# A New Edge Detection Algorithm for Flame Image Processing

Tian Qiu, Yong Yan* FIEEE, Gang Lu SMIEEE

Instrumentation, Control and Embedded Systems Research Group
School of Engineering and Digital Arts, University of Kent, Canterbury, Kent CT2 7NT, UK
t.qiu@kent.ac.uk, y.yan@kent.ac.uk, g.lu@kent.ac.uk

*Abstract*— **Digital image processing is playing an increasingly important part in imaging based flame monitoring systems. A crucial step in flame image processing is to detect the flame edge, a dividing boundary between the area where there is thermo-chemical reaction and those without. The determination of flame edges is a precursor to flame image processing and measurement of flame parameters. Several known edge detection methods have been tested to identify flame edges but the results achieved are disappointing. As a result of recent work in feature analysis of flame images, a novel flame edge detection method has been developed, which can detect the flame edges effectively and efficiently.**

***Keywords- image processing; combustion; flame image; edge detection***

## I. INTRODUCTION

To meet the stringent standards on combustion efficiency and pollutant emissions, flame monitoring with a quantitative means is becoming increasingly important in foil-fuel combustion systems, particularly in power generation plants [1]. This has led to a wave of research in advanced flame imaging technologies [2, 3], not only in the power generation industry, but also in fire safety engineering [4].

Edge detection is one of the important steps in flame image processing. There are several reasons why it is necessary to identify flame edges. First, flame edges determine the region of a flame. This is especially important in the 3-D reconstruction of the flame as without clearly defined edges, the flame cannot correctly be reconstructed. Second, the use of flame edges can reduce the amount of processing data and filter out unwanted information, such as background noise within the image. In other words, the edge detection can preserve the important structural properties of the flame and meanwhile shorten the processing time. Third, the edge detection can be used to segment a group of the flames. This is particularly important for the multiple flame monitoring in practical furnaces where a multi-burner system is normally used. Finally but most importantly, the flame edges form a basis for the quantitative determination of a range of flame characteristic parameters.

A number of methods have been reported for extracting flame edges for the geometric characterization of the flame [5, 6], and for the determination of fire location [7, 8]. A three-dimensional (3-D) flame monitoring systems has also been reported where three identical color CCD cameras together with optical transmission units were used to concurrently capture three images of a flame from different perspectives [5]. The outer edges of the flame images were detected and the information obtained were used to build the geometric model of the flame and thus to determine the size, volume and location of the flame.

In this research several edge detection methods are examined to assess their effectiveness for flame edge identification. Despite many parameters are delicately adjusted in the use of these methods, flame edges still cannot be identified. It is therefore desirable to develop a dedicated edge detection method for flame image processing. As a result of recent work in characteristic analysis of flame images, a novel computing algorithm is proposed where some of the unique features of a flame image such as the number of main objects and strong comparative brightness are used to identify flame edges. Adopting these features, one can detect the coarse and superfluous edges in a flame image and then identify the flame main edges and remove the unrelated edges.

## II. BASIC METHODS OF EDGE DETECTION AND THEIR APPLICATIONS TO FLAME IMAGES

A typical edge in an image might for instance be the border between blocks of different colors. The basic edge detection method is to determine the level of variance between different pixels by executing edge-detection operators for each small matrix area. The edge-detection operator is calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the value of this matrix area is above a given threshold, then the middle pixel is classified as an edge.

Mathematically, the edges are represented by first- and second-order derivatives. The first-order derivative (i.e., gradient) of a 2D function, f(x, y), is defined as a vector [9]:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial y}{\partial x} \\ \frac{\partial y}{\partial x} \end{bmatrix}, \tag{1}$$

where $G_x$ and $G_y$ are the gradients in the x and y coordinates, respectively. The magnitude of the vector is given by:

$$\mathrm{mag}(\nabla f) = \sqrt{{G_x}^2 + {G_y}^2} = \sqrt{(\frac{\partial y}{\partial x})^2 + (\frac{\partial y}{\partial x})^2}\ . \tag{2}$$

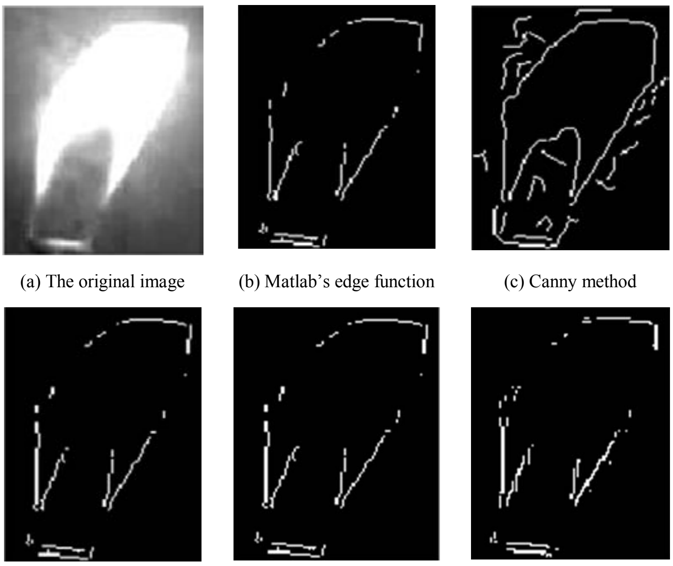
The angle α, at which the maximum rate of change occurs, is:

$$\alpha(x, y) = \tan^{-1}(\frac{G_y}{G_x})\ . \tag{3}$$

* Corresponding author

978-1-4244-7935-1/11/$26.00 ©2011 IEEE

All the gradient-based algorithms have kernel operators that calculate the strength of the slope in directions which are orthogonal to each other, commonly vertical and horizontal. Examples of gradient-based edge detection operators include Roberts, Prewitt, and Sobel operators. Then, contributions of the different components of the slopes are combined to give the total value of the edge strength. As an improved method using the Sobel operator, the Canny edge detection algorithm is known as an optimal edge detector [10]. In the present research, these common edge detection methods have been attempted with adequate parameters to process typical flame images. Figure 1 shows example results obtained along with the original image. As can be seen, those common edge detection methods can only identify a part of the flame edges, or wrongly identified small edges that are obviously not the edges of the main flame.



(a) The original image (b) Matlab's edge function (c) Canny method

(d) Prewitt method (e) Sobel method (f) Roberts method

Figure 1. Representive results using the common edge detection methods.

## III. A Novel Edge Detection Algorithm For Flame Image Processing

Since flame images are a special class of images, some of the unique features of a flame may be used to identify flame edges. There are some differences between flame images and other general images: flames are the main objects in the images; the brightness of the flame is generally much higher than the other objects while the background is comparatively dark; furthermore, there is generally only one flame or one group of flames in one image. The expected flame edge should be one and only one clear, continuous, and uninterrupted edge. Adopting these features, we have developed a novel flame edge detection algorithm. The algorithm can be divided into several logical steps as follows:

Step 1: Adjust the gray level of a flame image according to its statistical distribution.

Step 2: Smooth the image to eliminate noise.

Step 3: Use the Sobel operator to find basic edges. This is achieved by finding the gradients of all the pixels in the image so as to highlight the regions with high gray level contrast at their edges. The algorithm then tracks along those regions and suppresses any pixels that are not at the peaks of gradients. If the magnitude of the gradient is above the high threshold $T_H$, it is deemed an edge. And if the magnitude is between the two thresholds, i.e., high threshold $T_H$ and low threshold $T_L$, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above $T_L$.

The Sobel operator performs a two dimensional spatial gradient measurement on an image. Then the approximate absolute gradient magnitude (edge strength) at each point can be found. It uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and another estimating the gradient in the y-direction (rows). The Sobel operator is expressed as follows [11]:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{4}$$

$$M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{5}$$

Step 4: Adjust $T_H$ and $T_L$ to get a better result. This is done by giving the first pair of $T_H$ and $T_L$ initial values according to the apriori results of similar flame images, and then adjusting the values for a better result. The 'better' result is assessed by that how many edges there are: the more edge pixels detected in the edge image, the better the parameters are. Another threshold $T_E$ is also set to restrict the total number of edges, i.e., if the number of edge pixels exceeds $T_E$, the automatic adjustment will be terminated. Until now, a preliminary image with edges identified is obtained from the original flame image. It is designated as a Preliminary Edge Image (PEI).

Step 5: Remove unrelated edges in the PEI through the following steps;

5a) Select any edge point in the PEI, remove that point from the PEI, allocate a new temporary edge image and plot the point onto the temporary edge image.

5b) Use the selected point as the center and search in a 3×3 area. Store the location of all the neighboring pixels if they are edge pixels. In eight neighboring pixels, operations are taken for the following three different cases,

— If there is no neighboring pixel, the selected point is then an isolated point, and should be removed from the PEI. Terminate the search and go to Step 5d.

— If there is one neighboring pixel, the selected point is an end point. It should then be removed from the PEI, plotted onto the temporary edge image, and added into the endpoint list. Start the new search from the found neighbor and go to Step 5c.

— If there are more than two neighboring pixels, then the selected point is a normal transition point in an edge line or an intersection with more than three bifurcations. Set one of the neighboring points as the new search center and start a search. Store the other positions as unchecked conjunction points, and then go to Step 5b.

Figure 2 illustrates how the tracing step moves forward if the old search center is replaced by a new search center. In the pixels of the left image in Figure 2, pixel ‘5’ is the center selected. Suppose an edge point at pixel ‘9’ is found, then remove pixel ‘5’ from the PEI to temporary edge image, and pixel ‘9’ will be the new search center. In this way, the search moves forward pixel by pixel.

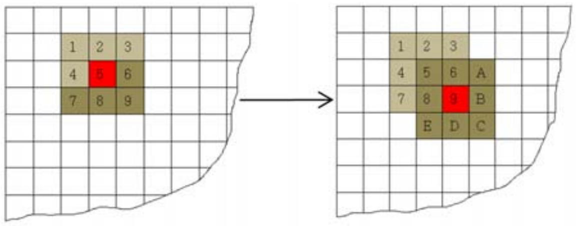



Figure 2. Illustration of movement of the edge search.

5c) Check the conjunction points. If all the conjunction points have been searched as a center, one temporary edge image is then completed. Compute the lengths of any two end points in the temporary edge image and pick out the longest one. Then go to Step 5d.

5d) If all the pixels in the PEI are moved to the temporary edge image, then go to Step 6.

Step 6: Plot the pixels of the longest edge in the final edge image which should have the same size as the original image. The whole process is then complete.

The flow chart of the whole process is given in Figure 3.

## IV. Experimental Results and Analysis

After implementing the algorithm as described in Section III, hundreds of flame images were processed using the algorithm so as to evaluate its effectiveness. The images used were taken for propane Bunsen flames burning in open air. The results have shown that clear flame edges have successfully been identified in all the flame images. Figure 4 shows typical processed flame images with edges identified. In comparison with the test results presented in Figure 1, it can be clearly observed that the developed algorithm can successfully detect the clear edges of the flame and disregard unrelated small edges, which the common edge detection methods cannot do. It makes much easier to distinguish the flame region from the background. The algorithm can also be used to extract the edges of complex flames such as turbulent diffusion flames or flames of pool fires [12]. The clearly defined flame edges will form a basis for subsequent processing of the flame images such as flame size computation, flame background removal, and determination of other flame parameters [3].

Although the algorithm appears to be long, it is not tedious in computation. Tested on a desktop computer with 2.66GHz Intel® Quad CPU, it can detect 180 flame edges of 141×161 pixels flame images in one minute.

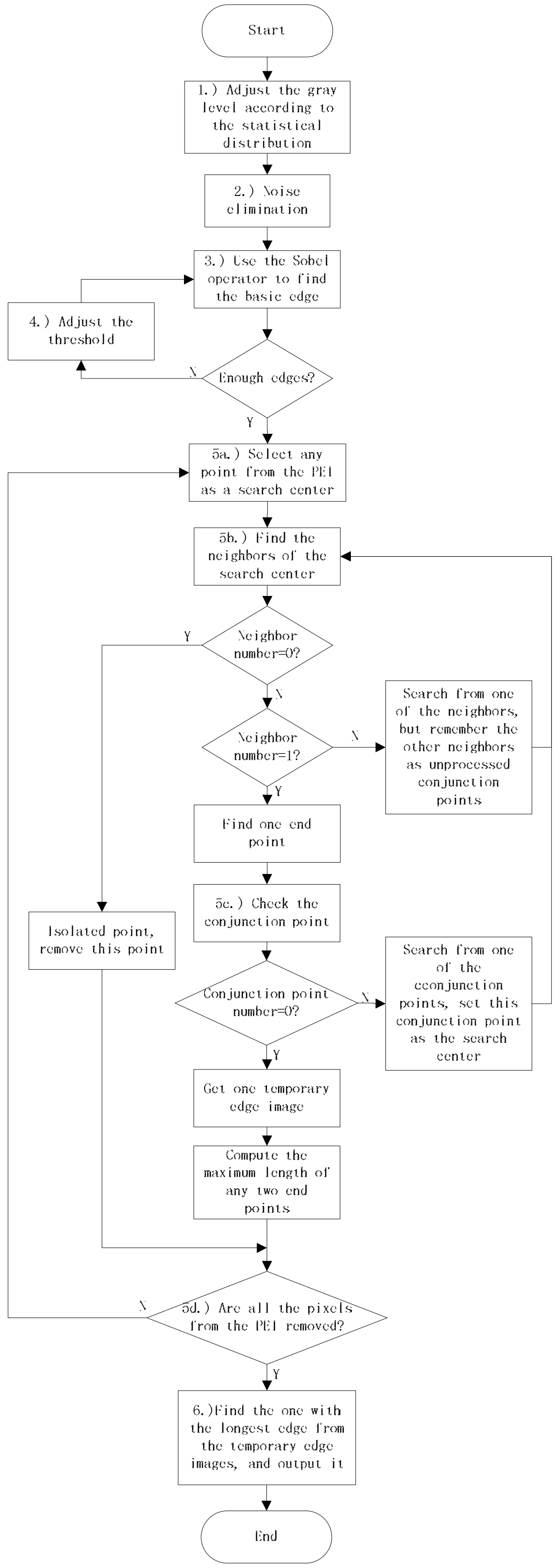



Figure 3. Flow chart of the flame edge detection algorithm.

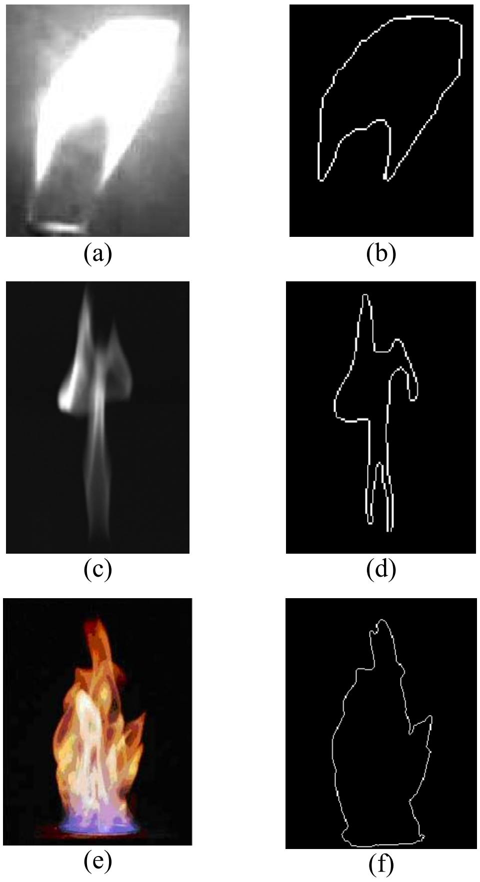



Figure 4. Some of the flame edge detection results [Left column: original images; right column: images with identified edges. (a) Diffusion pronane-flame, (c) Partially-premixed pronane-flame, (e) Small scale pool fire[13]].

## V. CONCLUSIONS

After flame characteristics are analyzed, a new flame edge detection method has been developed and evaluated in comparison with conventional methods. Experimental results have demonstrated that the algorithm developed is effective in identifying the edges of complex and irregular flames in noisy images. This reasonably fast and convenient flame edge detection method lays a good foundation for subsequent quantification of flame parameters and 3D flame reconstruction and visualization.

## ACKNOWLEDGMENTS

Acknowledgments are made to the RCUK's Energy Programme for grants (EP/G062153/1 and EP/G063214/1) in aid of this research, but the views expressed are those of the authors, and not necessarily those of the RCUK. The Energy Programme is an RCUK cross-council initiative led by the EPSRC and contributed to by ESRC, NERC, BBSRC and STFC.

## REFERENCES

[1] D. Roddy, Advanced Power Plant Materials, Design and Technology, Woodhead Publishing, ISBN 1 84569 515 1, 2010.

[2] G. Lu, Y. Yan, and M. Colechin, "A digital imaging based multi-functional flame monitoring system", IEEE Trans. Instrum. Meas., Vol. 53, No. 4, pp. 1152-1158, 2004.

[3] Y. Yan, T. Qiu, G. Lu, M. M. Hossain and G. Gilabertet, "Recent advances in 3D flame tomography", Proceedings of the 6th World Congress on Industrial Process Tomography, Beijing, China, pp. 1530-1539, September, 2010.

[4] D. S. Huang, L. Heutte, and M. Loog, "Real-time fire detection using camera sequence image in tunnel environment", ICIC 2007, LNCS 4681, pp. 1209–1220, 2007.

[5] G. Lu, G. Gilabert and Y. Yan, "Vision based monitoring and characterisation of combustion flames", Journal of Physics: Conference Series 15, pp. 194–200, 2005.

[6] H. C. Bheemul, G. Lu, and Y. Yan, "Three-dimensional visualization and quantitative characterization of gaseous flames," Meas. Sci. Technol.,vol. 13, No. 10, pp. 1643–1650, 2002.

[7] B. C. Ko, K. H. Cheong, and J. Y. Nam, "Fire detection based on vision sensor and support vector machines ", Fire Safety Journal, Vol. 44, pp. 322–329, 2009.

[8] Q. Jiang, and Q. Wang, "Large space fire image processing of improving canny edge detector based on adaptive smoothing", 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering, 2010.

[9] R. C. Gonzalez and R. EWoods, Digital Image Processing, Second Edition, Prentice Hall, 2002.

[10] T. Steinhaus, S. Welch, R. Carvel, and J. L. Torero, "Large-scale pool fires", Thermal Science Journal, Vol. 11, No. 3, special on fire, 2007.

[11] J. Canny, "A computational approach to edge detection", IEEE Transactions on PAMI, Vol. 8, No. 6, pp. 679–698, 1986.

[12] K. R. Castleman, Digital Image Processing, Prentice Hall, 1995.

[13] SWRI, "Small-scale liquid pool fire characterization", Available at: http://www.swri.org/4org/d01/fire/firetech/about.htm.