik, H. Demirel, and H. Ozkaramanli, “Automatic Fire Detection in Video Sequences,” Proc. European Signal Process.Conf., Florence, Italy, Sept. 2006.

[6] W. Krüll et al., "Design and Test Methods for a Video-BasedCargo Fire Verification System for Commercial Aircraft," FireSafety J., vol. 41, no. 4, 2006, pp. 290-300.

[7] G. Marbach, M. Loepfe, and T. Brupbacher, "An Image Processing Technique for Fire Detection in Video Images," Fire Safety J., vol. 41, no. 4, 2006, pp. 285-289.

[8] W.-B. Horng, J.-W. Peng, and C.-Y. Chen, "A New Image-Based Real-Time Flame Detection Method Using Color Analysis," Proc. IEEE Networking, Sensing Control, 2005, pp. 100-105.

[9] W. Phillips III, M. Shah, and N. da Vitoria Lobo, "Flame Recognition in Video," Proc. 5th Workshop Appl. Computer

Author Profile

Nooka Sai Sheshma Kumari received the M.Tech degree in Electronics and Communication Engineering from Andhra University in 2016.