

# Vision Based Flame Detection System For Surveillance Camera

Dedy Slamet Riyadi  
Mechatronics Engineering  
Politeknik Negeri Batam  
Batam, Indonesia  
dedyslametriyadi@yahoo.co.id

Siti Aisyah  
Mechatronics Engineering  
Politeknik Negeri Batam  
Batam, Indonesia  
siti\_aisyah@polibatam.ac.id

**Abstract**— Flame detection systems are useful for the development of fire early detection systems. In general, conventional fire detection system which is usually used heat detector cannot detect the presence of fire since the detector requires much time to detect temperature changes. The use of a camera as an image scanner sensor has the potential to be used as an identifier of a flame shape or color. This study aims to build an image processing flame detection system using image enhancement, segmentation and filtering methods which is applied the *Vision Assistant* available in LabVIEW. The system implemented in a frame rate of 30 fps with a resolution of 1024 x 768, in which the system accuracy reach up to 98%.

**Keywords**— Flame Detection, Image Enhancement, LabVIEW, Color Filtering, HSV Filtering

## I. INTRODUCTION

Mostly every house or building has potentially to get fire accidentally. Therefore, it is necessary to develop the system that can detect fire early. The system is expected to be used to detect fires inside a house or building. In conventional fire detection systems, sensors used to detect fires are heat detectors and smoke detectors that have an important role in fire detection process [1].

Fire detection systems by image processing techniques are usually analyzing two parameters namely flames and smoke. Most of the fire detection algorithms perform pixel analysis on the properties of flame or smoke such as color, the contrast between the flame color and its background and flickering properties [2].

There are many flame detection systems using image processing in which the RGB color model is used to detect the flame color that is dominated by the intensity of red component. Some methods used Sobel edge detection to measure the flame size. The combination of those techniques has been implemented to identify Region of Interest (ROI) of flame using color-based flame segmentation techniques [3].

Some research has been conducted to develop fire detection system by combining color and motion detection as well as regional dispersion system [4]. The first step of the algorithm is to place the desired color area in the video frame, and then determine its movement and the last step is calculating the number of pixel in the color area. Other research implemented several methods to improve image analysis such as converting RGB color plan to HSV, selecting the optimum threshold value, and applying Gaussian model of ROI [5].

There is a pre-processing stage of flame detection by image processing techniques in which various image processing operations are performed on the image frame.

These operations include blurring, RGB to HSV, Grayscale, Thresholding and spectral analysis for blob detection on segmented object [6]. The pre-processing shall begin with set the thresholding value, in which thresholding result is used to get the required data information and the result would be applied to the next process that is *blob detection*. If the presence of flame is detected by system, the system will produce an alarm as an indicator.

Visualization of the image processing process is really important especially to facilitate the analysis process. LabVIEW is one of the graphical programming system that can be used for image processing. The *Vision Assistant* modul in LabVIEW could be used in order to make identification and tracking an image or object easier [7]. Image processing algorithm in LabVIEW can be implemented by using a simple block diagram rather than doing more complicated programming. LabVIEW provides an easier way to do edge detection, object tracking, histogram analysis, pattern matching and many more.

## II. RELATED WORK

### A. Image Processing

Image Processing is a method used to process or manipulated images in 2-dimensional form [8]. Image processing is a technique that aim to improve, analyze or change images. There are several method that can be used in an image processing application [8]. One of the methods is image enhancement, a method to improve image quality. While image segmentation is a method to simplify the image that has passed the enhancement process to make it easier to analyze. Furthermore, another method is features extraction, a method which is used to detect any features such as shape, size, contour, and other features of an image [9].

However, in general image processing can be classified as the following types:

1. Image Enhancement: This operation aims to improve image quality by manipulating image parameters such as brightness, contrast, sharpness
2. Image Restoration: This operation aims to minimize defects in the image. Some of the methods include de-blurring and noise elimination
3. Image Compression: The purpose of this method is to represent image incompact form without decreasing its quality and it requires less memory.
4. Image Segmentation: The purpose of this method is to split an image into several segments with a certain criteria.

5. Image Analysis: This methods aims to extract certain information of an image
6. Image Reconstruction: The purpose of this method is to reshape objects from several reference images

### B. Brightness

Brightness is part of image enhancement methods to process the brightness level of an image. The image could be set its brightness by adding the values of R, G and B of the image with a coefficient of brightness (kb). The value of kb ranges from -255 to 255. If the number of values is between -255 to 0, the result of the image would be dark. To be able to brighten or darken the color of image, the pixel value of the original color should be added to the value of kb as represented in (1).

$$\begin{aligned} R_{b(i,j)}, G_{b(i,j)}, B_{b(i,j)} &= (R_{b(i,j)} + kb), (G_{b(i,j)} + kb), (B_{b(i,j)} + kb) \\ &= (((P_{(i,j)} \bmod 2^{16}) \bmod 2^8] + kb), \\ &= \left( \left\lceil \frac{P_{(i,j)} \bmod 2^{16}}{\bmod 2^8} \right\rceil + kb \right), \left( \left\lceil \frac{P_{(i,j)}}{2^8} \right\rceil + kb \right) \end{aligned} \quad (1)$$

### C. Contrast

Enhancement Contrast is the process of sharpening the image color in which the pixel value of original image is multiplied by the Contrast coefficient (kc). The value of kc ranges from 0.1 to 1.9. If the value  $P_{c(i,j)}$  (a combination of R, G and B values that have been contrasted) less than zero ( $P_{c(i,j)} < 0$ ), then the result would be the less color sharpness. Otherwise, when  $P_{c(i,j)} > 0$  then color will be more sharpness as defined in (2).

$$\begin{aligned} R_{c(i,j)}, G_{c(i,j)}, B_{c(i,j)} &= (((P_{(i,j)} \bmod 2^{16}) \bmod 2^8] + kb) \times kc), \\ &= \left( \left( \left\lceil \frac{P_{(i,j)} \bmod 2^{16}}{\bmod 2^8} \right\rceil + kb \right) \times kc \right), \\ &= \left( \left( \frac{P_{(i,j)}}{2^8} + kb \right) \times kc \right) \end{aligned} \quad (2)$$

### D. Segmentation in HSV color space

Segmentation in the Hue, Saturation and Value (HSV) color space is the process to filter unwanted colors in order to get the required color. The HSV value can be obtained from RGB value, where each color component of RGB is divided by 255 in order to get a value of 0 and 1. The complete process to transform RGB value to HSV is denoted in (3) [10].

Color filtering can be carried out by specifying the upper and lower thresholds where  $lower\ limit \leq Threshold\ value \leq Upper\ limit$ . The range of Hue threshold defined as  $(H_{Th} - Bandwidth) \leq H_{Th} \leq (H_{Th} + Bandwidth)$ , and Sue threshold as  $(S_{Th} - Bandwidth) \leq S_{Th} \leq (S_{Th} + Bandwidth)$ , and Value threshold as  $(V_{Th} - Bandwidth) \leq V_{Th} \leq (V_{Th} + Bandwidth)$ .

$$R' = \frac{R}{255}; G' = \frac{G}{255}; B' = \frac{B}{255} \quad (3)$$

$$C_{max} = \max(R', G', B'); C_{min} = \min(R', G', B'); \Delta = C_{max} - C_{min}$$

$$H = \begin{cases} 0^0, \Delta = 0 \\ 60^0 x \left( \frac{G' - B'}{\Delta} \bmod 6 \right), C_{max} = R' \\ 60^0 x \left( \frac{B' - R'}{\Delta} + 2 \right), C_{max} = G' \\ 60^0 x \left( \frac{R' - G'}{\Delta} + 4 \right), C_{max} = B' \end{cases}$$

$$S = \begin{cases} 0, C_{max} = 0 \\ \frac{\Delta}{C_{max}}, C_{max} \neq 0 \end{cases}$$

$$V = C_{max}$$

### E. Grayscale Image

Grayscale is a step in which the image is processed and turn dark but the shape remains as before, that means the color will be darker than the previous one. The method used is by calculating the averaging value of R, G and B as defined in (4).

$$RGB_{Grayscale} = \left[ RGB \in N : \frac{R_{f(i,j)} + G_{f(i,j)} + B_{f(i,j)}}{3} \right] \quad (4)$$

### F. Binary Image

Binary image is an image with the value of 1 and 0 as the representation of black and white color. The binary results are different from the gray scale. The differences appear in the detailed shape and color. Binary process aims to emphasize the pixel values of an image. If the pixel is more than 128 then it is changed to 255, otherwise if the pixel value is less than and equal to 128 then the pixel is represented by 0 as defined in (5).

$$RGB_{Binary} = \left[ RGB_{Grayscale} \in (N = 255) : 128 \leq RGB_{Grayscale} \right] \quad (5)$$

Binary image represented by black (zero) and white (one) image is depicted in Fig. 1.

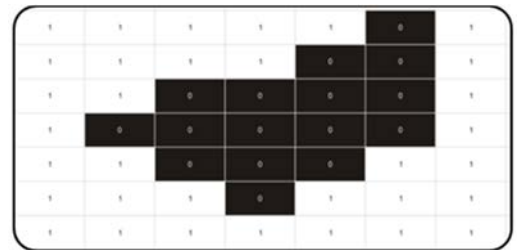


Fig. 1. Representation of binary images that occupy the dot point of the pixel in an image frame

### G. Blob Detection

Blob is a group of pixel that have neighbour relationships. Blob calculation process can be carried out by doing neighboring pixel analysis. The neighbor pixel is determined as pixels that are 1 apart from the original pixel. The blob calculation process utilizes the 8-neighbors pixel relationship as shown in Fig. 2 [11].

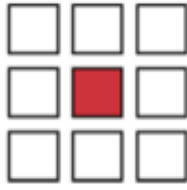


Fig. 2. 8-neighbours pixel relationship

In the Fig. 2 original pixel is brown pixel and white pixels are the neighbour pixels. Before applying the first blobs analysis, pre-processing must be implemented by changing *Grayscaleimage* (256-level image) to an image with only have two values, *Binary* (images with a value of 0 and 255). The goal is to separate important objects / blobs from the unimportant image information. A technique called thresholding can precisely separate the blobs from the background image. The result of the thresholding process is a binary image that has 1 and 0 pixel values.

### III. DESIGN APPROACH

The development of a flame detection system begins with the installation of a camera on the camera holder that has been designed using the Autodesk Inventor 2016 software. The vision sensor used in this flame detection system is the Logitech C170. The data from the camera in the RGB form captured by logitech C170 are processed on a PC / Laptop with a processor capacity of 2.40 GHz and 3.00 GB RAM Memory, in which it has been installed LabVIEW 2014. In addition, the PC is also used to send string data to the microcontroller which will later be converted into PWM form and used to control the movement of servo motors installed on the holder camera coupler. The entire detection system has shown in Fig. 3.

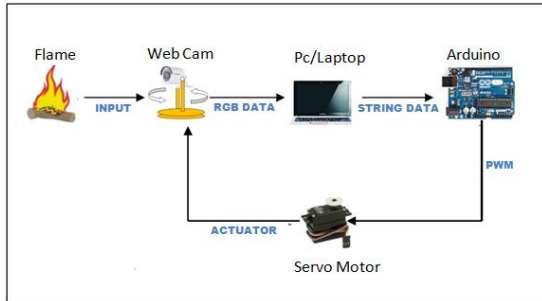


Fig. 3. Workflow of the Early flame detection system

Mechanical design of a camera holder which can move to the right and left at 90 degrees is designed using Autodesk Inventor 2016. To get optimal results, it is necessary to give more attention to the mechanical system that has been designed, because the mechanical aspect is one of the important factors in process for designing this device. The design that has been made is shown in Fig. 4

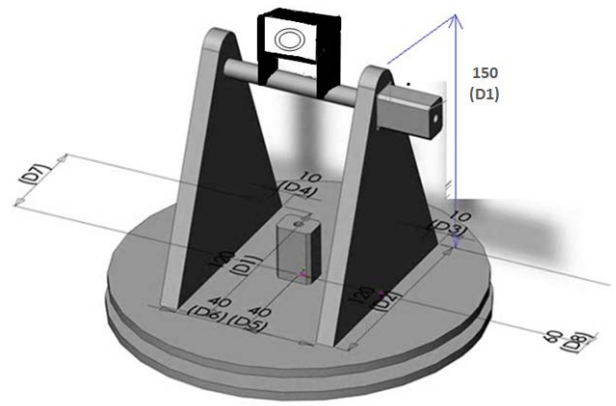


Fig. 4. Camera holder designed in millimeters scale

A digital webcam camera will capture images in real time and sent to a laptop to be processed through LabVIEW software. By using LabVIEW software, the captured image is processed and flame position is identified. Image data from the camera is segmented in the HSV color space and the thresholding process is carried out as filtering fire objects with other objects.

Image data that has been thresholded using the HSV color space is used for the bolb detection process to mark the flame position. If a flame is detected, the camera's holder stops moving and produces an alarm as fire presence indication.

In the software design, it takes several steps to design the program on the LabVIEW software and the Arduino IDE software as represented ini Fig. 5.

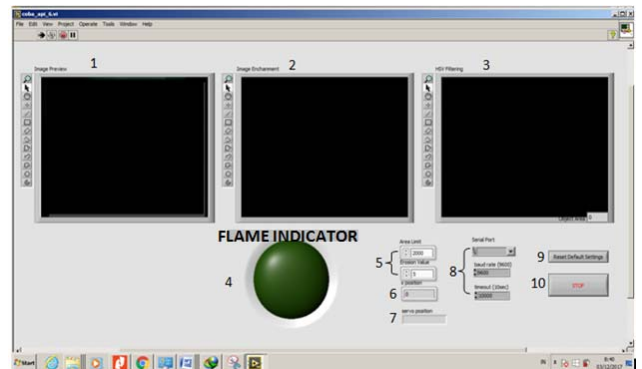


Fig. 5. User Interface Design of Flame Detection System

The design of the GUI Interface is shown in Fig. 5, where:

1. Actual images display
2. Image display after enhancement process
3. Image display after segmentation process
4. Indicator lamp as flame indicator
5. Threshold parameter setting
6. The coordinate pixels of flame position
7. Actual servo position in degrees
8. Serial port PC settings to microcontroller
9. Reset button to return to the default settings
10. Stop Button to stop the running program

In the purposed image processing methods, it has several steps for the pre-processing stage, as illustrated in Fig. 6.

The blob detection results from the object detection program are displayed in the User Interface. The detected object in the certain area is given a red bounding box to indicate where the position of the flame as shown in Fig. 11.



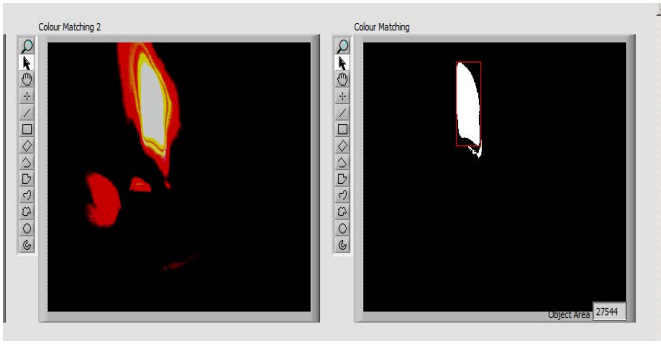


Fig. 11. ROI of flame area with a red rectangle line

#### IV. RESULTS AND DISCUSSION

##### A. Flame Detection System Testing in Different Distance of Camera and the Object

In this flame detection program, it is given the threshold value to the detected object area since not all detected objects are flame. If the number of particles in an area exceeds the threshold value of 2000 pixel units, the program will automatically give a ROI / bounding box to that area. Fig. 12 is the result data of flame detection system where the camera is placed in certain distance from the flame.

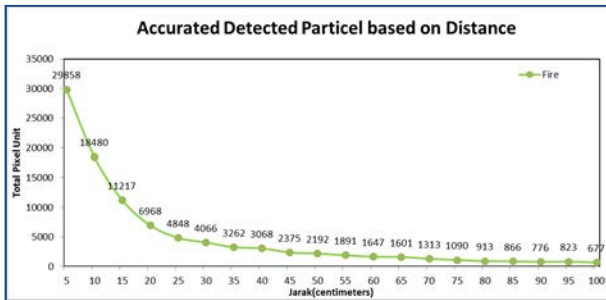


Fig. 12. The pixel detected in the certain distance of camera and the flame

From the test results shown in Fig. 12, it can be seen that the number of pixel detected changes as the distance change. When the object is farther away from the camera, the number of object pixel detected is getting smaller.

##### B. Flame Detection System Testing in Different Light Intensity

To be able to find out the effectiveness of the detector system, it is necessary to test at different levels of lighting at different times. In order to approach the real conditions, the test is carried out within one day with sampling at each condition with different lighting intensities, namely in the morning, afternoon, evening and night as well as when the lights lamp are off or on. This condition is considered to be sufficient to represent the function of the camera that works in a room for 24 hours non-stop.

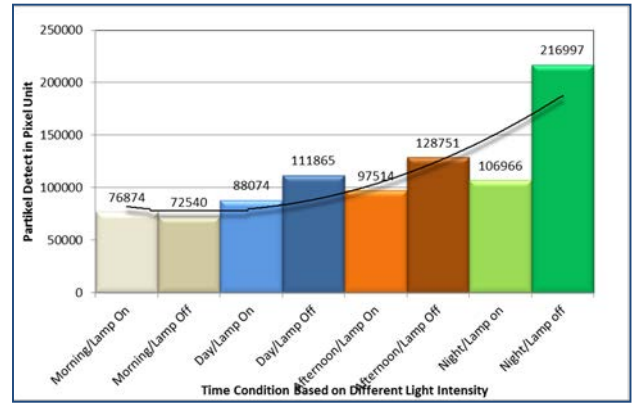


Fig. 13. The number of particles detected on a fire object in each lighting condition is different

From Fig. 13, the real object detected by the program is at most in the night and when the lamp is off. These results are obtained when the lighting conditions are minimal and the flame image has a good contrast with its background. Whereas the least detected object occurs during the morning and the lamp is in an off position. These results are obtained when the light intensity is quite high so that the flame is not too flashy to detect.

##### C. Flame Detection System Testing in Difference Kind of Object

This stage is expected the program can detect objects accurately. Therefore it is necessary to test the detection of real flame objects with objects that have an illumination pattern that almost resembles as flame. The sample object which have same properties with flame are the red LED light, the yellow / orange LED light, the white LED and the bulb. All sample data testing is taken with the camera condition pointing directly to the light source.

TABLE I. FLAME DETECTION RESULT WITH A THRESHOLD OF 2000 PIXEL UNITS IMPLEMENTED IN DIFFERENT SAMPLE OBJECTS

Experiment	Distance	Flame	Bulb	White LED	Red LED	Yellow LED	Detection Result
1	5	29858	16929	39028	35348	15043	TRUE
2	10	18480	12726	32290	7755	9739	TRUE
3	15	11217	20638	19149	3292	5584	TRUE
4	20	6968	7608	10237	2599	1734	TRUE
5	25	4848	7608	4287	1464	1077	TRUE
6	30	4066	7397	3691	1512	1041	TRUE
7	35	3262	7276	3251	1106	955	TRUE
8	40	3068	6809	3309	1009	697	TRUE
9	45	2375	6118	2716	911	655	TRUE
10	50	2192	5346	2300	629	452	TRUE
11	55	1891	5365	2012	570	403	TRUE
12	60	1647	4945	1702	477	346	TRUE
13	65	1601	5528	1496	399	320	TRUE
14	70	1313	5858	1374	313	0	TRUE
15	75	1090	6749	1213	0	0	TRUE
16	80	913	6450	1073	0	0	FALSE
17	85	866	5115	987	0	0	FALSE
18	90	776	6001	927	0	0	FALSE
19	95	823	4960	838	0	0	FALSE
20	100	677	2521	789	0	0	FALSE

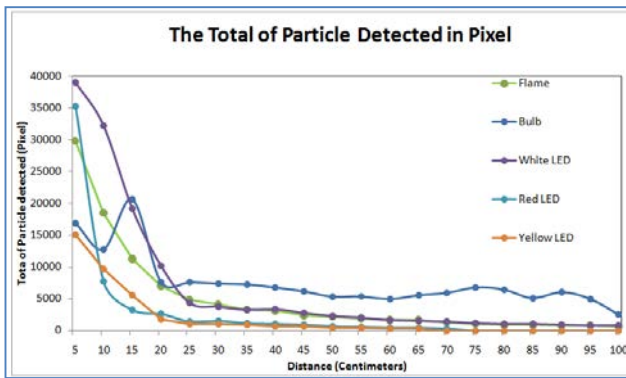


Fig. 14. Pixel detected in different object which have same properties with flame

Test results shown in the Table 1 and Fig. 14 represent that different kind of object with the same properties as flame would detect as flame in some condition depend on several reason, such as the set thresholding value and the testing condition. Each sample object has a total number of particles pixel detected differently. In the data graph of the bulb, the number of particles soared at a point of 15 cm, due to the object holder shifts during the testing and it changes the angle and direction directly towards the camera. When the light bulb is pointing directly towards the camera, the particles pixel detected are going to increase. This is because the bulb's light has a function to collect luminous light around the lamp so that the light beam becomes centered. This flame detection system utilizes one of the properties of flame as a source of light, that is if the detected object has a glowing light around it then the program will accumulate the total pixel of the object.

The overall program system is still unable to distinguish between light sources and real flame. Because the purpose detection system is utilizing the lumiance properties of a flame which can emit light in its coverage area so that it is different from the surrounding object. While the LED and bulb also possess these properties so the program is difficult to distinguish them.

## V. CONCLUSION

Flame can be detected using HSV color filtering by utilizing color patterns and intensity. The program is able to detect and differentiate the real flame and the fake one, but it is still not accurate, since some pixels of light source objects are detected as flame until the end of the segmentation process. The optimum detection of flame can be obtained by a condition where the light is off.

## REFERENCES

- [1] Khosro Rezae, S.Jalal Mousayirad, M.Rasegh, Javad Haddania, "Accurate Fire Detection System for Various environment using Gaussian Mixture Model and HSV Space", Journal of Information System and Telecommunication, Vol. 1, No. 1, Jan-March 2013.
- [2] Leonardo Millan-Garcia, Gabriel Sanchez, Mariko Nakano and Luis Rojas Cardenas, " An Early Fire Detection Algorithm Using IP CAmeras", MDPI journal Sensor,Electric Engineering Dept, Metropolitan Autonomous University, Mexico, 2012.
- [3] Kumarguru Poobalan, Siau-Chiun Liew, "Fire Detection Algorithm Using Image Processing Techniques", Faculty of Computer Science and Software Engineering, 3rd International Conference on AI and Computer Science (AICS2015), University Malaysia Pahang, Penang, Malaysia 2015.
- [4] Punam Patel, Shamik Tiwari, "Flame Detection Using Image Processing Techniques", International Journal of Computer Applications, MITS, Laxmangarh, 2012.
- [5] Khosro Rezae, S.Jalal Mousayirad, M.Rasegh, Javad Haddania, "Accurate Fire Detection System for Various environment using Gaussian Mixture Model and HSV Space", Journal of Information System and Telecommunication, Vol. 1, No. 1, Jan-March 2013.
- [6] Prof.S.M Rajbhoj, Ruchita A, Nishigandha N, Aishwarya A, "Real time Fire Detection for Video Surviellance Using Digital Image Processing", International Journal of Advance Research in Electrical, Electronics and Instrumentation Engineering, Vol. 5, Issue 3, March 2016.
- [7] Mirji Sairaj, Madral Hariprasad, Jawalkar Ashis, "A Review on Image Tracking Technique in LabVIEW", International Journal of Scientific Development and Research (IJS DR), Vol. 1, Issue 6, June 2016.
- [8] Gonzales, Rafael and Richard E.Woods, "Digital Image Processing (3rd Edition)", Pretince Hall, USA, 2008.
- [9] Kasdianto, Siti Aisyah, "Desain Sistem Pendeteksi untuk Citra Base Sub-Assembly denganAlgoritma Back propagation", Jurnal Rekayasa ElektriKa, Vol.13, No.1, April 2017.
- [10] Martin Loesdau, Sébastien Chabrier, Alban Gabillon, "Hue and Saturation in the RGB Color Space", Proceedings of 6th International Conference in Image and Signal Processing (ICISP), pp.203-212, Cherbourg, France, 2014.