

Fire Detection on a Surveillance System using Image Processing

Prof. Amit Hatekar¹, Saurabh Manwani², Gaurav Patil³, Akshat Parekh⁴
Department of Electronics and Telecommunication
Thadomal Shahani Engineering College, Bandra, Mumbai

Abstract - The process of oxidation of any material in the exothermic process of combustion, releasing heat and light as byproducts, is called Fire. The light parameter and the color of the flame helps in detecting fire. Fire detection using color information has many applications in computer vision and other domains. Our color model based method used for fire detection has many advantages over conventional methods of smoke detection etc., such as simplicity, feasibility and understandability. In order to enhance the performance parameters of fire flame detection based on a live video stream, we propose an effective color model based method for fire detection. Each and every pixel is checked for the presence or absence of fire using color features, and periodic behavior in fire regions is also analyzed. Dynamic boundary check is also done to detect the edges of the fire Region of Interest (ROI). Candidate fire regions are detected using the chromatic and dynamic measurements.

Keywords: Fire Detection, OpenCV, Raspberry Pi, Background Subtraction, Contour Detection, Image Processing, Motion Detection.

1. INTRODUCTION

The purpose of the project is to solve the existing problem of unreliable fire detection systems used in industrial warehouses. The project is aimed at using surveillance cameras in order to detect and monitor the occurrence of fire. Since the cameras are already installed in places, this system is aimed at diminishing the disadvantages of false alarm, making the system cost effective and fast method of detecting fire. The system uses Open Source Computer Vision, also known as OpenCV, is an open source freeware which is aimed at computer vision.

In warehouses, a surveillance network consisting of cameras exist due to security and insurance needs. Warehouses usually also contain an expensive hazard detection system consisting of fire detection and smoke detection devices. These devices may or may not be too reliable in case of a fire. Also, these devices cause plenty of false alarms due to cigarette smoke or incense smoke etc. This led us to developing an algorithm which could be using the already existing surveillance system so as to detect fire from a live video feed by processing it. This also reduces the cost of purchase and maintenance of the expensive and unreliable fire detection systems.

Our proposed system provides fire detection using a simple algorithm. It is as described below.

Firstly, the image frame is acquired from the live video feed. The RGB color model is then applied to the frame. The resultant RGB frame is then converted to a HSV frame. This frame is then passed through thresholding, median blurring (to remove noise), Background Subtraction, Sobel edge detection, and motion detection windows. The resultants of all these are then combined using Bitwise AND operation. Segmentation techniques are applied on this resultant to produce the final result, i.e. detection of the absence or presence of fire in the frame. A suitable response is displayed on the window monitor, and an alarm buzzer is sounded.

Experimental results show that our system is effective for various fire-detection tasks in real-world environment settings.

2. OTHER METHODS

Existing systems include Fire and Hazard Detection systems which employ heat sensors or temperature sensors or smoke sensors or a combination of these. These are installed at heights which are usually floor level (or ceiling level). These contain individual sensors which are not lined together, which leads to unpredictability and non-synchronous behavior of alarm.

A smoke detector is a device that senses smoke, typically as an indicator of fire. Fire alarm system known as smoke alarms, generally issue a local audible or visual alarm on detection of smoke. Generally, fire alarm consists of smoke detectors with a basic assumption that smoke will be generated by the fire. If we detect smoke, then the fire is detected. Even if there is any fire, the smoke may be generated quite later after burning the surroundings. For some fires, smoke may not be generated or it takes long time for the smoke detectors to detect the smoke.

Limitations of existing systems are:

1. Only after detection of smoke, the fire is detected.
2. Even if there is any fire, the smoke may be generated quite later after burning the surroundings. For some fires, smoke may not be generated.
3. It takes long time for the smoke detectors to detect the smoke.
4. Surrounding materials will be burnt till the next precautionary measure is taken. [1]

Large numbers of techniques have been developed for the fire detection from images of videos due to the number of vision based algorithms proposed in various literature surveys. Conventional methods of fire detection have been practically replaced with Video-based smoke detection methods due to various advantages over conventional methods like such as early fire detection, speedy response, non-presence of spatial limits, also information of the fire progress can be achieved due to live video and in fire investigation vision based is capable to provide forensic evidences.

3. PROPOSED SYSTEM

We propose a system which automatically detects the presence of fire based on the algorithm described subsequently.

The basic block diagram is given in Fig 3.1 as follows-

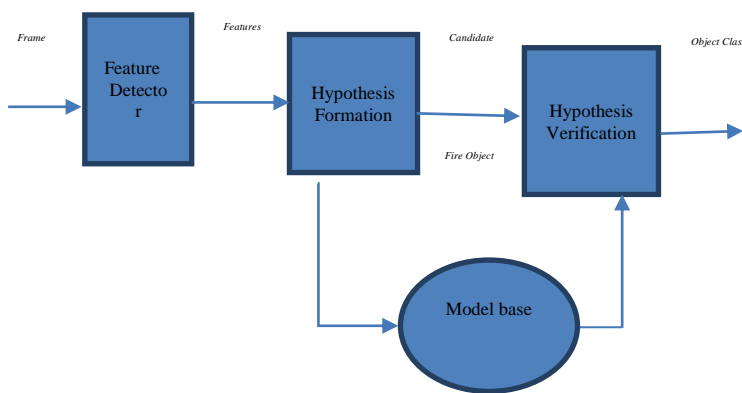


Fig 3.1: Block Diagram



Fig 3.2 Webcam

A. Webcam

A Webcam is installed at locations where inventory has to be monitored. It is connected to the computer via a USB 2.0 Cable where image processing of frames is done in OpenCV. It takes the snapshot from the captured video where further processing such as Background Subtraction, Segmentation, etc. is done. It has to be positioned properly so as to capture the entire area under observation.

B. Background Subtraction

Background subtraction is a major preprocessing steps in many vision based applications. For example, consider the cases like visitor counter where a static camera takes the number of visitors entering or leaving the room, or a traffic camera extracting information about the vehicles etc. In all these cases, first you need to extract the person or vehicles alone. Technically, we need to extract the moving foreground from static background.

If we have an image of background alone, like image of the room without visitors, image of the road without vehicles etc., it is an easy job. Just subtract the new image from the background. We get the foreground objects alone. But in most of the cases, we may not have such an image, so we need to extract the background from whatever images we have. It becomes more complicated when there is shadow of the vehicles. Since shadow is also moving, simple subtraction will mark that also as foreground. It complicates things.

Several algorithms were introduced for this purpose. OpenCV has implemented three such algorithms which are very easy to use. We will see them one-by-one.

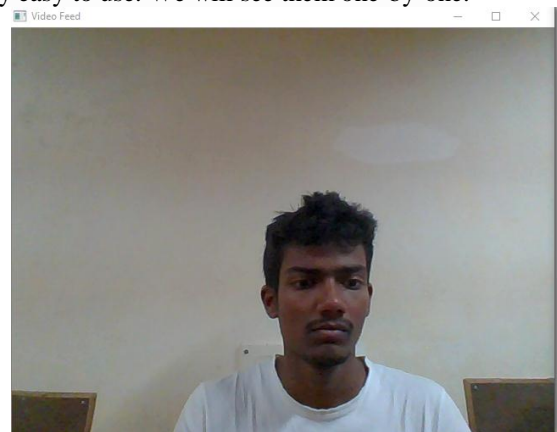


Fig 3.3 Video Feed



Fig 3.4 Background Subtraction

1. BackgroundSubtractorMOG

It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It uses a method to model each background pixel by a mixture of K Gaussian distributions (K = 3 to 5). The weights of the mixture represent the time proportions that those colors stay in the scene. The probable background colors are the ones which stay longer and more static.

While coding, we need to create a background object using the function, `cv2.createBackgroundSubtractorMOG()`. It has some optional parameters like length of history, number of Gaussian mixtures, threshold etc. It is all set to some default values. Then inside the video loop, use background sub tractor. Apply () method to get the foreground mask.[3]

2. BackgroundSubtractorMOG2

It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. One important feature of this algorithm is that it selects the appropriate number of Gaussian distribution for each pixel. It provides better adaptability to varying scenes due illumination changes etc. [4][5]

3. BackgroundSubtractorGMG

This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation. It uses first few (120 by default) frames for background modelling. It employs probabilistic foreground segmentation algorithm that identifies possible foreground objects using Bayesian inference. The estimates are adaptive; newer observations are more heavily weighted than old observations to accommodate variable illumination. Several morphological filtering operations like closing and opening are done to remove unwanted noise. You will get a black window during first few frames. It would be better to apply morphological opening to the result to remove the noises. [2]

C. Fire Contour Detection

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape

analysis and object detection and recognition. For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection.

Find Contours function modifies the source image. So, if you want source image even after finding contours, already store it to some other variables.

In OpenCV, finding contours is like finding white object from black background. So, remember, object to be found should be white and background should be black.

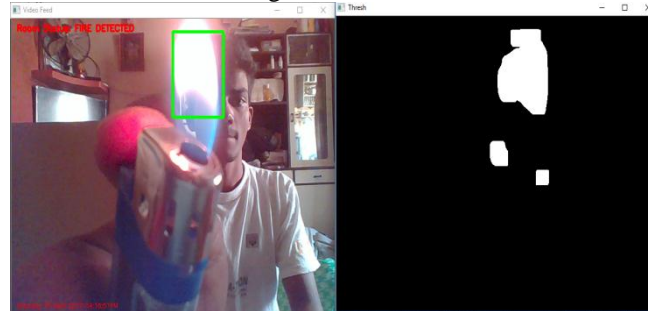


Fig 3.5 Contour analysis

As seen in Fig 3.5 rectangular boxes are made around detected flames which are the region of interests. At the left-hand corner presence or absence of fire in captured video frame is given.

See, there are three arguments in `cv2.findContours()` function, first one is source image, second is contour retrieval mode, third is contour approximation method. And it outputs the image, contours and hierarchy. Contours is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x, y) coordinates of boundary points of the object.

Contour Approximation Method

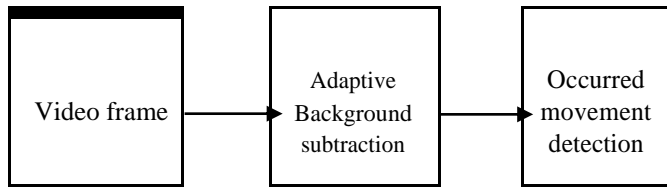
This is the third argument in `cv2.findContours` function. What does it denote actually?

Above, we told that contours are the boundaries of a shape with same intensity. It stores the (x, y) coordinates of the boundary of a shape. But does it store all the coordinates? That is specified by this contour approximation method.

If we pass `cv2.CHAIN_APPROX_NONE`, all the boundary points are stored. But actually, do we need all the points? For e.g., we found the contour of a straight line. Do we need all the points on the line to represent that line? No, we need just two end points of that line. This is what `cv2.CHAIN_APPROX_SIMPLE` does. It removes all redundant points and compresses the contour, thereby saving memory. [6][7]

D. Motion Detection

Motion detection method for fire is used to detect occurrence of any movement in a video. It is done by analyzing the difference in images of video frames. The most common approach used for motion detection is background subtraction which can identify any movement in the video frame. [8] There are three main approaches used in background subtraction for motion detection in a continuous video stream:



E. Color Detection System:

Normally the input is in RGB form, and then RGB form is converted to HSV form. Then the range representing HSV form of fire is applied to detect only fire characteristics. Fire in HSV form is then displayed. HSV color space is chosen purposely because it has ability to differ illumination information from chrominance more effectively than the other color spaces. Threshold values for the fire are loaded in to the system, as per the threshold values color detection system display result only if the fire is detected. [9][10][11]

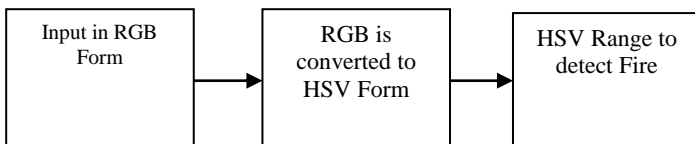


Fig 3.5: HSV Form

F. Background Registration:

In this on the basis of past several difference framespixels which are not moving for long time are used to identify the background for background registration. The binary frame difference map are used together to create the binary moving pixel map. This method has two major advantages that is the less computations are required and the background model is highly adaptive.

G. Segmentation:

In computer vision, image segmentation is the process of partitioning a digital image into something sets of pixels known as (subpixels). The goal of segmentation is to simplify and/or change the representation of an image. [8]

4. Flowchart

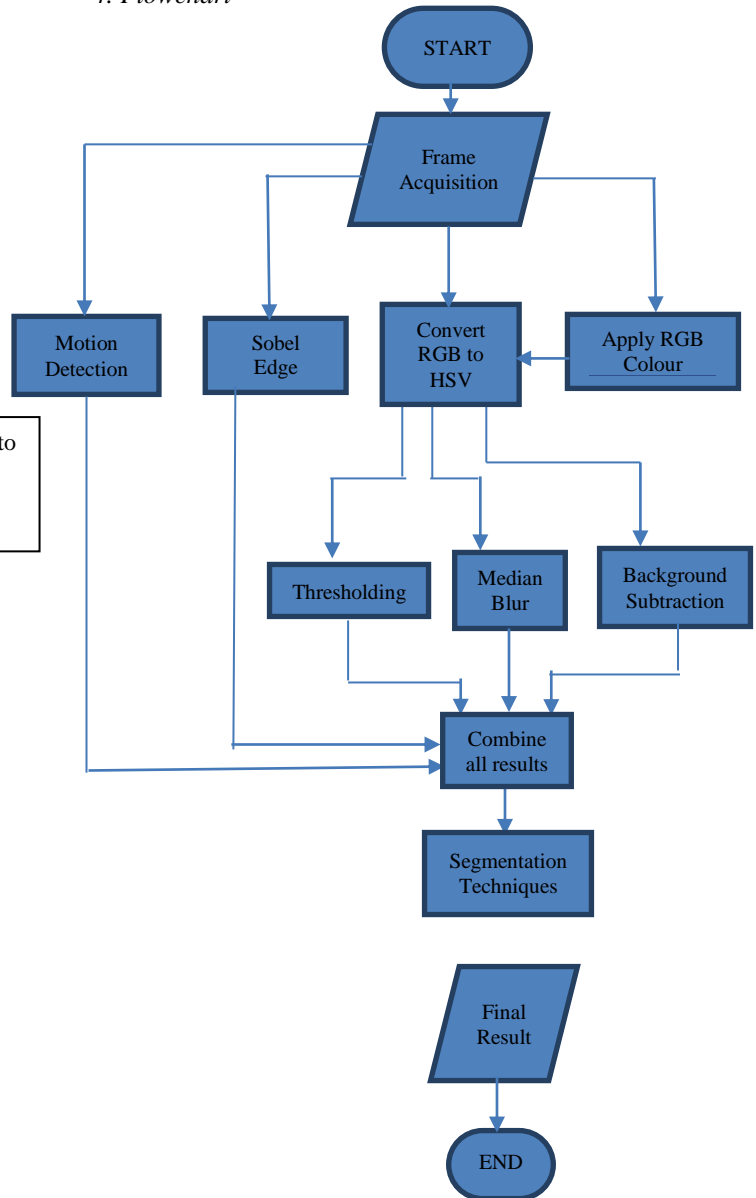


Fig 4.1: Flowchart of the proposed system

Fig 4.1 is the flow chart of our proposed system which summarizes the whole method.

5. CONCLUSION AND FUTURE SCOPE

This project proposed a fire detection algorithm which is free from sensors as the ordinary fire detection systems contain. The objective of this project was to create a system which would be able to detect fire as early as possible from a live video feed. System is expected to detect fire while it is still small and has not grown to mammoth proportions. Also, the hardware is minimal and has been already existent in places, thus saving capital. It also saves cost by getting rid of expensive temperature and heat sensors etc. Based on the results produced, the system has proven to be effective at detecting fire. This system is an amalgamation of various fire detection algorithms.

Future scope of our method is as follows:

1. The system can be made weatherproof
2. Smoke detection along with fire detection can be added as a feature
3. System Optimization and Delay Reduction i.e. lesser latency may be achieved
4. System can be used to detect forest fires and may be embedded on a drone or any other UAV for surveillance purposes of property
5. The system can have military applications
6. The system can be used for rescue operations on land and in sea

6. REFERENCES

- [1] Sonali K. M., Poojashree M., Shilpashree A., Sindhu N., Rekha K. S., "Automated Fire Detection Surveillance System", CSE, NIE, Mysuru, International Journal of Current Trends in Engineering & Research (IJCTER) e-ISSN 2455-1392 Volume 2 Issue 6, June 2016 pp. 49 – 52
- [2] OpenCV Documentation (Background Subtraction) http://docs.opencv.org/3.1.0/db/d5c/tutorial_py_bg_subtraction.html
- [3] P. KadewTraKuPong, R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection" 2001
- [4] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction" 2004
- [5] Z. Zivkovic, "Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction" 2006
- [6] OpenCV Documentation (Contour analysis) http://docs.opencv.org/trunk/d4/d73/tutorial_py_contours_begin.html
- [7] OpenCV Documentation (Structural analysis) http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html
- [8] S. Bharathi, S. Gokilapriya, N. Elango & P. Vidhya, "FIRE DETECTION AND FIRE SIGNATURE USING COLOR MODELS FOR SECURITY", International Journal of Current Research and Modern Education (IJCRME) Special Issue, NCFTECCPS – 2016
- [9] Akshay Thokale , Poonam Sonar, "Review on Vision Based Fire Flame Detection", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Issue 9, September 2015
- [10] Turgay Celik, Hasan Demirel, Huseyin Ozkaramanli, and Mustafa Uyguroglu, "Fire detection using statistical color model in video sequences," J. Vis. Comun. Image Represent., vol. 18, pp. 176–185, April 2007
- [11] Turgay Celik, "Fast and Efficient Method for Fire Detection Using Image Processing," ETRI Journal, Volume 32,

Number 6, December 2010

- [12] Rafael C. Gonzalez and Richard E. Woods: "Digital Image Processing", 2009
- [13] Raspberry Pi official website <http://www.raspberrypi.org>
- [14] OpenCV official website <http://www.opencv.org>