

Title: An Auto-adaptive Edge Detection Algorithm for Flame and Fire Image Processing

Authors: Tian Qiu

Yong Yan (Corresponding author), Fellow of IEEE

Gang Lu, Senior Member of IEEE

Address: Instrumentation, Control and Embedded Systems Research Group

School of Engineering and Digital Arts

University of Kent

Canterbury

Kent CT2 7NT

UK

Tel: 00441227823015

Fax: 00441227456084

Email: t.qiu@kent.ac.uk, y.yan@kent.ac.uk, and g.lu@kent.ac.uk

## ABSTRACT

The determination of flame or fire edges is the process of identifying a boundary between the area where there is thermo-chemical reaction and those without. It is a precursor to image-based flame monitoring, early-fire detection, fire evaluation, and the determination of flame and fire parameters. Several traditional edge detection methods have been tested to identify flame edges but the results achieved have been disappointing. Some research work related to flame and fire edge detection were reported for different applications, however, the methods do not emphasize the continuity and clarity of the flame and fire edges. A computing algorithm is thus proposed to define flame and fire edges clearly and continuously. The algorithm detects the coarse and superfluous edges in a flame/fire image first, and then identifies the edges of the flame/fire and removes the irrelevant artifacts. The auto-adaptive feature of the algorithm ensures that the primary symbolic flame/fire edges are identified for different scenarios. Experimental results for different flame images and video frames proved the effectiveness and robustness of the algorithm.

**Keywords**– Flame; Fire; Edge detection; Image edge analysis; Image processing; Feature extraction; Monitoring; Shape measurement.

## I. INTRODUCTION

To meet the stringent standards on combustion efficiency and pollutant emissions, quantitative flame monitoring is becoming increasingly important in fossil-fuel fired combustion systems, particularly in power generation plants [1]. This has led to a wave of research in advanced flame imaging technologies [2, 3] both in the power generation industry and in laboratory research. In fire safety engineering, flame image processing is also emphasized as image-based flame detectors are increasingly applied in fire detection systems [4-9]. Compared to conventional flame detectors such as those based on optical sensing, ionization current detection and thermocouple, image-based flame detectors are deemed more

appropriate in fire detection because of their capability for remote detection of a small sized fire, as well as having other advantages [10].

As one of the important steps in flame and fire image processing, edge detection is often the precursor and lays a foundation for other processing. There are several reasons why it is necessary to identify flame edges. First, the flame edges form a basis for the quantitative determination of a range of flame characteristic parameters such as shape, size, location, and stability. Second, the definition of flame edges can reduce the amount of data processing and filter out unwanted information such as background noise within the image. In other words, edge detection can preserve the important structural properties of the flame and meanwhile shorten the processing time. Third, edge detection can be used to segment a group of flames. This is helpful for multiple-flame monitoring in some industrial furnaces where a multi-burner system is used. Furthermore, timely determination of flame edges can trigger a fire alarm and provide the fire-fighters with information on fire type, combustible substances, and exterior of the flame etc. For instance, the movement of a detected flame edge can be used to distinguish real and false fire alarms [11].

A number of methods have been reported for identifying flame edges for the geometric characterization of a flame [12, 13] or fire [14, 15]. Adkins [16] developed a software tool to analyze fire images, with which one can use a mouse to trace the flame edge. It is a manual edge detection method, but it does show the importance and usefulness of the flame/fire edge detection. Bheemul et al. [13] introduced an effective method to extract flame contours by detecting the changes of the brightness in the horizontal direction line by line over a flame image, but the method is only suitable for simple and steady flames. Zhang et al. [5] presented a new method using FFT and Wavelet transform for the contour analysis of forest fire images on a video. Lu et al. [6] proposed an algorithm

for early fire detection and test it on video clips. Toreyin et al. [7, 8, 11] succeeded in detecting the fire in a real-time video using different methods such as hidden Markov models and wavelet transform. Chacon-Murguia and Perez-Vargas [9] managed to detect and analyze fire information on a video through the analysis of shape regularity and intensity saturation feature. Razmi et al. [10] used a background subtraction and Prewitt edge detection approach to detecting flames for fire protection systems. She and Huang [17] proposed a C-V active contour model for the edge detection of flames in a power plant. Qin and Wang [15] also demonstrated an improved Canny edge detector which was used to detect moving fire regions in large space fire images. Although each of these methods has its own advantages for the given tasks, such as fire detection or shape reconstruction in a complex background, or helping to detect an early fire and trigger a fire alarm, they have some limitations. For instance, some flame edges detected are unclear, discontinuous, or do not well match the actual flame shape. For the purpose of detecting the flame's size and shape, and consequently, the geometric characteristics, it is necessary to attain the clear, continuous, and where possible, closed edge of the flame.

In this research several conventional edge detection methods have been examined to assess their effectiveness in flame edge identification. Despite the delicate adjustment of many parameters in the use of these methods, results were still unsatisfactory. Edges extracted from non-trivial images are often hampered by fragmentation, meaning that the edge curves are not connected, edge segments are melted, or false edges that do not correspond to significant phenomena in the image are shown. It is therefore desirable to develop a dedicated edge detection method for flame and fire image processing. Accordingly, a new computing algorithm is proposed in this paper to process a combustion image and to identify flame/fire edges.

Section I of this paper is a background introduction and a brief literature review of flame edge detection methods. Section II discusses related edge detection methods and their application to flame images. Section III proposes a new methodology of detecting edges of flame images with a detailed description of each step. Section IV presents experimental results and gives some examples of how to use the detected edges for characterizing flames. Concluding remarks and a scope for further research is given in Section V. The basic methodology that is applied to develop the algorithms, together with preliminary results, was reported at the 2011 IEEE International Instrumentation and Measurement Technology Conference [25]. This paper presents a detailed description of the methodology that has been developed along with the improvement of the algorithm, more experimental results and detailed discussions.

## II. CONVENTIONAL 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 or different gray levels. 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 vector [18]:

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

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

$$\text{mag}(\nabla f) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}. \quad (2)$$

The angle  $\alpha$ , at which the maximum rate of change occurs, is:

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

Generally the variance of the gray level is calculated with one of these edge-detection operators, or kernel operators. The slopes in the x- and y-directions are combined to give the total value of the edge strength. The edge-detection operator is then 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 [18].

The edge detection methods that have been published may be grouped into two categories according to the computation of image gradients, i.e., the first-order or second-order derivatives. In the first category edges are detected through computing a measure of edge strength with a first-order derivative expression. Examples of gradient-based edge detection operators include Roberts, Prewitt, and Sobel operators [23]. The Canny edge detection algorithm [20], an improved method using the Sobel operator, is known to be a powerful edge detection method. In the second category, edges are detected by searching a second-order derivative expression over the image, usually the zero-crossings of the Laplacian or a non-linear differential expression.

In the present research, these common edge detection methods have been applied with appropriate parameters to process typical flame images. Despite many parameters being finely and appropriately adjusted in the use of these methods, flame edges could not be clearly identified. Fig. 1 (a)-(f) shows examples of results obtained by the conventional edge detection methods along with the original image. The expected flame edge should be one and only one clear, continuous, and uninterrupted edge. However, as the results have shown, the edges identified using these methods are often disconnected and fragmented (Fig.1 (b)-(f)); some of the methods can only identify a part of the flame edge (Fig.1 (b)-(d)), or wrongly identify small edges that are obviously not the edges of the main flame (Fig.1 (e)).

The results have therefore suggested that it is not always possible to obtain ideal edges from real life images of moderate complexity, thus complicating the subsequent task of interpreting the image data.

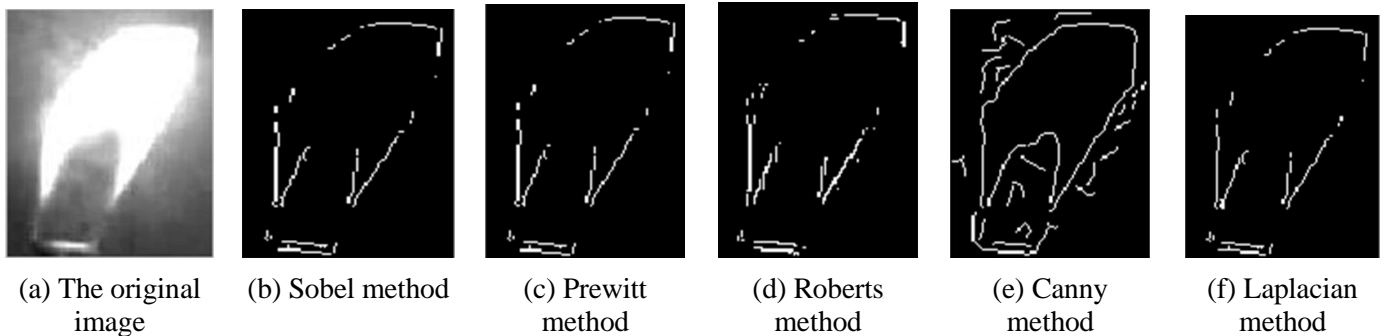


Fig. 1 Representative results using the common edge detection methods and Laplacian method.

There are some other algorithms proposed for the flame/fire edge detection for various applications [10-17]. Although we are unable to test all these methods, the published results have shown that these methods are not suitable for our purpose. It is therefore desirable to develop a dedicated edge detection method for flame/fire image processing.

### III. A NEW EDGE DETECTION ALGORITHM FOR FLAME IMAGE PROCESSING

In general, a flame region has a stronger luminance in comparison to its ambient background, and the boundary between the flame region and its background is mostly continuous. Furthermore, in most cases, there is only a main flame in the image; otherwise, the image can be segmented so that each segmented area contains only one main flame. Accordingly, a computing algorithm is proposed where these features are used to identify flame edges. The basic strategy is to detect the coarse and superfluous edges in a flame image, then identify the flame's principle edges and remove irrelevant ones. The algorithm can be divided into following logical steps:

*Step 1: Adjusting the gray level of a flame image.*

The first step is to adjust the gray level of a flame image according to its statistical distribution. Consider a discrete grayscale image  $\{x\}$  and let  $n_i$  be the number of occurrences of gray level of  $i$ , the probability of the occurrence of a pixel of gray level  $i$  in the image is [21]:

$$P_x(i) = p(x = i) = \frac{n_i}{n}, 0 \leq i \leq L, \quad (4)$$

where  $L$  is the total number of gray levels in the image,  $n$  the total number of pixels in the image, and  $p_x(i)$  the histogram for pixels with  $i$ , normalized to  $[0,1]$ . Also the cumulative distribution function (CDF) corresponding to  $p_x$  can be defined as:

$$CDF_x(i) = \sum_{j=0}^i p_x(j), \quad (5)$$

which is also the accumulated normalized histogram of the image.

Next, create a transformation of form  $y = T(x)$  to produce a new image  $\{y\}$ , such that its CDF will be linearized across the value range with a constant number  $K$ , i.e.

$$CDF_y(i) = iK. \quad (6)$$

To map the values back to their original range, the following transformation is applied to the result:

$$y' = y \times (\max\{x\} - \min\{x\}) + \min\{x\}. \quad (7)$$

*Step 2: Smoothing the image to eliminate noise.*

The second step is to filter out any noise in the image before locating and detecting any edges. A Gaussian filter can be achieved using a simple mask. Gaussian smoothing [26] is performed using standard convolution methods after a suitable mask is selected. The larger the width of the Gaussian mask, the lower the detector's sensitivity to the background noise in the flame/fire image, but a large mask may also make the detected flame/fire edge so precise that the localization error in the detected flame/fire edges also increases slightly with the Gaussian width. After certain tests and comparison the Gaussian mask, as shown in Fig.2, is used in the implementation.



	2	4	5	4	2
	4	9	12	9	4
$\frac{1}{115}$	5	12	15	12	5
	4	9	12	9	4
	2	4	5	4	2

Fig. 2 Discrete approximation to Gaussian function.

*Step 3: Using the Sobel operator for finding basic edges.*

Finding basic edges 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 the edge along these regions and suppresses any pixels that are not at the peaks of the gradients. If the magnitude of the gradient is above high threshold  $T_H$ , it is deemed an edge. And if the magnitude is between the two thresholds, i.e., the  $T_H$  and  $T_L$  (low threshold), it is set to zero unless there is a path from this pixel to a pixel with a gradient above the  $T_L$ .

The Sobel operator performs a two-dimensional spatial gradient measurement over the 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 [18]:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \quad (9)$$

*Step 4: Adjusting  $T_H$  and  $T_L$  for better results.*

Better results are achieved 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 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 the  $T_E$ , the automatic adjustment will be terminated. At this point, a preliminary image with edges identified is obtained from the original flame image. It is designated as a Preliminary Edge Image (PEI).

*Step 5: Removing unrelated edges in the PEI.*

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  $3 \times 3$  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 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 two or more than two neighboring pixels, 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 new search. Store the other positions as unchecked conjunction points, and then go back to Step 5b).

Fig.3 illustrates how the tracing step moves forward if the old search center is replaced by a new search center. For instance, in the left image of the figure, pixel ‘5’ is the center selected. Suppose an edge

point at pixel '9' is found, then remove pixel '5' from the PEI to the temporary edge image, and pixel '9' will be the new search center. In this way, the search moves forward pixel by pixel.

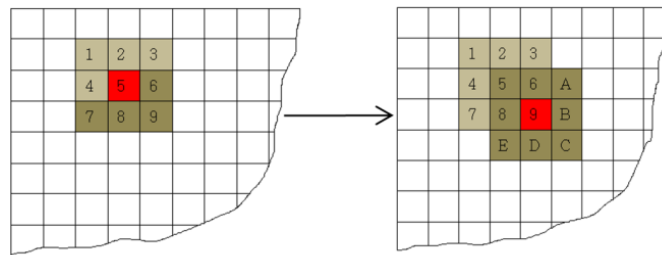


Fig. 3 Illustration of movement of the edge search.

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: Achieving a clearly defined edge.*

Select the pixels of the longest edge in the final edge image which should have the same size as the original image. The flow chart of the whole process is given in Fig.4.

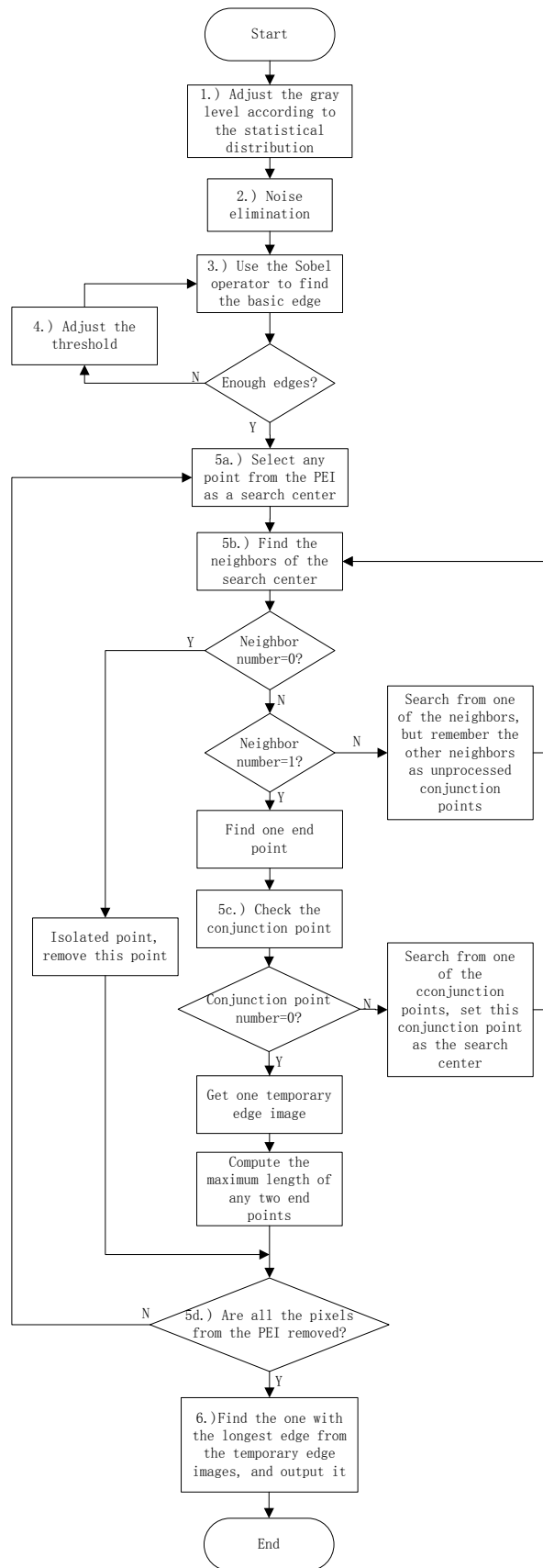


Fig. 4 Flow chart of the flame edge detection algorithm.

In order to speed up the process of finding proper  $T_H$  and  $T_L$ , a Least-Mean-Squares (LMS) algorithm is used, which is a class of adaptive filters by finding the filter coefficients that relate to producing the least mean squares of the error signal between the desired result and the actual result. The idea is to use steepest descent to find filter weights which minimize a cost function. From the method of the steepest descent, the weight vector equation is given by [24],

$$w(n + 1) = w(n) + \frac{1}{2} \mu [-\nabla(E\{e^2(n)\})] \quad , \quad (10)$$

where  $\mu$  is the step-size parameter which controls the convergence characteristics of the LMS algorithm;  $e^2(n)$  is the mean square error between the former output  $y(n)$  and the reference signal which is given by,

$$e^2(n) = [d^*(n) - w^h x(n)]^2 \quad (11)$$

The gradient vector in the above weight update equation can be computed as

$$\nabla_w(E\{e^2(n)\}) = -2r(n) + 2R(n) \quad (12)$$

Where  $r(n)$  and  $R(n)$  are covariance matrices which are defined as follows:

$$r(n) = x(n)d^*(n), \quad (13)$$

$$R(n) = x(n)x^h(n). \quad (14)$$

And then the weight update can be given by,

$$w(n + 1) = w(n) + \mu x(n)[d^*(n) - x^h(n) w(n)], \quad (15)$$

so,

$$w(n + 1) = w(n) + \mu x(n)e^*(n). \quad (16)$$

In this application the two parameters,  $T_H$  and  $T_L$ , need to be auto-adjusted. Give initial  $T_H$  and  $T_L$  according to the apriori results and set one of the parameters as fixed in order to adjust the other one. For example, suppose  $T_H$  is fixed, then  $T_L$  is adjusted in every step. The Euclidean distance of a curve's start point  $C_s$  and end point  $C_e$ , noted as  $D$ , is used as the output to judge the effectiveness of the new  $T_L$ . The coordinates of  $C_s$  ( $S_x, S_y$ ) and end point ( $E_x, E_y$ ) should be stored in the memory in the tracing process. Thus  $D$  can be computed as:

$$D = \sqrt{(E_x - S_x)^2 + (E_y - S_y)^2}. \quad (17)$$

After the LMS computation process, a suitable  $T_L$  is chosen. If  $D$  is small enough, the computation process terminates; if  $D$  is still greater than the desired value, a further LMS computation process is applied to  $T_H$ .

Fig. 5 shows some of the results with the proposed method given the fixed  $T_H$  and  $T_L$ . It can be seen that a pair of proper thresholds are necessary for the integrated flame edge. As explained in Step 4, the first pair of  $T_H$  and  $T_L$  values are selected according to the apriori results of similar flame images. The apriori  $T_H$  and  $T_L$  will work in most situations, but there are exceptions as the scenario may change, and the flame may change enough to make the previous threshold invalid. Fig. 5(f) shows a closed curve and the Euclidean distance between the start point and the end point is zero, which is regarded as the best result that in the checked  $T_H$  and  $T_L$  region. Obviously, if  $D=0$ , no other adjustment of  $T_H$  and  $T_L$  is necessary, thus the auto-adaptive process is over. If  $D$  is still big enough, as that in the situation of Fig. 5(c) and Fig. 5(e), a further adjustment has to be executed. The  $T_H$  and  $T_L$  should be adjusted in a greater scale. When  $D$  becomes very small, say, less than 20 pixels, the  $T_H$  and  $T_L$  should be adjusted through finer steps.

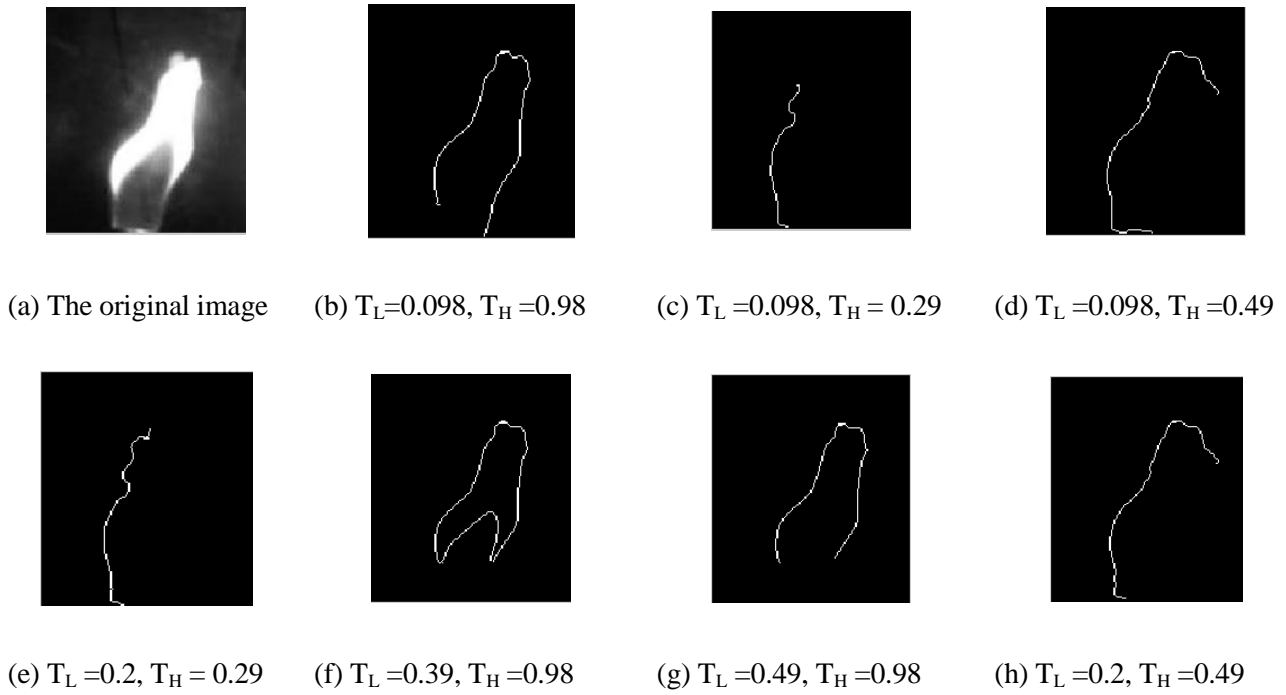


Fig. 5 Illustration of longest curve in a flame image with minimum D.

#### IV. RESULTS AND DISCUSSIONS

After implementing the algorithm as described in Section III, thousands of flame images were processed using the algorithm so as to evaluate its effectiveness. Most of the flame images were taken for propane Bunsen flames burning in open air. Some of the images were attained from the internet with courtesy of permission of use. The desktop computer used has a 2.66GHz Intel® Quad CPU and can detect the edges of about 120 flame images of  $141 \times 161$  pixels in one minute. Fig. 6 shows typical processed flame images with edges identified. In comparison with the test results presented in Fig. 1, it can be clearly observed that the developed algorithm can successfully detect clear edges of the flame and disregard unrelated artifacts, which common edge detection methods cannot achieve. The proposed method makes it much easier to distinguish the flame region from the background. The algorithm can also be used to extract the edges of more complex flames such as turbulent diffusion flames or flames of pool fires [14]. The clearly defined flame edges will form a basis for subsequent processing of the flame images for example, flame size computation, flame background removal, and determination of other flame parameters [3].

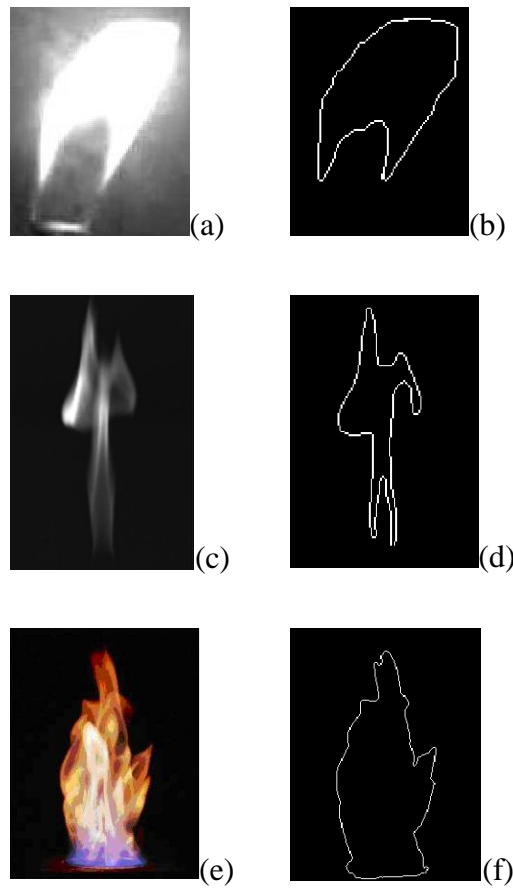
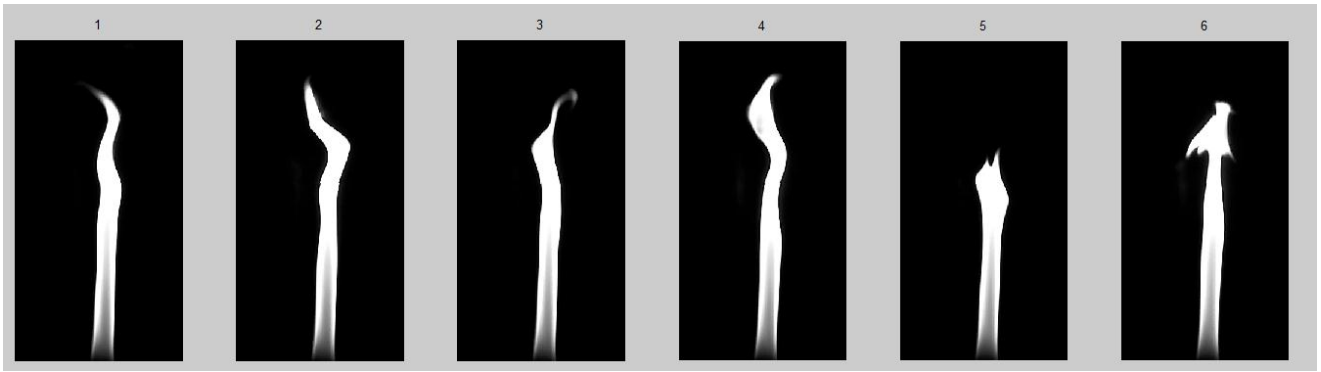


Fig. 6 Some of the flame edge detection results.

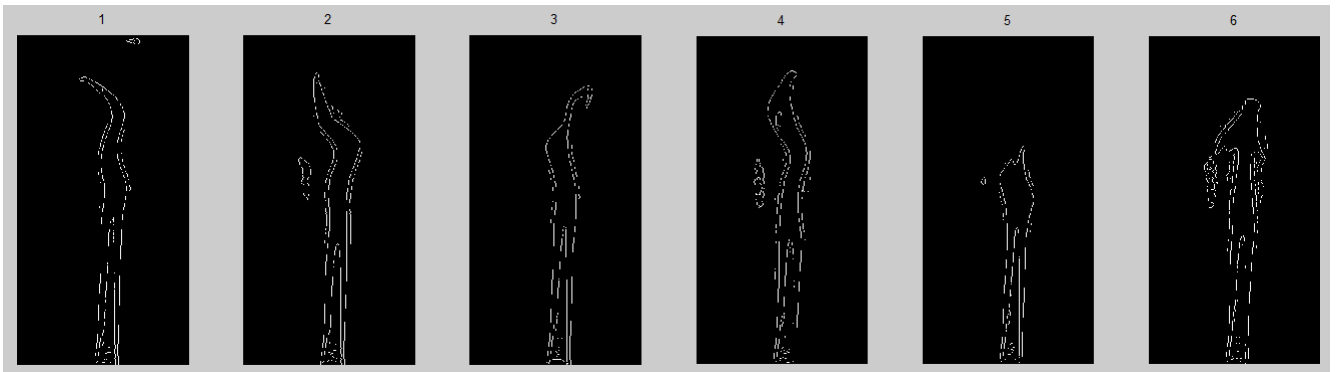
(Left column: original images; right column: images with identified edges. (a) Diffusion propane-flame, (c) Partially-premixed propane-flame, (e) Small scale pool fire[17]. Note that the image in (a) is reproduced from Fig. 1).

Many flame videos are also tested for continuous edge detection so as to evaluate the robustness of the system. Fig. 7(a) shows a series of frames acquired from a flame video. Fig. 7(b) and Fig. 7(c) illustrate the edge detection results using both the Canny edge detection method and the proposed algorithm. It is clear that the flame edges detected using the Canny edge detection method are unclear and discontinuous, whilst the results obtained using the proposed algorithm show clear and continuous edges with parameters automatically adapted.

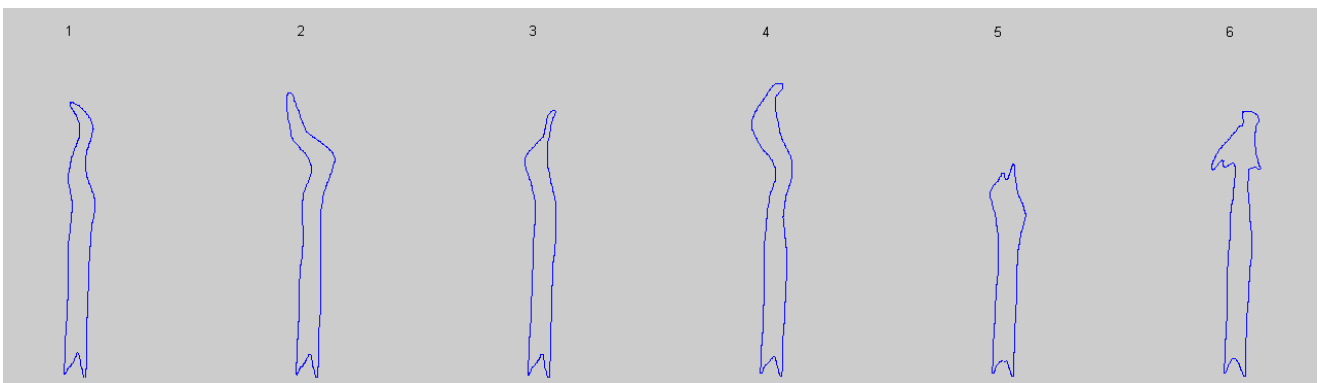




(a) Frames in a flame video



(b) Detected flame edge from the video sequence using the Canny edge detection method



(c) Detected flame edge from the video sequence using the proposed method

Fig. 7 Edge detection result for a flame video.

With a clearly defined flame/fire edge, various flame/fire parameters can be easily computed for the shape description. For instance, the flame area can be counted by the number of pixels inside the flame edge; the chain coding of a flame edge can be used to describe a 2D flame/fire shape; the perimeter of a flame can be achieved by the total number of pixels of the detected flame edge boundary. Fig. 8 is an

example of the uninterrupted computing of the flame perimeters from a flame video. It would be difficult to obtain this result without the clear edge detection. Using the proposed edge detection algorithm further work can be done to characterize the geometric features of flames/fires, and consequently establish their relationship with combustion conditions such as air/fuel inputs and emissions.

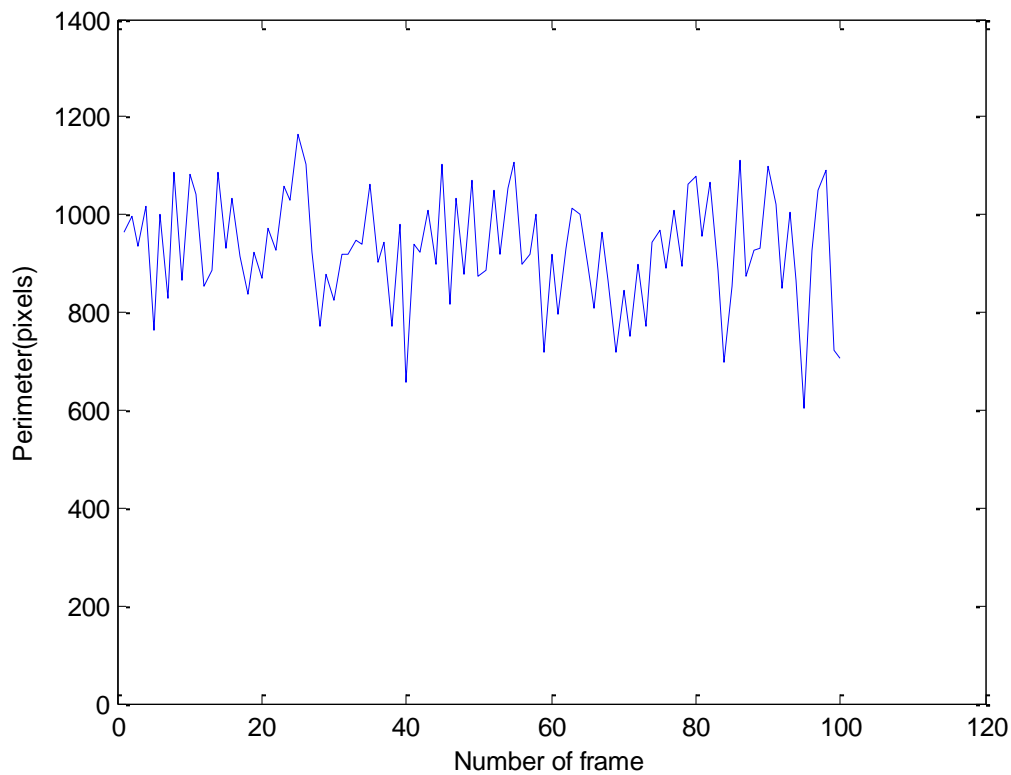
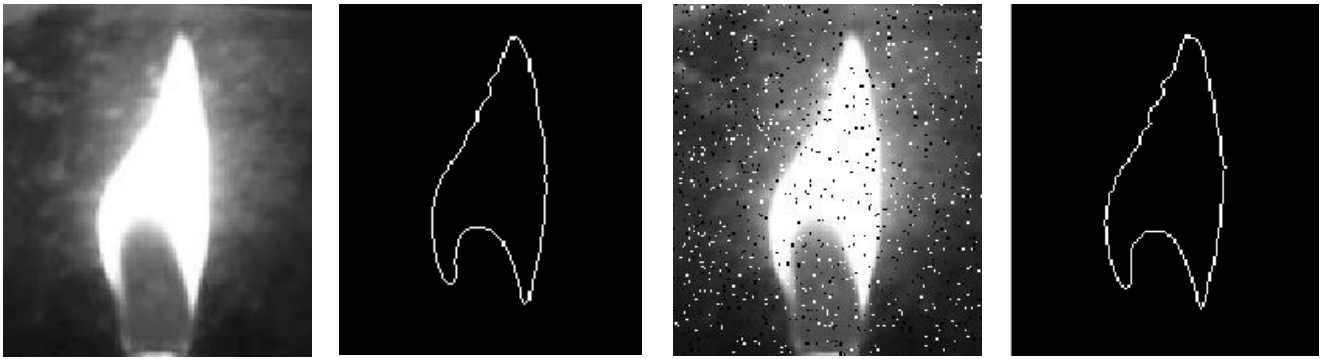


Fig. 8 The uninterrupted computing of the flame perimeters from a flame video.

To assess the anti-noise effect of the algorithm, different types of random noise are added to the flame images before the images are processed. Fig.9 shows an example of processing results with pepper and salt noise added. Fig. 9(b) is the flame edge detected from the original flame image Fig. 9(a) whilst Fig. 9(d) illustrates the flame edge detected from the flame image with pepper and salt noise added. It can be seen that the shape detected in Fig. 9(d) is almost identical to the shape detected in Fig. 9(b).



(a) Original image

(b) Processed image of (a)

(c) Original image with salt  
and pepper noise added

(d) Processed image of (c)

Fig. 9 An example of noisy and processed flame images.

## 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 irregular flames. The advantage of this method is that the flame and fire edges detected are clear and continuous. Furthermore, with the change of scenarios, the parameters in the algorithm can be automatically adjusted. The clearly defined combustion region lays a good foundation for subsequent quantification of flame parameters [27], such as flame volume, surface area, flame spread speed and so on. It is envisaged that this effective flame edge detection algorithm can contribute to the in-depth understanding and advanced monitoring of combustion flames. Meanwhile, the algorithm provides a useful addition to fire image processing and analysis in fire safety engineering. The work presented was aimed for the processing of flame and fire images captured in laboratories. Further work is required to evaluate the performance of the algorithm in real-life flame detection scenarios.

## ACKNOWLEDGMENT

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. Gilabert, "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] Z. Zhang, J. Zhao, D. Zhang, C. Qu, Y. Ke and B. Cai, "Contour Based Forest Fire Detection Using FFT and Wavelet," *International Conference on Computer Science and Software Engineering, (CSSE 2008)*, Wuhan, China, 12-14 December 2008, pp.760-763.
- [6] X. Zhou, F. Yu, Y. Wen, Z. Lu and G. Song, "Early Fire Detection Based on Flame Contours in Video," *Information Technology Journal*, ISSN 1812-5638, Volume: 9; Issue: 5; pp. 899- 908, 2010.
- [7] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Flame detection in video using hidden Markov models," *IEEE Int. Conf. On Image Proc., (ICIP 2005)*, 11-14 Sept. 2005, pp. II - 1230-3.

- [8] B. U. Toreyin, Yigithan Dedeoglu, Ugur. Gudukbay, and A. Enis Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, Volume 27, Issue 1, 1 January 2006, pp. 49-58.
- [9] Mario I. Chacon-Murguia and Francisco J. Perez-Vargas, "Thermal Video Analysis for Fire Detection Using Shape Regularity and Intensity Saturation Features," *Lecture Notes in Computer Science*, 2011, Volume 6718/2011, pp. 118-126.
- [10] S.M. Razmi, N. Saad, and V.S. Asirvadam, "Vision-based flame analysis using motion and Edge Detection," 2010 International Conference on Intelligent and Advanced Systems (ICIAS), Issue Date: 15-17 June 2010, On page(s): 1 – 4
- [11] B. U. Toreyin, and A. E. Cetin, "Online Detection of Fire in Video," *cvpr*, pp.1-5, 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] Q. Jiang, Q. Wang, "Large Space Fire Image Processing of Improving Canny Edge Detector Based on Adaptive Smoothing," *cicc-itoe*, pp.264-267, 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering, 2010.
- [16] C. W. Adkins, "Users guide for fire image analysis system-version 5.0: a tool for measuring fire behavior characteristics," *Gen. Tech. Rep. SE93*. Asheville, NC: U.S. Department of Agriculture, Forest Service, Southern Research Station, 15 p,1995.

- [17] X. She, and F. Huang, "Flame Edge Detection Based on C-V Active Contour Model," *Artificial Intelligence and Computational Intelligence, International Conference on*, vol. 2, pp. 413-417, 2009 *International Conference on Artificial Intelligence and Computational Intelligence*, 2009.
- [18] R. C. Gonzalez and R. E Woods, *Digital Image Processing, Second Edition*, Prentice Hall, 2002.
- [19] 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.
- [20] J. Canny, "A computational approach to edge detection", *IEEE Transactions on PAMI*, Vol. 8, No. 6, pp. 679–698, 1986.
- [21] K. R. Castleman, *Digital Image Processing*, Prentice Hall, 1995.
- [22] SWRI, "Small-scale liquid pool fire characterization", Available at: <http://www.swri.org/4org/d01/fire/firetech/about.htm>.
- [23] D. Ziou and S. Tabbone, "Edge detection techniques: An overview", *International Journal of Pattern Recognition and Image Analysis*, 8(4):537–559, 1998.
- [24] Simon Haykin: *Adaptive Filter Theory*, Prentice Hall, 2002, ISBN 0-13-048434-2.
- [25] T. Qiu, Y. Yan and G. Lu, "A new edge detection algorithm for flame image processing", *Proceedings of IEEE International Instrumentation and Measurement Conference (I2MTC 2011)*, Hangzhou, China, 10-12 May 2011, pp. 281-284.
- [26] W.S. Steven, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Pub, 2003.
- [27] T. Qiu, Y. Yan and G. Lu, "A Medial Axis Extraction Algorithm for Combustion Flames through Digital Image Processing" the 6th *International Conference on Image and Graphics (ICIG 2011)*, Hefei, China, 12-15 August 2011, pp.182 -186.