國 立 交 通 大 學

電機與控制學系

碩士論文

基於區域與全域分析之影像式火焰偵測系統

Video Fire Detection System Based on Local and Global Analysis

研究生:許廷維

指導教授:周志成 博士

中華民國 一百零一年 十 月

基於區域與全域分析之影像式火焰偵測系統

Video Fire Detection System Based on Local and Global Analysis

研 究 生：許廷維　　Student：Ting-Wei Hsu

指導教授：周志成 博士　　Advisor：Dr. Chih-Cheng Jou

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

September 2012

Hsinchu, Taiwan, Republic of China

中華民國 一百零一 年 十 月

基於區域與全域分析之影像式火焰偵測系統

學生:許廷維　　指導教授:周志成 博士

國立交通大學電機與控制工程研究所

摘要

近年來，基於影像式的火焰偵測技術在智慧型監控系統中受到廣泛的重視與研究。然而，在影像式偵測系統中有許多事件的干擾造成系統判斷上的困難，例如閃爍的紅光或紅色移動物體，建立一個穩定且有效率的火焰偵測系統能仍然是一個有趣且困難的挑戰。本篇論文中利用了區塊式的特徵擷取方式，可利於獲得區域內資訊並減少運算資料量。火焰的區域特徵模擬火焰物體的細部特性，分別為火焰顏色分析、火源不動性、混亂分析，三種特徵都有足夠的偵測率且可濾除不同的誤報。配合全域特徵分析火焰的整體特性，如質地分析、火焰面積分析，兩項特徵可幫助通過區域分析的鄰近候選區塊進一步的驗證，使得誤報率降低至理想的狀況。實驗結果顯示本篇論文提出的火焰偵測系統在各種不同的環境條件下擁有良好偵測率以及低誤報率，證明在實際防災應用上的穩健性及可靠性。整個系統所使用的演算法採取運算量低的計算方式，使得此系統可應用在更多的硬體設備上，提高此系統在智慧型監控系統被廣泛使用的可能性。

Video Fire Detection System Based on Local and Global Analysis

Student: Ting-Wei Hsu Advisor: Dr. Chih-Cheng Jou

Department of Electrical and Control Engineering

National Chiao Tung University

Abstract

In recent years, visual-based fire detection technology in intelligent surveillance systems has received wide attention and research. However, many disturbances can cause problems on a visual-based detection system, such as twinkling light or red moving objects. Establishing a stable and effective fire detection system is still a difficult challenge. This thesis uses the block-based feature extraction method, which easily analyzes local information in the region and reduces the computing data. Local feature of fire block are extracted from the detailed characteristics of fire objects, which are fire color analysis, fire source immobility, and disorder analysis. All three features have great detection rate and filter out different false-positive cases. Analyzing global features with textures and fire area of local candidates selected by local feature analysis further reduces false alarms of the proposed system.

Experimental results show that the fire detection system proposed in this thesis has high detection rate and low false alarm rate under various environments, and prove the reliability and stability in real fire-safety applications. The proposed system is composed of algorithms with low computation, so the system has high possibility for wide use by intelligent monitoring applications based on embedded devices.

致謝

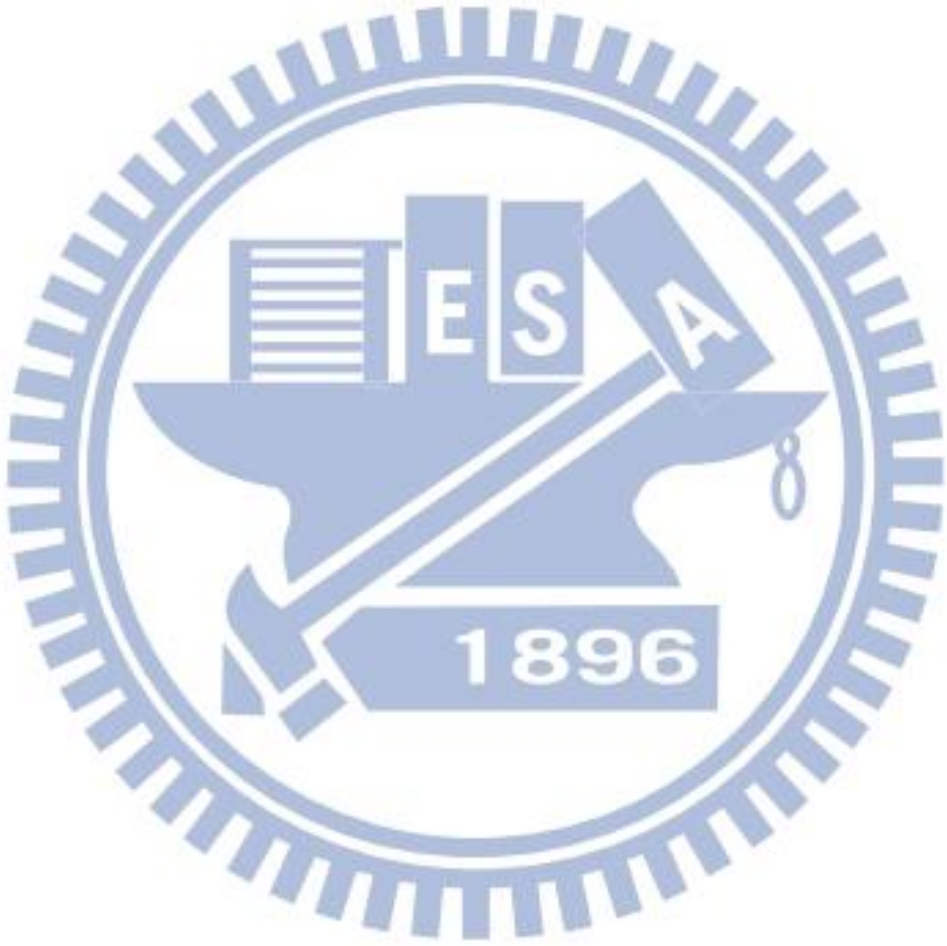
本論文的完成，首先要感謝指導教授周志成博士與林進燈博士這兩年來的悉心指導，讓我學習到許多寶貴的知識，在學業及研究方法上也受益良多。另外也要感謝口試委員們的建議與指教，使得本論文更為完整。

其次，感謝超視覺實驗室的大家長鶴章，還有剛維、建霆、肇廷、東霖、子貴、勝智學長以及 Linda 學姐，提供許多寶貴的意見與耐心的指導。良成、庭伊及榮宗同學的相互砥礪，及所有學長、學弟們在研究過程中所給我的鼓勵與協助。尤其是肇廷學長，在研究方法以及論文實驗上給予我相當多的幫助與建議，讓我獲益良多。

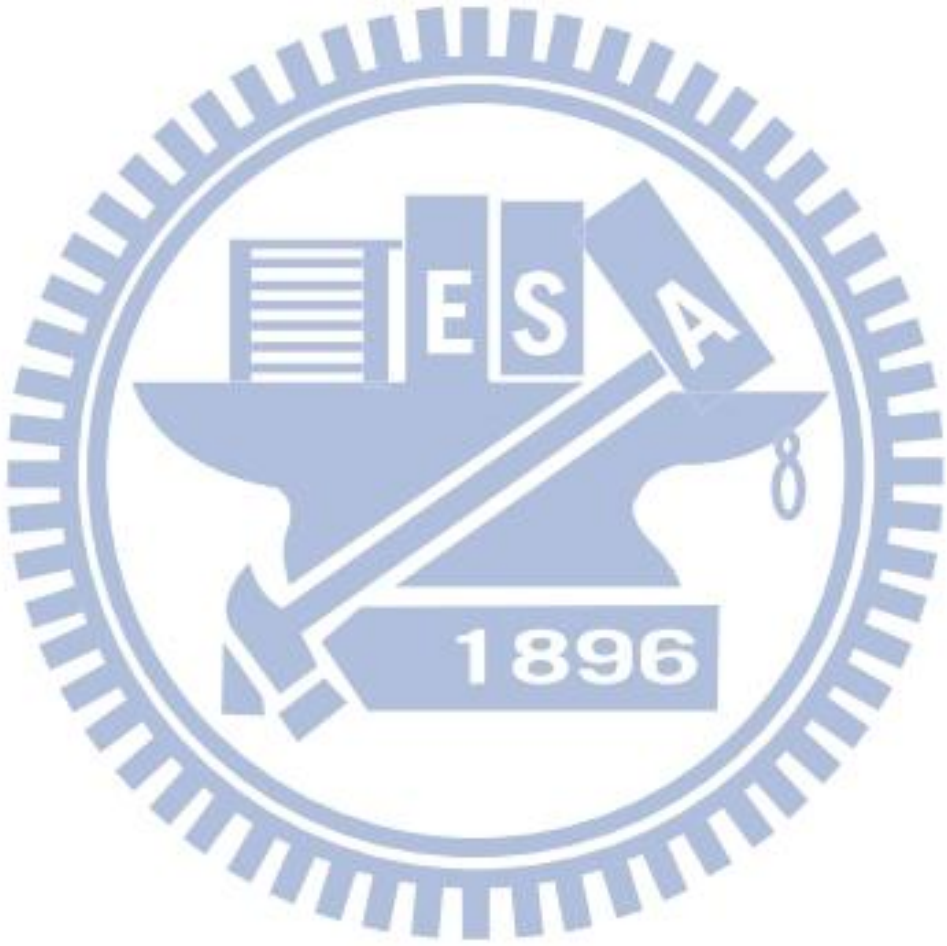
感謝我的父母親對我的教育與栽培，並給予我精神及物質上的一切支援，使我能安心地致力於學業。此外也感謝兩個姐姐及女友對我不斷的關心與鼓勵。

謹以本論文獻給我的家人女友及所有關心我的師長與朋友們。

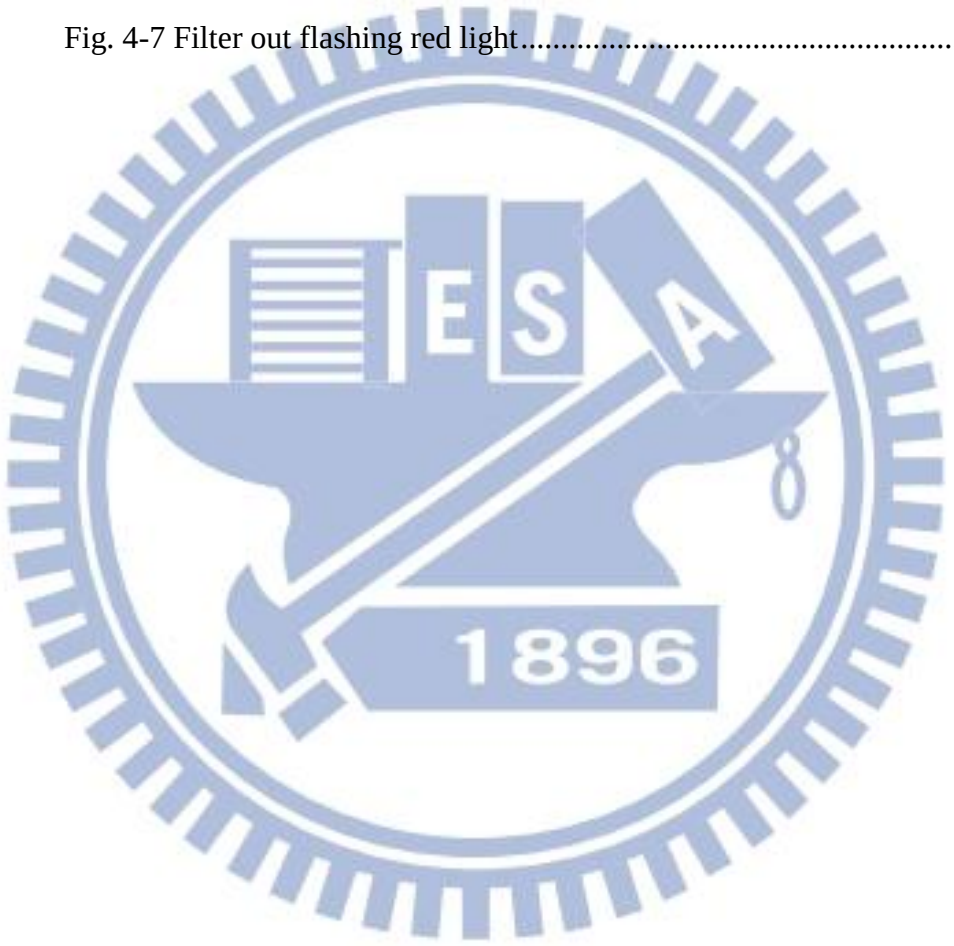
Content



List of Tables



List of Figures



Chapter 1
Introduction

1.1 Motivation

In last few years, there were average 3189.7 fire accidents per year according to the statistic report from the National Fire Administration. The number of dead and injured people was nearly 500 and the property loss was about 1.45 billion NT dollars each year. If the fire accident could be found much earlier, it is more likely to reduce the loss of life.

The process of fire development mostly divided into four periods: Ignition, Fire Growth Period, Fully Developed Period and Decay Period as shown in Fig.1-1.

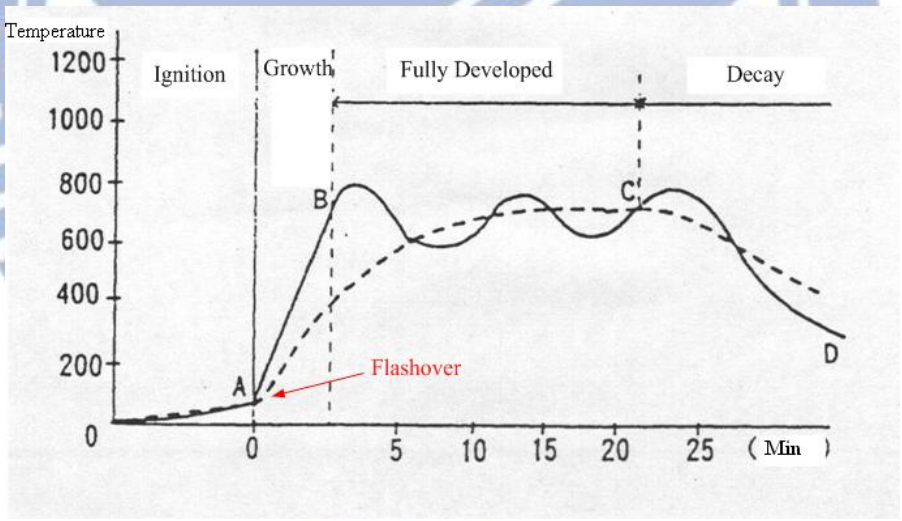


Fig. 1-1 Processing diagram of fire development

Fire was born in ignition but it can not make serious damage in this period. After flashover, fire spreads quickly and burns all spaces continuously. If people do not escape from the scene of a fire before flashover, they probably wouldn't save their life. Therefore, it is important that the beginning of fire can be observed soon before it causes any real damage.

Conventional point-based fire detectors typically detect the temperature of space exceptionally rise by fire. These detectors rely on the temperature reach to a fixed point. An important weakness of point detectors is that in large space, it may take a long time for rise temperature to alarm a detector and they can't be operated in open spaces such as hangers, tunnels, storage, and offshore platform.

Owing to the limitation of the traditional concept, point-based detectors can't detect fire in early stage. In recent years, many researches are devoted to video fire detection that does not rely on proximity of fire to the detector. This enables it to incorporate standard video surveillance cameras with sophisticated image recognition and processing software to identify the distinctive characteristics of fire patterns.

1.2 Related Work

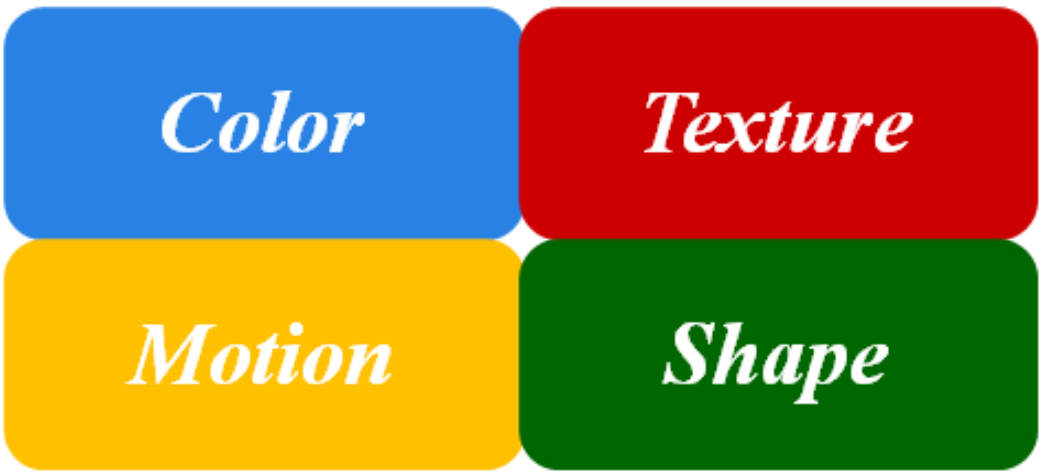


Fig. 1-2 Four categories of video fire detection

Methods of fire detection used for image processing thesis can be divided into four categories, as illustrated in the Fig.1-2. The first category is based on the color of the fire. Normally, fire will exhibit yellow, yellow-red, and red color. A fire model will be established with different color space according to the color distribution of the fire. Chen et al [1] and [3] noted that fire must comply with RGB color space

$R \geq G$, $G \geq B$, and $R \geq R_T$ distribution law. In addition, Horng et al [2] further established different color model in the HIS space according to saturation differences. This method can be adapted to different brightness environments, so that the fire model can be more widely used in various environments. Horng et al [8] combined the RGB model and the HIS model to identify the color of fire distribution. Celik et al [7] proposed that using the YCbCr space flame color model can efficiently separated fire luminance and chroma. This thesis flame color model also refers to this method, and making further improvements.

The second category is judged the fire texture. From a distance the burning flame appears similar, but upon closer look, nearby pixel brightness or chroma shows clear subtle changes. Ebert et al [3] uses preset masks to calculate neighboring pixel's red, green, and blue value difference of the area being observed. Only if there is enough blue color difference will it be judged to be flame pixel. Toreyin et al [4] uses wavelet analysis on the colors in the flame region. They believe that there must be enough variation in the fire area's neighboring pixel brightness, and using 2-D wavelet analysis, calculate the absolute value of the high-band and low-band to define the parameters. Only objects with bigger parameters can be considered fire bodies. Zhao et al [9] mentioned that entropy formula can be used to calculate the texture parameters of the region of interest.

The third category is determining the movement of the flame. The flame is a very special part of the fire. It will usually keep flashing and swaying on its own, or swing specifically due to the interference of the wind. This type of swaying will cause the flame to appear suddenly red and then suddenly returning to the background color in the red channel of the pixel. Gunay et al [5] proposed that the detected fire color moving body's pixel red channel be accumulated, and then use the wavelet

analysis on the time of the accumulated data to find its frequency, and then the Markov model be used to train the optimal threshold value.

The last category is to use the fire shape as the judgment standard. Fire shape is usually changing and rough, and different than the relatively smooth features of other objects. Borges [6] uses the object's actual circumference length and the circumference length ratio of the convex body nearest to this object to determine if this object is the fire's exterior. Gunay et al [5], according to different angles of the object, will uses data analysis of the distance from the center of mass of the object to the fire's boundary, and utilizing wavelet analysis to determine if this object's appearance is rough. Zhang et al [10] uses Laplacian operator to identify the boundaries of the object and conduct Fourier analysis to determine if it is the fire contour.

Chapter 2
System Overview

The following diagram is my proposed fire detection system structure, which can be divided into three parts, pre-processing, local block feature, and global feature.

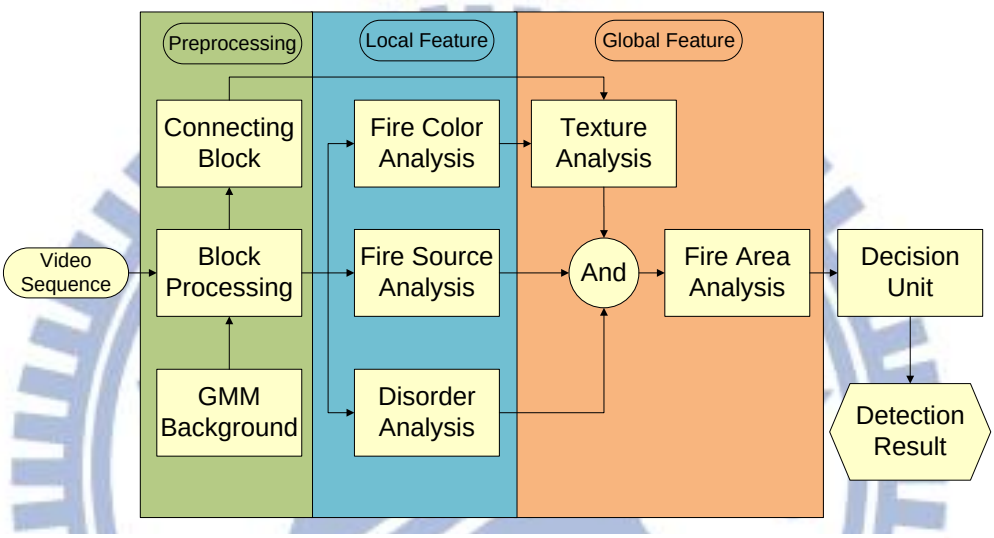
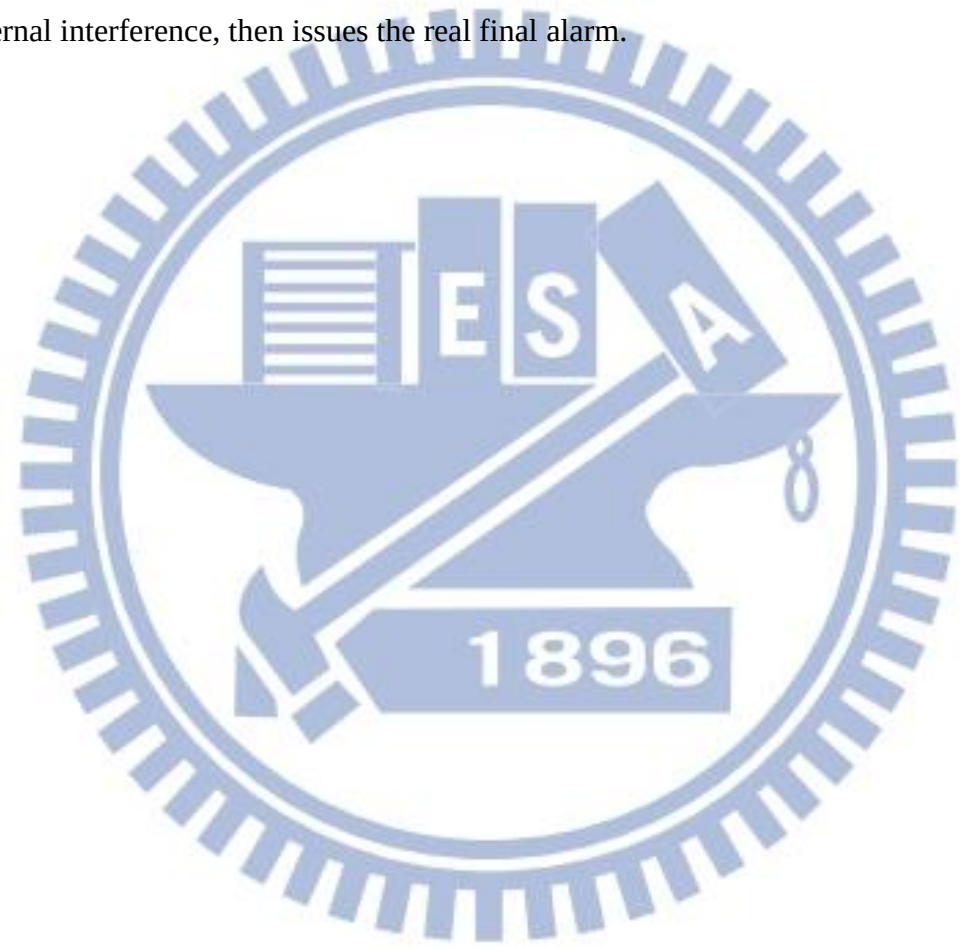


Fig. 2-1 System overview

Pre-processing is cutting the image into a fixed position block, and using the block as a unit to find if there is moving foreground in the block. If the side label is the candidate block, continue with the later feature calculation after labeling the candidate block. In this way, the calculation of blocks with no moving foreground can be eliminated. Connecting Component Block is using the same number label for the adjacent candidate blocks, forming it into one body. This part is the pre-processing for specialized variance analysis.

Local block feature is the feature analysis of fire body features when judging for candidate blocks, including fire color, fire source, and temporal difference. When the candidate block matches these three features, then it becomes a strong candidate

block. Global features include variance calculation and fire area analysis. Variance calculates the changes in pixel texture in the blame body. Fire area analysis uses strong candidate block as a benchmark to identify a region of interest, including the entire fire candidate object, and conducts a more accurate analysis of this region of the fire area. If the global feature is matched, it will give a single fire alarm signal. ADU conducts a buffer zone analysis of this alarm signal to avoid alarms caused by external interference, then issues the real final alarm.



Chapter 3
Fire Detection Algorithm

3.1 Block Processing

Figure is the flow chart of block processing. The input is the gray level image sequence, and the output is candidate blocks with moving property. There are two common methods for obtaining foreground image. One is temporal difference, the other is background subtraction. Temporal difference method is that we subtract frame *t-1* from frame *t*, and the regions with a obvious intensity variation are considered as foreground region. Background subtraction is also in similar way but we use a constructed background image instead of frame *t-1*.

Generally, cameras are usually set fixedly by considering surveillance monitoring system. Therefore, background subtraction is a better choice for our proposed algorithm. Foreground regions can be found by background subtraction, but they could also include static objects. Next, temporal difference of two successive frames will be calculated. The two methods are combined to further filter out static objects.

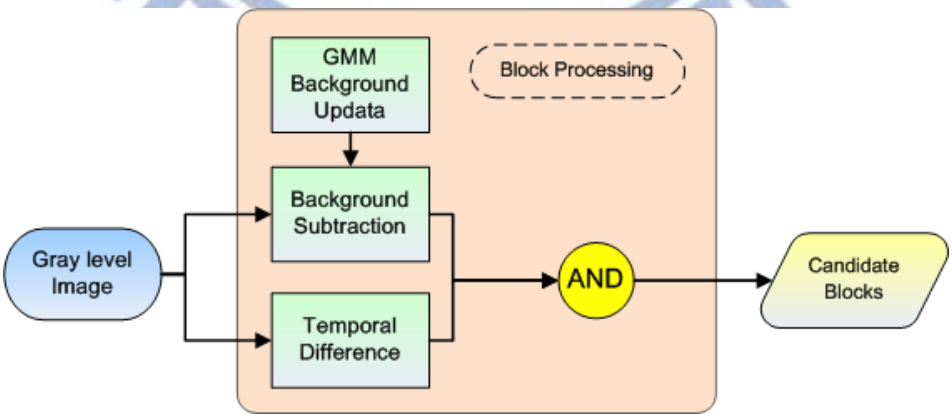


Fig. 3-1 Block processing

3.1.1 Background Modeling

Gaussian Mixture Model (GMM) [9] is a common and robust method in background construction, and we choose GMM to build the background image. It will be described as follows.

Generally speaking, the intensity of each pixel varies in a small interval except the region of foreground objects. It is proper to use a Gaussian model to construct the background image. But in many surveillance videos, we would observe that there are waving leaves, sparking light, etc. In these situations, some background pixels would vary in several specific intervals. In other words, using two, three or more Gaussian distributions to model a pixel will obtain a better performance. We present the flow chart of GMM background construction in Fig. 3-2.

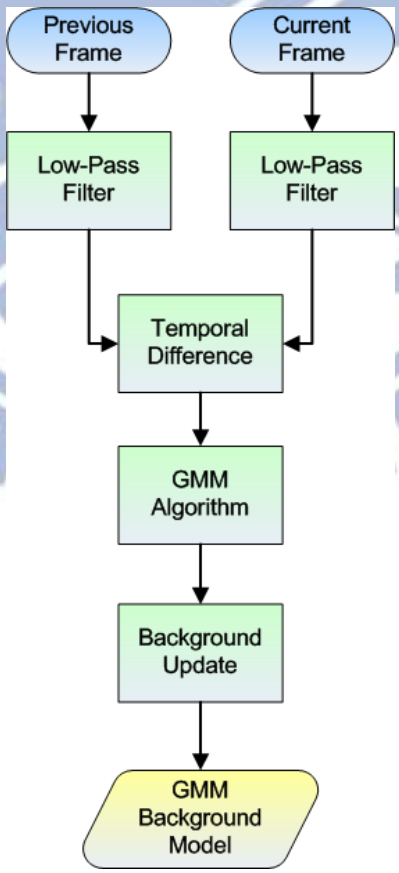


Fig. 3-2 GMM background model construction

Firstly, we use a low-pass filter to reduce the noise. The GMM method models intensity of each pixel with K Gaussian distributions. The probability that a certain pixel has a value of X_t at time *t* can be written as.

$$P(X_t) = \sum_{k=1}^{K} \omega_{k,t} \cdot \eta(X_t, \mu_{k,t}, \sum_{k,t}) \tag{3.1}$$

where K is the number of distributions that we used, $\omega_{k,t}$ represents the weight of *k*-th Gaussian in the mixture at time *t*, $\mu_{k,t}$ is the mean of *k*-th Gaussian in the mixture at time *t*, $\sum_{k,t}$ is the covariance matrix of the *k*-th Gaussian in the mixture at time *t*, and η is a Gaussian probability density function shown in Eq. (3.2)

$$\eta(X_t, \mu_t, \sum_t) = \frac{1}{(2\pi)^{n/2} |\sum_t|^{1/2}} \exp\{-\frac{1}{2}(X_t - \mu_t)^T \sum_t^{-1} (X_t - \mu_t)\} \tag{3.2}$$

where n is the dimension of data. In order to simplify the computation, it assumed that each channel of data are independent and have the same variance, and then can assume the covariance matrix as Eq. (3.3):

$$\sum_{k,t} = \sigma_k^2 \mathrm{I} \tag{3.3}$$

Temporal difference is applied to extract the possible background regions, and update pixels inside these regions. Then, we sort Gaussian distributions by the value of ω/σ, and choose the first B distributions to be the background model, i.e. shown as Eq. (3.4):

$$B = \arg\min_b (\sum_{k=1}^{b} \omega_{k,t} > T) \tag{3.4}$$

When a new pixel is inputted (intensity is X_{t+1}), it will be checked against the K distributions by turns. If the probability value is within 2.5 standard deviations, and this pixel is considered as background. Then, we update weight, mean, variance by Eq. (3.5), (3.6), (3.7):

$$\omega_{k,t+1} = (1-\alpha)\omega_{k,t} + \alpha(M_{k,t+1}) \tag{3.5}$$

$$\mu_{t+1} = (1-\rho)\mu_t + \rho X_{t+1} \tag{3.6}$$

$$\sigma_{t+1}^2 = (1-\rho)\sigma_t^2 + \rho(X_{t+1}-\mu_{t+1})^T(X_{t+1}-\mu_{t+1}) \tag{3.7}$$

where α is a learning rate, $M_{k,t+1}$ is 1 for the model which matched and 0 for remaining models, and Eq. (3.8) shows the second learning rate ρ.

$$\rho = \alpha\eta(X_{t+1}|\mu_{k,t},\sigma_{k,t}) \tag{3.8}$$

Besides, the remaining Gaussians only update the weight. If there is no any distribution is matched, we replace the mean, variance and weight of the last distribution by X_{t+1}, a high variance and a low weight value, respectively. Figure 3-3 shows the constructed background image by GMM. Figure 3-4 shows the foreground image obtained by background subtraction.

(a) Video sequence (b) GMM Background Image

Fig. 3-3 Background image construction by GMM

(a) Current image (b) Foreground image

Fig. 3-4 Foreground image obtained by background subtraction

3.1.2 Candidate Selection

Moving regions come into existence and disappear continuously because of the special particle property during ignition and combustion as shown in Fig.3-5. It is inefficient to track or analyze the target using object-based method. Block-based technique provides a better way to solve this problem. The image will be divided into non-overlapped blocks, and each block has the same size in a same image. First, we will find out the blocks with a gray-level change. The foreground image will be obtained by the GMM approach, and we compute the summation of foreground image for each block as shown in Eq. (3.9)

$$S_k = \begin{cases} 1, & if \left| \sum_{x,y \in S_k} \left[foreground(x,y,t) \right] \right| > T_1 \\ 0, & otherwise \end{cases} \quad (3.9)$$

where S_k is the kth block and x,y is the coordinates of the scene. T_1 is the predefined threshold.

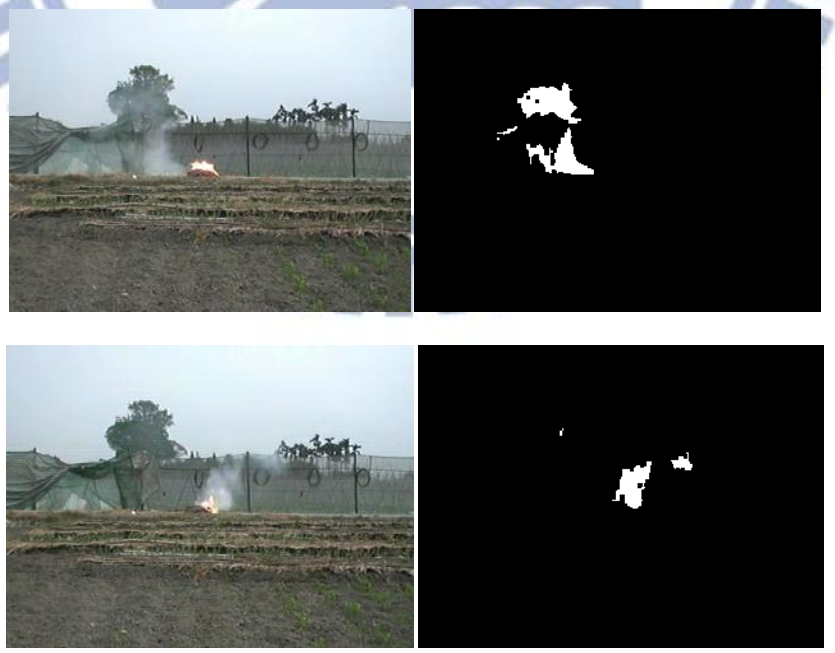


Fig. 3-5 Moving regions come into existence and disappear continuously

Foreground regions can be found by the GMM approach, but they could also include static objects. Next, temporal difference of two successive frames will be calculated. In dynamic image analysis, all pixels in the difference image with value "1" are considered as moving objects in the scene. As we know, video images usually have a great amount of noises due to intrinsic electronic noises and quantification. So the difference of two successive frames pixel inevitably produces false segmentation. To reduce the disturbance of noises, we also compute its summation for each block to determine the moving property. The block difference is defined as

$$T_k = \begin{cases} 1, & if \left| \sum_{x,y \in I_k} \left[f(x,y,t) - f(x,y,t-\Delta t) \right] \right| > T_2 \\ 0, & otherwise \end{cases} \tag{3.10}$$

where T_k is the k[th] block and x,y is the coordinates of the scene, f is the input image and T_2 is the predefined threshold.

In order to reduce the computational cost, only when the value of background subtraction and temporal difference lager than the predefined thresholds will be regarded as candidates containing moving objects by Eq. (3.11).

$$B_k = \begin{cases} 1, & if\ S_k\ AND\ T_k = 1 \\ 0, & otherwise \end{cases} \tag{3.11}$$

We consider the information of a particular block over time as a "block process" in the following sections. Fig.3-6 illustrates some results produced by block processing.

Fig. 3-6 Results of block processing

3.2 Fire Color Model

Digital Imaging is a combination of three RGB color information; and it is because humans has the highest sensitivity in accepting these three colors. Generally speaking, RGB channel in computer visualization is stored using 8bit for the color data. This also means that a channel of values ranging from 0 to 255 will have the higher concentration color the higher the value. The highest color white representation concentration is RGB (255,255,255), and the black color representation is RGB (0,0, 0). Chen et al [1] proposed that the main features of the fire in RGB space, speaking in terms of single pixel, are that R channel value will be greater than the G Channel value. The G Channel value would be greater than the B channel value, and the R value saturation level will be greater than the average R value of the entire screen. But the drawback of the RGB fire color model is that it does not take into account the impact of the camera on the fire in different brightness. That is, this algorithm may vary with the change in intensity, and resulting in a decrease in detection rate.

In order to improve this problem, Hasan Demirel, [3] proposed the fire color model based on YCbCr color space base. Y represents the luminance; Cb and Cr represent the blue-difference and red-difference chroma components. This fire color model also changed from the above described RGB color space algorithm calculation to the YCbCr color space calculation algorithm. In addition, it adds adaptive adjustment to the surrounding brightness, and improved other harmful effects, increasing detection performance. The algorithm we propose also refers to this algorithm, and improved. Because RGB space converting to YCbCr space is linear, we can use the following matrix to turn RGB color space to YCbCr color space:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (3.12)$$

The Y range is [16 235], and Cb and Cr range is [16,240].

Next we need to calculate the average of the three components of the YCbCr color space in the entire screen:

$$Y_{mean} = \frac{1}{K}\sum_{i=1}^{K} Y(x_i, y_i) \quad (3.13)$$

$$Cb_{mean} = \frac{1}{K}\sum_{i=1}^{K} Cb(x_i, y_i) \quad (3.14)$$

$$Cr_{mean} = \frac{1}{K}\sum_{i=1}^{K} Cr(x_i, y_i) \quad (3.15)$$

$(x_{i,} y_i)$ Represent the pixel coordinates. Y_{mean}, Cr_{mean}, and Cb_{mean} represents this screen's average luminance value and the concentration of blue and red average color difference chroma components, respectively. K Represent this screen's total pixel quantity.

The RGB space fire formula R> G> B and R> R_{mean} converted to YCbCr formula becomes:

$$Y(x, y) > Cb(x, y) \quad (3.16)$$

$$Cr(x, y) > Cb(x, y) \quad (3.17)$$

Therefore,

$$F(x, y) = \begin{cases} 1, & if\ Y(x, y) > Cb(x, y) \,\&\, Cr(x, y) > Cb(x, y) \\ 0, & otherwise \end{cases} \quad (3.18)$$

$F(x, y)$ Represent whether the pixel complies with the above formula. If it does, then it may be that the fire color pixel is 1. Otherwise, it is 0.

$Y(x, y)$, $Cr(x, y)$, and $Cb(x, y)$ separately represent the luminosity, red difference chroma component, and blue difference chroma component of the coordinates (x, y). That is to say, the luminosity must be greater than blue difference, and the red

difference must be greater than blue difference.

The fire picture statistics of the YCbCr distribution picture is as follows:

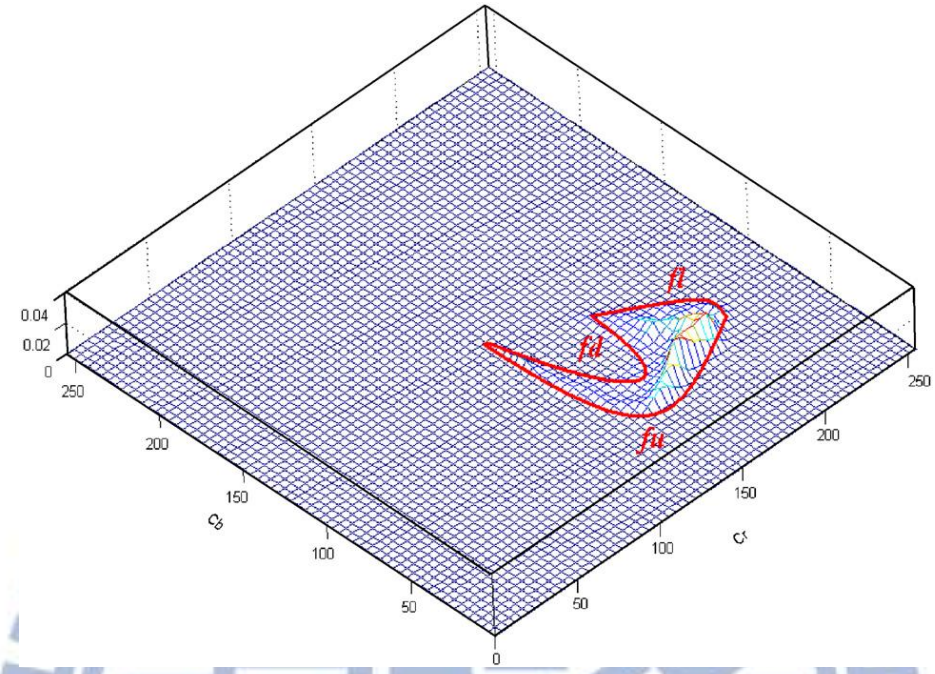


Fig. 3-7 distribution of fire color in YCbCr color space

In addition, from various fire pictures some futures can be observed. Pixel with fire usually will have greater luminosity than the average luminosity of the entire picture and the red difference chroma component will be greater than the average difference of the entire picture. The blue difference chroma component will be smaller than the average of the entire picture. The sum of the above-mentioned features is expressed with the following formula:

$$F(x,y)=\begin{cases}1, & if\ \ Y(x,y)>Y_{mean}\ \&\ Cb(x,y)<Cb_{mean}\ \&\ Cr(x,y)>Cr_{mean}\\ 0, & otherwise\end{cases} \tag{3.19}$$

Last, the fire color pixel will have a value difference between the red difference chroma component and the blue difference chroma component, as in the following:

$$F(x,y)=\begin{cases}1, & if\ |Cb(x,y)-Cr(x,y)|\geq\tau\\ 0, & otherwise\end{cases} \tag{3.20}$$

τ is a preset constant with the default range of 0 to 224. This parameter can be

adjusted according to different situation. The value suggested in the thesis is 40. Results retrieved from fire color model will usually include some objects that has similar color with the fire, but is not the fire, such as metal reflecting the fire in the picture. Thus, simply using color for analysis is not enough. Fire color is just a looser feature analysis. The experimental diagram is as follows:

Fig. 3-8 Fire color analysis results

However, we discovered in the experiment that this algorithm only meet the general red and yellow fire color. For some high temperature, closer to white color, high intensity fire, it cannot accurately be detected. Thus we cut the color model into two areas. One is normal intensity fire (Y <220), which we use the above algorithm. For high intensity fires (Y> 200), we adjust the conditions as follows:

$$f(Y(x,y) > 220) \tag{3.21}$$

$$F(x,y) = \begin{cases} 1, & if\ \ Y(x,y) > Y_{mean}\ \&\ Cr(x,y) > Cb(x,y) \\ 0, & otherwise \end{cases} \tag{3.22}$$

Main reasons which cause this situation is because for high intensity, Cr and Cb values will be closer to the maximum value. There will not be sufficient gap between the two values, so it is easy to not comply:

$$Cb(x,y) < Cb_{mean} \tag{3.23}$$

$$|Cb(x,y) - Cr(x,y)| \geq \tau \tag{3.24}$$

The pre and post modification comparison of these two formulas are as follows:

Fig. 3-9 High intensity formula modification comparison.

As the above figure shows, the modified algorithm can also detect high intensity fire. Conversely, false detection will increase. But because the color is the first hurdle, later we will propose other algorithms to filter out these errors. When the fire pixels in the block are over the block's total pixel by 10%, this block matches the fire color features.

3.3 Fire Source Analysis

In general, fire will have a burning fire source. Although the flame will move with the wind, we can observe that the fire source location will not usually move, or will expand with the fire, forming a slow movement. The burnt location will form a leftover concept.

The remnants represent objects that originally do not belong in the screen. After moving by certain methods into the observation screen, it will remain in the same place for a long time. For example, luggage dropped by traveler. Fire will also have similar features. Flames will instantaneously combust after a flammable material passes the ignition point. Flame originally does not belong in the screen, but is suddenly produced in the screen, and the fire source position will not change.

As can be seen from the above figure, fire at the fire source contains two features. First, high intensity feature in the fire source area, and second, the position is fixed. The fixed location feature just happens to be able to match our use of candidate blocks to judge the fire starting point. First, we must make sure there are fire pixels we are interested in inside the block:

$$\mathrm{B}_{count}(i,j)=\begin{cases}1, & FirePixel(x,y)\in B_k(i,j)\\ 0, & otherwise\end{cases} \tag{3.25}$$

Next, calculate the average intensity within the block, and calculate the average intensity of the entire image, and compare the two average intensity. If the block average intensity is greater than the average intensity of the image, then it matches the high intensity feature of the fire starting point. Then calculate this candidate block high intensity state holding time. This high intensity state must be greater than t times to match the fixed position feature. The judgment is as follows:

$$\mathrm{B}_{count}(i,j)=\begin{cases}Count++, & B_kYMean > IYMean\\ 0, & otherwise\end{cases} \tag{3.26}$$

If it matches the above described two features, this means that the candidate block passes the features

$$\mathrm{B}_{FS}(i,j)=\begin{cases}1, & B_{count}(i,j)>t\\ 0, & otherwise\end{cases} \tag{3.27}$$

3.4 Disorder Analysis

Fire will move with the wind, but will produce different changes in intensity due to the physical features of the combustion. Intensity change usually is produced randomly, with no set regularity. Even if the fire location area of the flame does not have any substantial changes, the pixel intensity will be different for every frame. We hope to be able to find a way to calculate this chaotic fire intensity change feature.

Intensity timeline changes have many different analysis methods, such as:

intensity timeline's differential, wavelet analysis. The actual significance is to calculate the difference in pixel intensity within the calculation time interval. The method we used is the Temporal Difference as our analysis method. The reason for using this method is that for the same calculation of the difference in timeline intensity, this calculation method is simpler, and has lower calculation computation compared to other methods. However, it can still calculate the intensity difference of the fire.

Temporal Difference is a simple method for calculating changes in two different time span pictures. Using the t time and t-1 time grayscale image subtraction in order to filter out the noise effect, so the subtracted absolute value must be greater than a threshold value as follows:

$$\mathrm{TD}(x,y)=\begin{cases}1, & \left|gray(t)-gray(t-1)\right|>TH_{gray}\\ 0, & otherwise\end{cases} \tag{3.28}$$

TH_{gray} is a preset threshold value which can be obtained via experimental experience. After calculation, the picture use thresholding method that matches the formula is displayed as shown below:

Fig. 3-10 temporal difference result

Because the method we proposed uses block as our observation unit, so we must use the calculated result and extend it to block unit feature analysis. A block must have a certain number of mixed intensity pixel to be considered matching this feature. TH_{count} is the default value.

$$B_{TD}(i,j)=\begin{cases}1, & TDCount(x,y)>TH_{count};TD(x,y)\in B_{TD}(i,j)\\ 0, & otherwise\end{cases} \quad (3.29)$$

3.5 Global Analysis

So far we used a total of three feature analysis, including fire color, disorder analysis, and fire source analysis. These three analysis methods all belong to the block local feature analysis, leaning towards using the fire's details and physical features as analysis. Next, we combine the local feature results to observe the fire's global feature.

This chapter's focus is placed on the global feature analysis of the fire. We must use a macroscopic observation angle to analyze the fire features, to make up for the inadequacies of the local features analysis. Global analysis contains three steps, texture variation value analysis, fire area analysis, and decision-making unit. The flow chart is as follows:

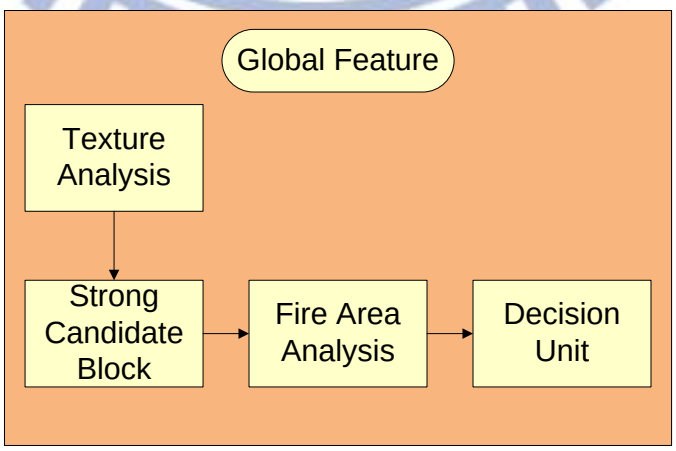


Fig. 3-11 Flow chart of global feature

3.6 Texture Variance Analysis

Simply relying on fire color for detection cannot completely filter out some objects that are fire like colored, such as red and yellow clothes or orange car. We have to put forward a new feature to filter the above-mentioned objects. And one thing that clearly separate fire from other objects is that, although fire objects are all red and yellow color, but the intensity and red difference chroma of each pixel will have subtle changes. That is to say, fire is not smooth textured, as shown below:

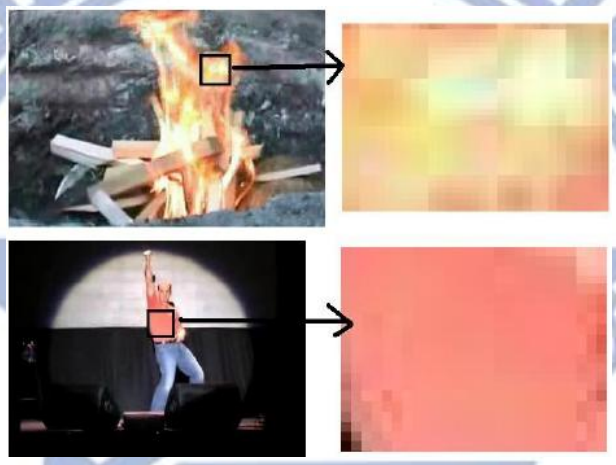


Fig. 3-12 Fire and fire like objects texture

Obviously, fire will have different color information in each pixel, whether it is intensity or red color difference. We must find a way to calculate the color feature differences in the candidate region.

Variance analysis is a common data analysis model. The variance of a set of data describes the degree of dispersion of this set of data. That is, the distance of this set of data in comparison to the expected value (means). Simply put, the variance also represents the average degree of dispersion for that set of data.

The first step in calculating the variance is to calculate this data set's combined average value:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.30}$$

x_i is our observed feature value, and the feature value we use is based on connecting block labeling as the same object inside the fire pixel's Cr value for calculation. N represents the fire pixel number within this Label. Pay attention here, because not all the label's pixels are calculated for Cr variance. Only fire pixels that match the above fire color algorithm are listed into calculation. Foreground moving object calculated by block algorithm often includes non-flame objects near the fire position (i.e. smoke), which we must eliminate.

The following is the variance formula:

$$Var(l) = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N} \tag{3.31}$$

Similarly, because high intensity fire's Cr is close to the maximum value, it will cause special circumstances in the variance formula calculation. So in high intensity fire, we must modify the variance formula to match the high intensity fire.

High intensity fire object pixel Cr values are close to the maximum value (Cr> 230), the gap spacing of the Cr values with each other is not large enough, to improve this problem, the easiest way is to adjust the threshold value of the variance σ, but the high intensity of the fire edge might not match $Y(x, y) > 220$. In order to be able to complete analysis of all the pixel in the fire, our method is to increase the fire object's weight that matches $Y(x, y) > 220$, and modify the formula to:

$$Var(l) = \frac{\sum_{i=1}^{N} w(x_i - \mu)^2}{N} \tag{3.32}$$

When the fire pixel intensity $Y(x, y) > 220$ multiplies with w weight, the fire intensity is $Y(x, y) < 220$ but fire weight is constant w = 1. In this way we can

ensure all the pixels in the fire has been included in the calculation.

l is the label given to a specific object through connecting block label. The same label's block represent the same object's components, which when combined, forms a complete object. Different objects will have different variable calculation, and the variable after calculation must match the following formula:

$$B_{var}(i,j)=\begin{cases}1, & Var(l)>\sigma \,\&\, FirePixel \in B(i,j)\\ 0, & otherwise\end{cases} \tag{3.33}$$

σ is a preset constant, representing the variance of the data collection must have a level of dispersion. A block must include fire color pixel to be labeled as a possible fire-block.

3.7 Strong Candidate Block

We used fire color analysis, disorder analysis, fire source analysis, and texture variance analysis, all of the analysis results all use block as the unit for result output. Strong fire candidate block represent matching all of the 4 aforementioned feature analysis blocks, as in the below figure:

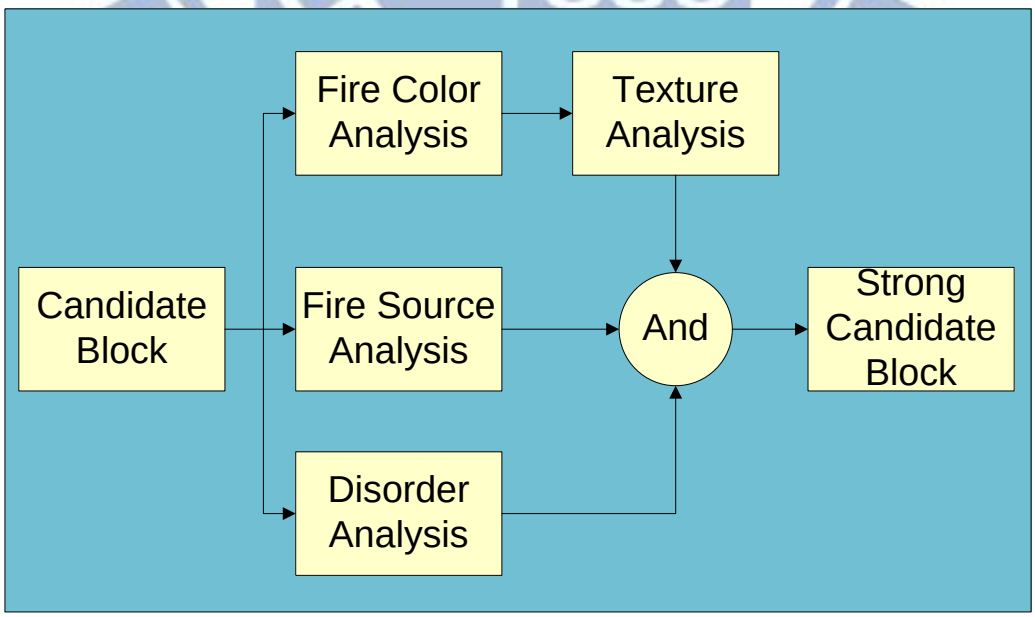


Fig. 3-13 Flow chart of strong candidate block

Strong candidate block is very close to the fire decision, but some non-flame objects can pass through these four feature calculations, such as flashing lights. Therefore, there needs to be an additional feature for verification. Strong candidate block will be used as global feature analysis benchmark.

3.8 Fire Area Analysis

Strong candidate block that has been through local feature analysis as well as texture variance analysis already has a certain effect on fire detection, and the false alarm rate has also been lowered to a certain level. But flashing red light not only matches flame color and changing artistic features, it also matches fire source and screen disorder features.

Fig. 3-14 Flashing light

Fig. 3-15 False detection of flashing lights

Therefore, a new feature analysis must be put forward to filter out this type of

common false report. The biggest difference between flashing light and fire is their rate of area change. Flashing lights usually will instant shine and instantly darken. This will have rapid changes in area of fire color detection. However, although fire area will shrink and expand with the wind, and expand with the movement of the fire, but the surface area's sudden change rate will be at a certain rate.

3.8.1 Candidate Region Extend

First, we must find a new observation area. This observation region must contain the entire fire objects, so it is necessary to identify the possible location of the fire, and use the possible location as the benchmark, drawing a new observation area.

Since the prior local block analysis has helped us find strong candidate blocks of possible fire, we use the strong candidate block as the benchmark. Find the top most, bottom most, left most, and right most strong candidate block, and extend out one block coordinates as our observation area boundary.

Fig. 3-16 Extend region

The red block in the left figure is the strong candidate block that match the four feature analysis. The right figure is the extended observation area.

Because the flame tip may move with the wind at any time, it may not match the fire source analysis. Extending the block out one will be able to include the moving flame in the observation area.

3.8.2 1-D Wavelet Analysis

We will store the fire surface area in the observation area into 1-D continuous signal. Facing the sudden change rate of the fire surface area, the most common is to make a differential signal. Differential signal is convenient for retrieval of sudden changes in the signal, with low cost in computing. It is a method suitable for calculating changing values. However, general differential can only be used to obtain differential in the previous data or the future data, not both.

In contrast, one-dimensional wavelet conversion is able to focus the analysis in a short period of time, at the same time, consider changes in the past and future, and is suitable for transient signal analysis. Although wavelet conversion and short-time fourier transformation(STFT) is used for time domain and frequency domain localized analysis. But there are some frequency area and time resolution problems that wavelet often can give better results after passing multi-resolution analysis, and wavelet conversion calculation's complexity is also lesser. Wavelet analysis time only requires O(N) time, but fast Fourier needs O(N log N).

Discrete wavelet transformation analysis is very useful in computer numerical analysis. It has discrete input and discrete output, exactly in line with our computing needs. However, it does not have a simple and clear formula, only expressed using hierarchy structure. The inputted $x[n]$ is a continuous 1-D signal representative of

the observation area continuous time period fire area. $g[n]$ and $h[n]$ separately represents low-band filter and high-band filter. The high and low band filter coefficients are $\{-\frac{1}{2},\frac{1}{4},-\frac{1}{2}\}$ $\{\frac{1}{2},\frac{1}{4},\frac{1}{2}\}$. After low band and high band filter processing, the signal needs to go through a down-sampling. $\downarrow 2$ represents a down-sample filter, and after down-sampling a high-band signal and low –band signal can be obtained.

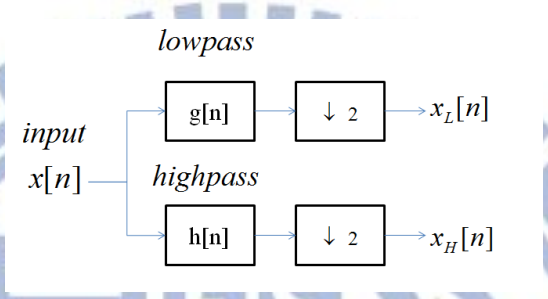


Fig. 3-17 Block diagram of filter analysis

The focus of our concern is the instantaneous rate of change of the area. The focus of observation is placed on the high-frequency portion of the signal. From the figure we can see the fire surface area change rate and the flashing light area change rate trend. We can see that the flashing light fire area change is more dramatic, but the fire's surface area change is milder.

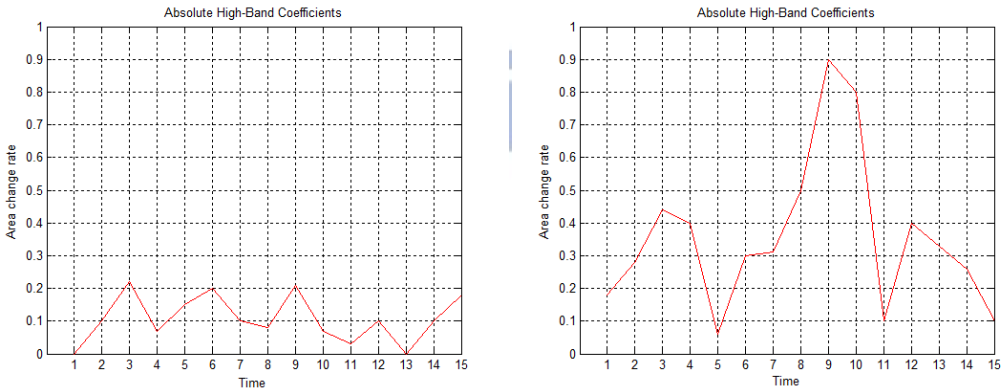


Fig. 3-18 Comparison of changes in value of area ratio at the passage of

(a) fire object (b) non-fire object

We propose a feature value:

$$\beta = \frac{\sum_n \left| x_H [n] \right|}{n} \tag{3.34}$$

This feature is used to accumulate the fire surface area change rate over a period of time. n is representative of the amount of data, and β feature is inversely proportional to the fire probability.

3.9 Alarm Decision Unit

Image-based detection system often receive the wrong image information to the camera due to external influences, such as the shaking or change in brightness caused by strong winds or the bus passing by. Although the image is returned to normal after a short time, this type of wrong information can cause the detection system's momentary judgment error. However, this type of alarm is not accurate and instantaneous.

In order to make the entire detection system more reliable and widespread, these sudden false alarms must be filtered out. Because f will not suddenly be produced or suddenly disappear, we can use the alarm buffer to eliminate this type of situation, as demonstrated in the figure. The buffer collects all the system produced false alarms from the image sequence and ratio of alarm data and the non-alarm data in this time period.

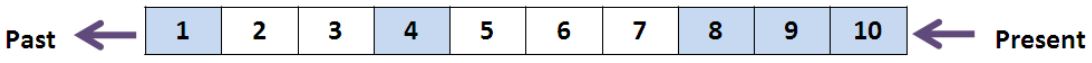


Fig. 3-19 Alarm decision unit

If the alarm ratio is over 50% of the time interval, then a real alarm is issued. This method passes the signal through the buffer to produce the real fire alarm, and can handle the noise of the camera caused by sudden external influence and reducing false alarm reports. Clearly, this method requires adding a little reaction time, but the system reliability and stability is beyond doubt. Below picture is a real fire alarm produced by the buffer

Fig. 3-20 Final fire alarm

Chapter 4
Experimental Results

In this chapter, several results of fire detection will be presented. Our algorithm was implemented on the platform of PC with Intel Core i5 2.53GHz and 2GB RAM. Borland C++ Builder is our development tool and operated on Windows 7. All of our testing inputs are uncompressed AVI video files and DVD video data acquired by USB video capture. The resolution of video frame is 320*240.

In section 4.1, we will show the experimental results of the proposed algorithm on different scenes. Besides, accuracy rate and comparison between features are demonstrated in section 4.2. In section 4.3, we have a brief discussion of efficiency of our proposed algorithm.

4.1 Experimental Results of Fire Detection

In the following, we use “yellow boundary” to represent global candidate fire region and the region become “red” region that means this region exist fire object. The columns of left side contain original video sequence and the columns of right side contain detection results of the proposed algorithm.

Figure 4-1 illustrates the outdoor environment situation. There is no wind in Fig. 4-1(a).But the wind is blowing hard in Fig. 4-2(b).Our features wouldn’t be affected by the external environment and fire region can be detected correctly.

(a)

(b)

Fig. 4-1 Outdoor environment

Figure 4-2 illustrates the indoor environment situation. Fire region can be detected correctly.

(a)

(b)

(c)

(d)

Fig. 4-2 Indoor environment

Figure 4-3 For situations when fire is reflected on metal, it can still capture the correct fire region.

(a)

(b)

Fig. 4-3 Fire is reflected on metal

Figure 4-4 Fire in an extremely dark space.

Fig. 4-4 extremely dark space

Figure 4-5 Is a high intensity fire case, as shown in figure, can still find the correct fire region.

Fig. 4-5 High intensity fire

Figure 4-6 are non-fire test videos, most of which are traffic video or pedestrian videos. Because generally these videos do not have fire characteristics, thus no strong candidate blocks are generated, and will not produce global candidate fire region. Therefore, we show the foreground movement candidate block to represent that this video has a foreground moving object. 4-6 (d) is a special case. The entire screen is reddish hued, and color signal judgment becomes useless. However, we can still rely on other fire features to accurately judge if a fire exists or not.

(a)

(b)

(c)

(d)

Fig. 4-6 Non-fire situations

Figure 4-7 event is flashing red car lights. This event is in line with most of the fire features, including fire color, fire source, temporal difference, and variance analysis. Fire surface area feature was finally used for filtering out.

Fig. 4-7 Filter out flashing red light

A large variety of conditions are tested including indoor, outdoor and sunlight variation each containing fire events, pedestrians, bicycles, motorcycle, tourist coaches, trailers, waving leaves, etc in Table 4-1. In our experiments, there are 13257 positive samples (fire events) and 62614 negative samples (ordinary moving objects).

Table 4-1 Properties of the testing videos

Movie List	**Descriptions**
Movie_01	Outdoor fire with pedestrians
Movie_02	Outdoor fire with pedestrians at farmland
Movie_03	Fire in the living room
Movie_04	Fire in yellow tone living room
Movie_05	Sofa on fire
Movie_06	The boiler burn up
Movie_07	High intensity fire under bridge
Movie_08	Fire in the dark room
Movie_09	Fire in the wide storehouse
Movie_10	Christmas tree burn up
Movie_11	Car with blinking light in tunnel

Movie_12	Cars in red tunnel at night
Movie_13	A trailer tow away a truck with pedestrians
Movie_14	Cars with dark shadow
Movie_15	Cars in tunnel in day time
Movie_16	Cars in tunnel at night

4.2 Accuracy Discussion

Data in Table 4-1 show the evaluations of each feature and the global testing result without ADU (Alarm Decision Unit). The reaction time is obtained by the ratio between frames to detect and frames per second. The detection rate and the false alarm rate are calculated as follow:

$$\text{Detection Rate} = \frac{N_{\text{detected}}}{N_{fire}} \times 100\% \quad (4.1)$$

$$\text{False Alarm Rate} = \frac{N_{\text{false detected}}}{N_{\text{non-fire}}} \times 100\% \quad (4.2)$$

Fire color feature threshold is set lower because fire color is the most basic judgment feature. We hope that all fire-like color objects can pass this feature. From Table 4-1 we can see that the false positive rate of the detection rate is also high. Fire source represent the higher flame intensity and fixed location feature. This has good affects in moving light or objects that does not have enough intensity. Temporal difference means that fire in a fixed block have continuous movement. This feature can eliminate fixed position, but non-moving lights, i.e. street light. Next, using global feature for verification, the variance can be filter out objects with fire like colors, but with smooth texture, i.e. red clothes, yellow car. After match the previous four feature values and the block becomes a strong candidate block, and has a high likelihood of being a fire body. Lastly, the fire surface area analysis will be used to make the last judgment.

We can see from the table that each feature will complement each other in filtering out false positives. Each algorithm forms a complementary relationship with each other, with variance as a strong feature, can lower the most false positive rates.

Table 4-2 Experimental results without ADU based on single frame

	Detection Rate	False Alarm Rate	Reaction Time (sec)
Fire Color Analysis	99.2%	35.6%	--
Fire Source Analysis	93.7%	18.7%	--
TemporalDifference Analysis	93.3%	24.4%	--
Variance Analysis	92.2%	5.3%	-
Strong Candidate	91.2%	3.0%	-
Fire Area Analysis	90.2%	2.8%	2.95

Data in Table 4-2 show the global testing result with ADU (Alarm Decision Unit). Although the final verification step might lose some detection rate, the false alarm rate can further decrease from 2.8% to 0.8%.

Table 4-3 Experimental results with ADU based on single frame

	Detection Rate	False Alarm Rate	Reaction Time (sec)
Local + Global Analysis	90.2%	2.8%	2.95
Local + Global Analysis + ADU	**89.8%**	**0.8%**	**3.45**

4.3 Comparison

The analysis of experiments implementing the proposed process derived in previous sections is presented in this section. Although most of the papers in the literature don't provide experimental data, we implement some approaches and calculate the experimental results. The comparison results are presented in Table 4-3.

We were using the same test video in the proposed method in this thesis [1], [3], but the result of the comparison is based on the video number as a unit rather than frame unit. Because the performance of the image processing algorithms is more dependent on video scenes, and different test film are different length, specific algorithms will result in significant increase in false positive rates if the film is longer. In order to avoid this situation, we changed to using the video number as a calculation unit for detection rate and false alarm rate.

Our test includes 10 fire videos, 6 non-fire videos, using three algorithms for testing respectively, and the test results are as in the following table:

Table 4-4 Comparison between the proposed method and work in [1][3]

	Detected Fire Videos	False Alarmed Videos	Reaction Time (sec)
Proposed	**10**	**1**	**3.45**
Chen et al[1]	9	3	2.27
Celik et al[3]	9	4	1.69

In method [1], using RGB fire color model, and using two consecutive video's differential to measure the degree of fire disorder. This has the same effect as using the temporal difference, both measuring fire disorder feature. However, [1] not raise

more algorithms for verification, thus we can see from the table that method [1] has high detection rate, but also high false alarm rate. And this paper does not consider the high intensity fire, so the only fire that is undetectable is the high intensity fire.

Method [3] in this thesis refers to and improves the YCbCr fire color model. This paper has very precise fire color judgment; however, this thesis only used color information, and did not use any other fire object feature for further confirmation. Thus the reaction time is the shortest, but the false positive rate is the highest for the three methods, and similarly does not have high intensity fire detection affects.

The method proposed in this thesis has the highest detection rate and the lowest rate of false positives. This is using the fire's different features to come up with different algorithms that filter out a variety of false positives. This is the result of mutual authentication of all the algorithms, and even lacking just one is indispensable. However, the proposed algorithm uses more time accumulated data, thus the reaction time will be longer, which is a pity. But we hope that this set of algorithm calculation can be practically applied to daily life, so system stability is our emphasis.

Chapter 5
Conclusions and Future Works

It is inefficient to track or analyze the target using object-based method. Block-based technique provides a better way to solve this problem; Three local features separately calculate the fire feature in the block, filtering out specific false alarm, so that these algorithms can be applied to a variety of environments. Then global feature is used on global information to judge if the object connected to the block is a fire body. Decision Unit will enable the system to allow false alarm caused by instantaneous changes.

Most of other's algorithms are only seeking higher detection rate. It does not provide enough information on how accurate the system might be. When considering accuracy of fire detection, people care more about how to decrease the false alarm rate and detect fire events quickly, rather than just increase the detection rate. Here the false alarm rate of the proposed system is significantly lower than others. This system can also locate the fire regions correctly even when both fire and non-fire objects exist in the same frame due to block processing, while other systems only detect whether there is fire existing in the whole video or the single frame.

The proposed algorithm can be operated well in variant environment. However, to further improve the performance of our system, some enhancements or trials can be made in the future. First, the reaction time slightly increased. This is because the proposed algorithm uses more information from the timeline, and the accumulation time of the data will cause the reaction time to become longer. Second, a chandelier equipped with spiral fan will cause the chandelier to have minor shaking, thus matching the fire feature algorithms and cause a false alarm. Therefore, if these

problems can be solved, our algorithms will be reliable.

This paper demonstrates a robust and efficient system for fire detection, and it involves the global and local features analysis for each candidate block. Experimental results show the opportunity of the real-time operation of surveillance systems and advanced applications. The false alarm rate of the proposed system is lower than that of the state of art. The proposed algorithm has low computation load and has been implemented on embedded system.

References

[1] T. H. Chen, P. H. Wu and Y. C. Chiou, "Application of panoramic annular lens for motion analysis tasks: surveillance and smoke detection," *2004 International Conference on Image Processing (ICIP).*

[2] W. B. Horng, J. W. Peng and C. Y. Chen, "A new image-based real-time flame detection method using color analysis," *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE.*

[3] J. Ebert and J. Shipley, "Computer vision based method for fire detection color videos," *Pattern Recognition Letters 1 January 2006, Pages 49–58.*

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

[5] O. Gunay, K. Tasdemir, B. U. Toreyin and A. E. Cetin, "Fire detection in video using LMS based active learning," *Fire Technology Volume 46, Number 3(2010), 551-557.*

[6] P. V. K. Borges and E. Izquierdo, "A probabilistic approach for vision-based fire detection in videos," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 20, NO. 5, MAY 2010.*

[7] T. Celik and H. Demirel, "Fire detection in video sequences using a generic color model," *Fire Safety Journal 44 (2009) 147– 158.*

[8] W. B. Hrong, J. W. Peng and C. Y. Chen, "A new image-based real-time flame detection method using color analysis," *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*

[9] J. Zhao, Z. Zhang, S. Han, C. Qu, Z. Yuan and D. Zhang, "SVM based forest fire detection using static and dynamic features," *Computer Science and Information*

Systems Volume 8, Issue 3 (June 2011)

[10] Z. Zhang, J. Zhao, D. Zhang, C. Qu, Y. Ke and B. Cai, "Contour based forest fire detection using FFT and wavelet," *2008 International Conference on Computer Science and Software Engineering*

[11] Stauffer. C, Grimson. W.E.L, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, June 1999.

[12] http://www.openvisor.org

[13] R. C. Gonzales and R. C. Woods, Digital image processing, Prentice Hall,2002