# DSP Embedded Early Fire Detection Method Using IR Thermal Video

**Won-Ho Kim**[1]
[1] Department of Electrical, Electronic and Control Engineering, Kongju National University
Cheonan, Republic of Korea
[e-mail: whkim@kongju.ac.kr]
*Corresponding author: Won-Ho Kim

*Received July 17, 2014; revised September 11, 2014; accepted October 14, 2014; published October 31, 2014*

## *Abstract*

Here we present a simple flame detection method for an infrared (IR) thermal camera based real-time fire surveillance digital signal processor (DSP) system. Infrared thermal cameras are especially advantageous for unattended fire surveillance. All-weather monitoring is possible, regardless of illumination and climate conditions, and the data quantity to be processed is one-third that of color videos. Conventional IR camera-based fire detection methods used mainly pixel-based temporal correlation functions. In the temporal correlation function-based methods, temporal changes in pixel intensity generated by the irregular motion and spreading of the flame pixels are measured using correlation functions. The correlation values of non-flame regions are uniform, but the flame regions have irregular temporal correlation values. To satisfy the requirement of early detection, all fire detection techniques should be practically applied within a very short period of time. The conventional pixel-based correlation function is computationally intensive. In this paper, we propose an IR camera-based simple flame detection algorithm optimized with a compact embedded DSP system to achieve early detection. To reduce the computational load, block-based calculations are used to select the candidate flame region and measure the temporal motion of flames. These functions are used together to obtain the early flame detection algorithm. The proposed simple algorithm was tested to verify the required function and performance in real-time using IR test videos and a real-time DSP system. The findings indicated that the system detected the flames within 5 to 20 seconds, and had a correct flame detection ratio of 100% with an acceptable false detection ratio in video sequence level.

***Keywords:*** video surveillance system, IR image processing, early fire detection, flame detection, digital signal processing

This research was supported by the International Science and Business Belt Program through the Ministry of Science, ICT and Future Planning (former Education, Science and Technology) (2013K000490).

**http://dx.doi.org/10.3837/tiis.2014.10.011**

## 1. Introduction

Intelligent fire detection systems based on computer vision processing were recently developed to prevent individual or nation-wide disasters due to urban fires or forest fires [1]-[12]. Two types of video camera systems are commonly used for video surveillance. One is a charge-coupled device color camera system, and the other is an infrared (IR) thermal camera system. Color camera systems are cost-effective, but their operations are limited under no-light and heavy fog conditions [1]-[6], [11]-[12].

In [1], the proposed algorithm comprises pixel-based fire motion and color analysis, with temporal analysis using a wavelet transform and spatial analysis using energy calculations of the fire area. In [2], the authors propose a pixel-based fire color modeling that was developed in CIE-Lab color space, and fire motion analysis is used to detect fire events. In [3], pixel-based fire motion is detected by background subtraction, edge detection, flame height, and width analysis, and color analysis methods are used for fire detection. In [4], pixel-based frame difference calculations are used to extract the moving fire area, and color analysis and optical flow computation of dynamic features extraction methods are applied for detecting fire. In [5], fire regions are detected using pixel-based potential fire region extraction processing with a spectral and spatial model, and temporal property calculations using Fourier coefficients are used to detect fire regions. In [6], a pixel-based modified histogram back-projection algorithm is used to detect flame regions (FRs). In [11], the authers present pixel-based color analysis using a Gaussian smoothed color histogram, fire motion analysis based on temporal variations in pixels, and erosion and region growing are performed to detect fire. In [12], the author proposes a forest-fire monitoring system based on field information, IR and visual image processing. This system is based on forest fire properties such as the fire front, flame height, flame inclination angle, fire base width, a three dimensional perception model, sensor fusion techniques involving telemetry sensors, and GPS. In [1]-[6] and [11]-[12], the authors propose a color video fire alarm system using pixel-based calculation algorithms implemented on a personal computer. These algorithms place a heavy computational load on personal computers due to the pixel-based calculations and complex schemes, and are therefore not suitable for early fire detection and fast processing using an embedded digital signal processor (DSP) system.

IR thermal camera systems are advantageous for unattended fire surveillance. Monitoring is possible regardless of illumination and climate conditions, and the quantity of data to be processed is one-third that of color videos. A significant drawback of IR thermal camera systems, however, is the cost of the IR thermal camera. Fortunately, the cost tends to decrease with the evolution of IR camera technology. Therefore, fire video surveillance systems using IR thermal cameras are being developed to improve fire detection reliability and vision processing time [7]-[10]. In [7], the different behaviors of medium (3-5 μm) and thermal (8-12 μm) IR spectral regions are used to calculate the Fire Index (FI) and mid-IR Fire Index (MFI) using two IR cameras working in the medium and thermal IR spectral windows, which are used to identify fires. In [8], several software components based on threshold, oscillation detection, matching of visual and IR images, memory, meteorologic information, motion, size, shape, solar conditions, and location are used to implement personal computer-based fire alarm system. The proposed forest fire detection system in [9] combines computer vision tools, neural networks, and expert fuzzy rules to detect forest fires in open areas, whereas [10] proposes a forest-fire detection system based on images obtained from IR cameras. The

algorithm utilize data obtained from different detectors that exploit different expected characteristics of a fire, such as persistence and growh. These IR image-processing methods comprise pixel-based calculations and complex schemes and are therefore not suitable for an embedded DSP system. The main type of IR vision-processing method is temporal correlation-based fire detection [8]-[9]. In temporal correlation methods, temporal changes in the brightness generated by the irregular motion and spreading of the flame pixels are measured. These brightness oscillations can be observed as changes in the shape and brightness of the FR in IR thermal images. These characteristics of FRs are measured using a correlation function. Uniform correlation values indicate no brightness fluctuations between FRs. Irregular correlation values indicate brightness fluctuations between FRs.

For practical application, all fire detection techniques should be implemented within a very short period of time to achieve early detection. In early monitoring systems, the conventional pixel-based correlation function is computationally intensive. Conventional fire detection systems [7]-[10] are implemented using either a personal computer or server computer due to complex algorithms and heavy computational calculations. A compact embedded DSP hardware-based fire detection system is necessary to achieve a device with low cost, high hardware endurance, and easy maintenance requirements. In this paper, we propose an IR video processing-based early flame detection algorithm optimized with a compact embedded DSP system. To reduce the computational load, pixel-based calculations were replaced with block-based calculations to select the candidate FRs and to evaluate the temporal motion of the flames. These functions were combined to obtain an early flame detection algorithm. The proposed method is described in Section II. In Section III, flame detection functions and performance were verified using a real-time DSP board. In the final section, we present our conclusion.

## 2. Block-based DSP Embedded Flame Detection Method

In IR thermal images, flame pixels have high intensity values compared with background values, as shown in **Fig. 1-(a)** and **Fig. 2-(a)**, and non-flame pixels generating false alarms have high intensity values compared with background pixels, as shown in **Fig. 1-(b)** and **Fig. 2-(b)**. A near-distance fire image is shown in **Fig. 1** and far-distance fire image is shown in **Fig. 2**. The source types of false detection are an artificial heated object, artificial lights and solar reflections as shown in **Fig. 1-(b)** and **Fig. 2(b)**. These may be falsely recognized as the FRs when using simple intensity threshold detection due to high-intensity pixels being regarded as flames.In the non-flame pixels, however, the temporal variations of intensity are static. In contrast, the pixel values in FRs fluctuate due to the temporally irregular motion of the flames.

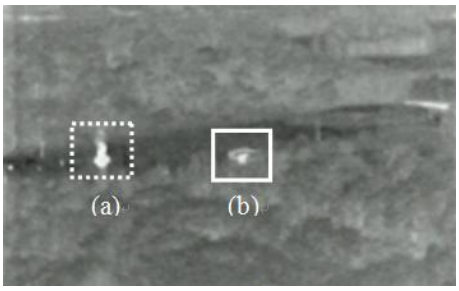



**Fig. 1.** Flame region (a) and non-flame region (b) in a near-distance IR thermal image

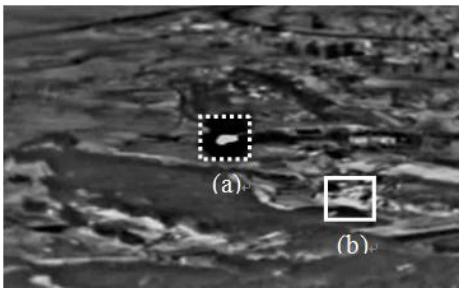



**Fig. 2.** Flame region (a) and non-flame region (b) in a far-distance IR thermal image

Block-based temporal intensity variations of the FR in **Fig. 1-(a)** are shown in **Fig. 3**, and block-based temporal intensity variations of the non-FR in **Fig. 1-(b)** are shown in **Fig. 4**.

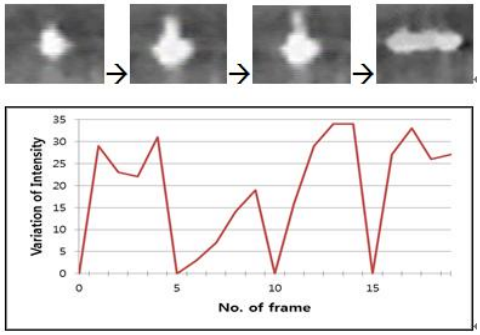



**Fig. 3.** Block-based temporal intensity variations of the flame region (field fire)

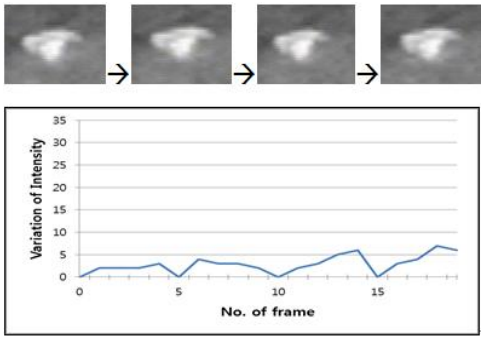



**Fig. 4.** Block-based temporal intensity variations of the non-flame region (heated rock)

Based on the above graphs, dynamic or static intensity variation characteristics of high-intensity regions are suitable for distinguishing flame occurrence. To achieve fast and simple flame detection and moderate false detection, block-based intensity threshold and region-based temporal similarity calculations can be integrated in an efficient manner. The proposed simple flame detection algorithm comprises a two-stage operation. First, block-based FR segmentation was performed using a fixed intensity threshold. Second, block-based temporal similarities between successive FRs are calculated to decide whether flames are present.

### 2.1 Block Based FR Segmentation

A two-step process is used to segment FRs in IR images. First, a flame block (FB) is determined according to the following procedures. The reference image is divided into blocks in MxN pixels, and the mean block intensity (MBI) is calculated using equation (1). The MBI values are filtered using the first threshold to determine candidate FRs with high intensity.

$$MBI_{ij} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} BLOCK_{ij}(m,n) \tag{1}$$

Here, *m* and *n* are the coordinates of the pixel in the block. *i*,*j* are horizontal and vertical indices of the block, and M and N are horizontal and vertical sizes of the block, respectively. The pixel intensity is digitized with 8-bit resolution, resulting in a maximum value of 255.

Next, the FR is segmented using the determined FBs. Temporally, the shape and intensity of FR changes dynamically and enlarges, as shown in **Fig. 3**. The motion directions of the flames can be usually predicted as moving above, left, or right. This phenomenon is applied to segment the FR based on a FB. **Fig. 5** shows the FR, and it comprises one FB and the eight blocks that are above, left, and right of the FB. The segmented FRs are processed in the next stage for the final detection of flame existence. These two steps are an efficient approach for reducing computation complexity.

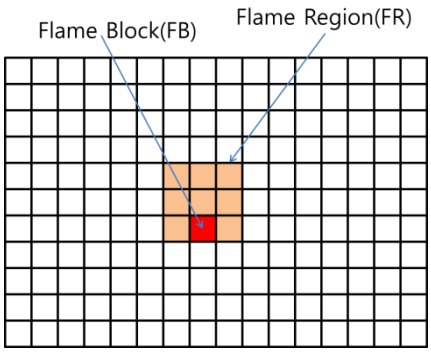



**Fig. 5.** Flame block-based flame region segmentation

### 2.2 Calculation of Block-based Temporal FR Similarity

After segmenting the FR, the block-based temporal similarity of FR is calculated to decide the final flame existence in the image. In general, correlation functions are well known and extensively used to detect similarities between two signals. Three methods are commonly used to calculate block-based similarities between image regions, as follows. The three methods are the mean of absolute difference (MAD), correlation (COR), and mean squared difference (MSD), as shown in equations (2), (3), and (4).

$$MAD_{ij} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left| BLOCK_{t-1}(m,n) - BLOCK_{t}(m,n)\right| \tag{2}$$

$$COR_{ij} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} BLOCK_{t-1}(m,n) \times BLOCK_{t}(m,n) \tag{3}$$

$$MSD_{ij} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[BLOCK_{t-1}(m,n) - BLOCK_{t}(m,n)\right] \tag{4}$$

In a general-purpose micro-processor, the computation load is significantly affected by the number of multiplication operations. Multiplication operations require more clock cycles than addition. **Table 1** shows the computational load of the three above methods in the case of an 8x8 block image. In general, the number of operation cycles required for multiplication is four times those required for addition, subtraction, or absolute instructions [11]. The computational load of MAD was the lowest among the three similarity calculation methods.

**Table 1.** Comparison of the computational load of a general-purpose micro-processor for an 8x8 block

| Method | ADD/SUB | MUL | ABS | Total cycles |
|---|---|---|---|---|
| MAD | 128 | 1 | 64 | 196 |
| COR | 64 | 64 | 0 | 320 |
| MSD | 128 | 65 | 0 | 388 |

Therefore, the MAD calculation can be suitable for real-time DSP operations and is efficient for early alarm generation. To evaluate the dynamic temporal motion of FR, Region Absolute Difference (RAD) is defined as shown in equation (5). A small RAD value indicates little change between two regions. A region with a small RAD value is classified as a non-FR. Regions having large RAD values are classified as FR.

$$RAD_{ij} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left| REGION_{t-1}(m,n) - REGION_{t}(m,n)\right| \tag{5}$$

In the next step, RADs are averaged for a number of successive frames to reduce erroneous flame detection. In every K-frame, the mean RAD (MRAD) is calculated using equation (6) and the result is filtered by a second threshold. If the MRAD value exceeds the threshold, a flame alarm is generated.

$$MRAD_{ij} = \frac{1}{K}\sum_{k=1}^{K} RAD_{ij}(n) \tag{6}$$

The above-described functions form the core of an early flame detection algorithm and **Fig. 6** shows the flow chart of the algorithm. Whole flame detection operations of the algorithm are as follows. First, a reference image is captured and divided into MxN blocks. MBI values are calculated for every block, and FBs are selected based on the first threshold. FRs of the reference image are segmented based on FBs, as shown in **Fig. 5**. The segmentation of FRs is processed for every reference image. After reference image processing, the next image frame is captured, and RADs of selected FRs are calculated between successive frames. After every K-frame, MRAD is calculated and filtered by the second threshold. Whenever the MRAD value exceeds the second threshold, a flame alarm is generated.

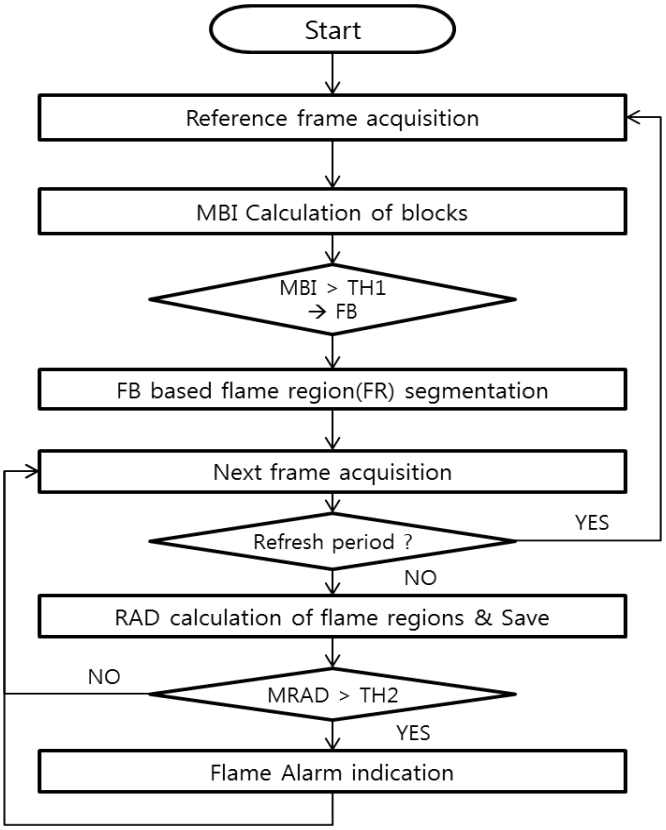



**Fig. 6.** Flow chart of a simple block-based flame detection algorithm for a real-time DSP system

## 3. Experimental Results

To verify real-time operations of the proposed block-based simple flame detection algorithm, 10 IR thermal videos and a real-time DSP board were used to test the algorithm, as shown in **Fig. 7**. Test videos were recorded with 30-s duration at 30 frames per second. The flame alarms were generated correctly in seven videos including fire scenes. False alarms were generated in 3 of 10 test videos. The test results are summarized in **Table 2**, and the test parameters for the proposed algorithm test were as follows: video size 720x480, interval time of frame acquisition 0.5 s, MBI threshold value 197, MRAD threshold value 5, frame number (K) for MRAD calculation 5, and block size 4x4 pixels. Two threshold values were selected experimentally. The frame acquisition interval time was selected to achieve optimal performance. The processing capability of the DSP system was evaluated at more than 20 frames per second (1000 ms/44 ms=22.7 for test video-1), as shown in **Fig. 7-(c)**. To illustrate the operation of the implemented DSP software, the pseudo codes of the DSP are shown in below.

```
//define main parameters
#define MBS          M      // size of macro block [MBS* MBS]
#define REF_PRD      P      // reference period
#define MBI_TH       TH1    // MBI threshold
#define MRAD_TH      TH2    // MRAD threshold
```

```
#define T                INTT     // frame processing interval, msec

video capture configuration;
video display configuration;

void tskVideoprocessing()
{
   int count_acquisition=0;
   allocate both capture and display frame buffers in external heap memory;
   initialization of capture driver;
   initialization of display driver;
   configure video encoder & decoder;
   request a frame buffer from display & capture driver;
   buffer initialization;

        while(1)
        {
                if(count_acquisition == 0) //reference video processing
                {
                        MBI computation;
                        flame block and flame region selection for reference image;
                count_ acquisition++;
                }
                else //input video processing
                        {
                        RAD calculation and memory;
                        if (count_ acquisition == REF_PRD )
                                {
                         MRAD calculation;
                            final flame decision using MRAD;
                         output alarm signal;
                                }
                        count_ acquisition++;
                        }
                if (count_acquisition == REF_PRD+1)  count_ acquisition = 0;
                wait_microsec(T*1000); // 500 msec delay function
                capture a new frame;
              display the processed frame;
        }
}
```

(a)

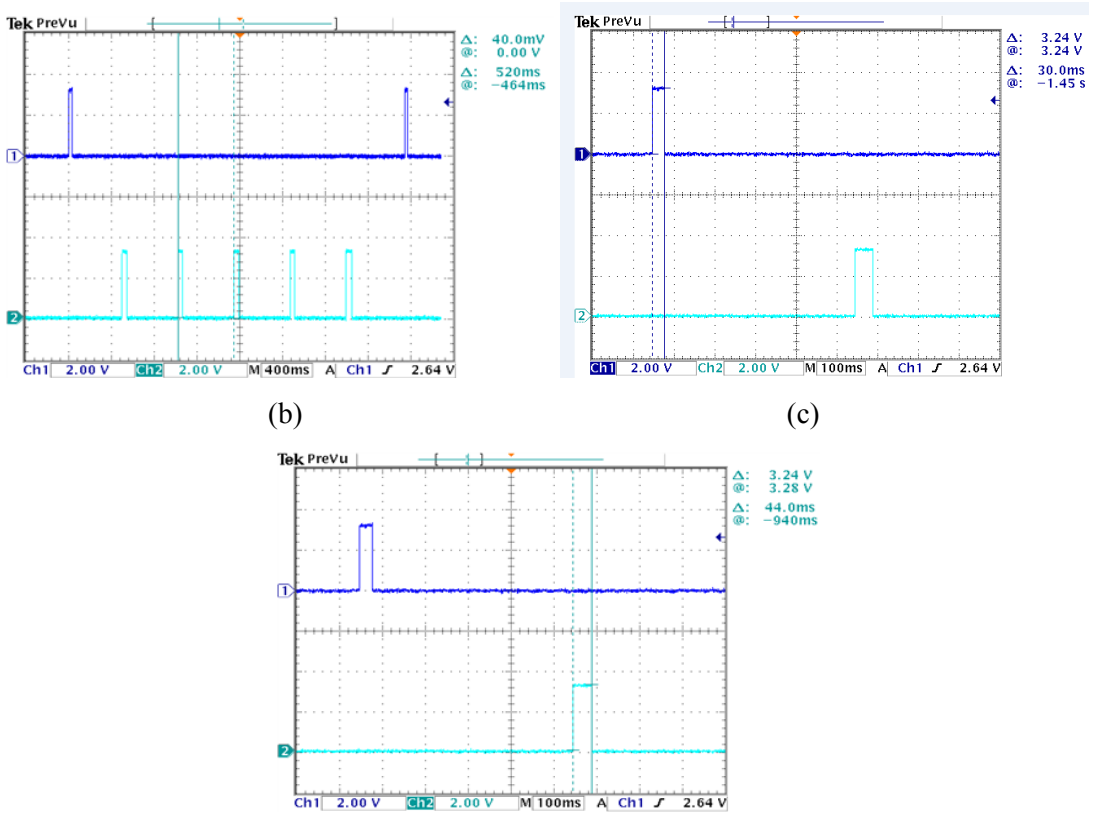



**Fig. 7.** Implemented real-time DSP system and mersured timing results for test video-1, (a) picture of test bed (b) oscilloscope picture of DSP operation timing, upper signal is reference image processing timing, bottom signal is input image processing timing (c) oscilloscope picture of reference image processing time (Δ=30 ms) (d) oscilloscope picture of input image processing time (Δ=44 ms)

**Table 2**. The summarized test results for block-based processing algorithm with video sequence level, frame number (K) for MRAD calculation 5, frame acquisition interval 0.5 s, decision interval 3 s

| No. | Correct Detection | False Detection | Detection Time[sec] | Notes |
|---|---|---|---|---|
| Video 1* | Yes | No | 5 | Near field fire |
| Video 2* | Yes | Yes | 13 | Far field fire |
| Video 3* | Yes | Yes | 8 | Far field fire |
| Video 4 | No | No | - | Sun reflection |
| Video 5 | No | No | - | Lamps of running cars |
| Video 6* | Yes | No | 7 | Torch fire in tunnel |
| Video 7* | Yes | No | 20 | Small fire |
| Video 8* | Yes | No | 14 | Far field fire |
| Video 9 | No | Yes | - | Moving excavators |
| Video10* | Yes | No | 6 | Near house fire |

∗Video contains fire scenes

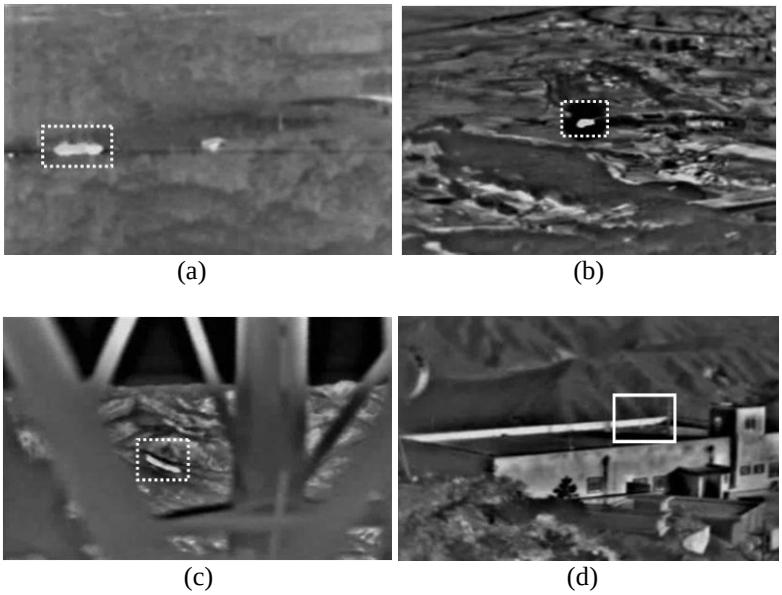
**Table 3** shows detailed test results for frame level, under the same conditions. The (K+1) refresh period is one reference frame plus K input frames duration.

**Table 3**. The summarized test results for block-based processing algorithm with video frame level, frame number (K) for MRAD calculation 5, frame acquisition interval 0.5 s, flame decision interval 3 s

| Test Video No. | Total No. of frames (30s duration) | No. of processed frames (0.5s interval) | No. of flame decision (every 3s) | No. of correct decision (every 3s) | No. of false decision (every 3s) |
|---|---|---|---|---|---|
| Video 1* | 900 | 60 | 10 | 10 | 0 |
| Video 2* | 900 | 60 | 10 | 10 | 9 |
| Video 3* | 900 | 60 | 10 | 7 | 2 |
| Video 4 | 900 | 60 | 10 | 0 | 0 |
| Video 5 | 900 | 60 | 10 | 0 | 0 |
| Video 6* | 900 | 60 | 10 | 10 | 0 |
| Video 7* | 900 | 60 | 10 | 6 | 0 |
| Video 8* | 900 | 60 | 10 | 5 | 0 |
| Video 9 | 900 | 60 | 10 | 0 | 8 |
| Video10* | 900 | 60 | 10 | 10 | 0 |

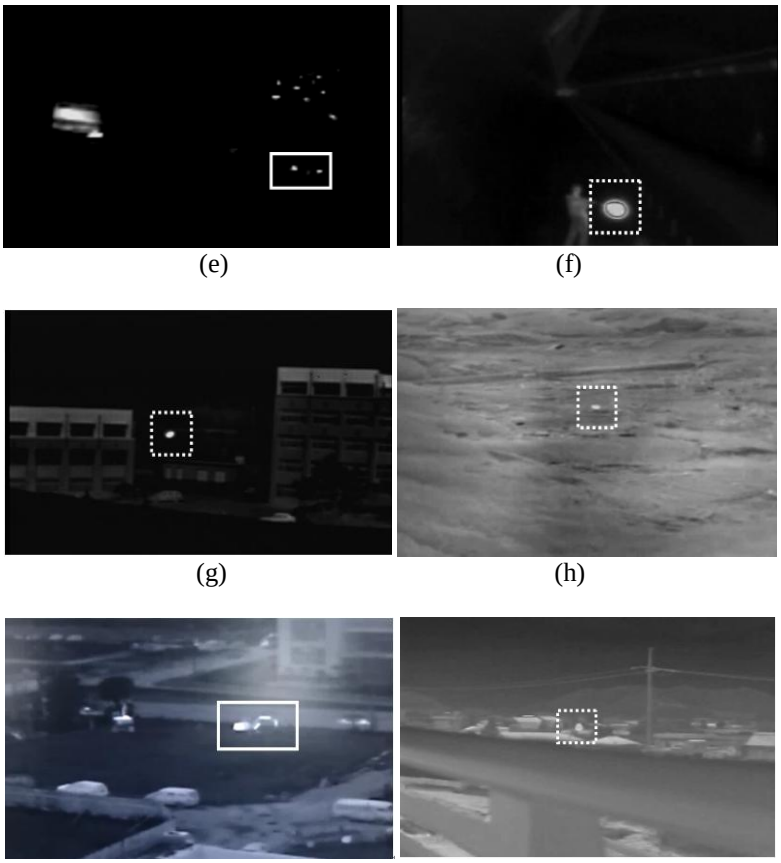
*Video contains fire scenes

The characteristics of the test videos used for the testing were as follows. Seven fire videos (video-1,2,3,6,7,8, and 10) contained fire scenes, and three non-fire videos (video-4,5, and 9) did not contain fire scenes. The seven fire videos included real FRs (dashed-line boxes in **Fig. 8-a,b,c,f,g,h,j**). The three non-fire videos included false alarm sources with high intensity (solid-line boxes in **Fig. 8-d,e,i**). The IR thermal camera used to record the test videos was the micro-bolometer type, which can sense a temperature difference of 0.07 ℃.



(a) (b)

(c) (d)



(e) (f)

(g) (h)

(i) (j)

**Fig. 8.** Experimental results. **(a)** video-1, near field fire, **(b)** video-2, far field fire, **(c)** video-3, far mountain fire, **(d)** video-4, sun reflection of roof**, (e)** video-5, lamps of running cars in tunnel**,(f)** video-6, torch fire in tunnel**,(g)** video-7, small building fire, **(h)** video-8, far field fire, **(i)** video-9, moving excavators, **(j)** video-10, near house fire

The seven fire videos (**Fig. 8-a,b,c,f,g,h,j**) generated correct flame alarms. These videos show the high intensity values and dynamic motion characteristics of FRs. Thus, these flames were correctly detected by the proposed simple algorithm. Video-2 and video-3 (**Fig. 8-b,c**) generated both of a correct flame alarm and a false alarm. They have far-distance FRs and false alarm sources with high-intensity regions due to sun reflection. In these cases, the apparent motion of the sun reflection regions was created by strong winds moving the camera during the recording. The motion characteristics of sun reflection regions are usually stationary, but strong winds moved the surveillance camera, thus creating the apparent motion of the sun reflection regions, which was detected as motion of the FRs. Therefore, the two conditions of high intensity and dynamic motion were both satisfied. Test video-9 (**Fig. 8-i**) contains no FRs, but a false alarm was generated. The reason for the false alarm was the hot diesel engine and slow working arm of the excavators, which resulted in the two conditions of high intensity and dynamic motion being simultaneously satisfied. To reduce these kinds of false alarms, movements of the surveillance camera, and flame-like motion of a fixed object must be carefully considered.

The detection time of the proposed algorithm ranged between 5 and 20 s, as measured from flame ignition to decision. Therefore, the required early detection function for a real-time DSP system was clearly met. Consequently, the detection time was less than 20 s, and flame detection ratio of 100% is confirmed in video sequence level. The flame detection ratio is calculated in a video sequence level by the equation (7).

$$\text{FLAME DETECTION RATIO}(\%) = \frac{\text{No. of video sequences with correct flame detection}}{\text{No. of video sequences including flames}} \times 100 \quad (7)$$

For comparison of correct-detection performance and real-time processing capability, pixel-based flame detection algorithm based on the proposed algorithm was tested using same 10 IR thermal videos and a real-time DSP board. The flame alarms were generated correctly in seven videos including fire scenes. False alarms were generated in 3 of 10 test videos. Two threshold values were selected experimentally. The test results are summarized in **Table 4,** and the test parameters for pixel-based algorithm test were as follows: video size 720x480, interval time of frame acquisition 0.5 s, MBI threshold value 197, MRAD threshold value 10, frame number (K) for MRAD calculation 5, and block size 1x1.

**Table 4**. The test results for pixel-based flame detection algorithm with video frame level, frame number (K) for MRAD calculation 5, frame acquisition interval 0.5 s, flame decision interval 3 s

| Test Video No. | Total No. of frames (30s duration) | No. of processed frames (0.5s interval) | No. of flame decision (every 3s) | No. of correct decision (every 3s) | No. of false decision (every 3s) |
|---|---|---|---|---|---|
| Video 1* | 900 | 60 | 10 | 9 | 0 |
| Video 2* | 900 | 60 | 10 | 8 | 7 |
| Video 3* | 900 | 60 | 10 | 4 | 0 |
| Video 4 | 900 | 60 | 10 | 0 | 0 |
| Video 5 | 900 | 60 | 10 | 0 | 2 |
| Video 6* | 900 | 60 | 10 | 9 | 0 |
| Video 7* | 900 | 60 | 10 | 7 | 0 |
| Video 8* | 900 | 60 | 10 | 1 | 0 |
| Video 9 | 900 | 60 | 10 | 0 | 8 |
| Video10* | 900 | 60 | 10 | 9 | 0 |

*Video contains fire scenes

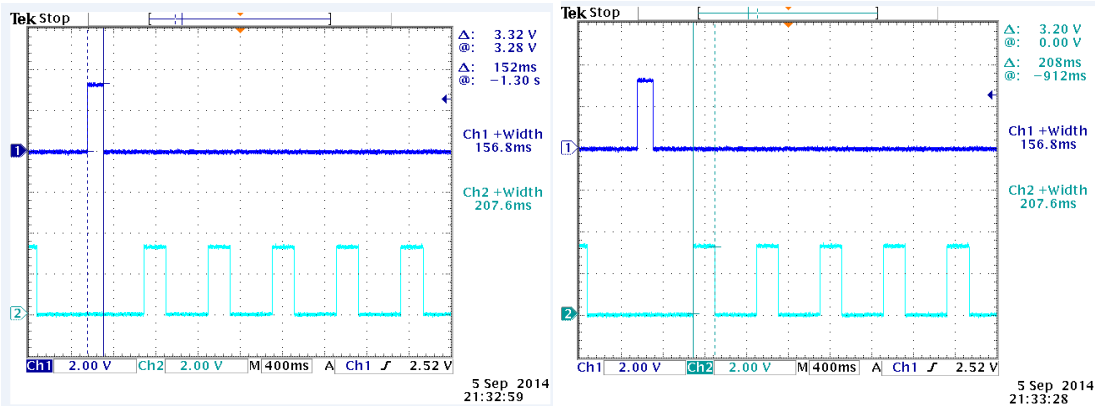
The comparison results of flame detection performance are summarized in **Table 5.** Compared with pixel-based algorithm, the proposed block-based algorithm showed better correct-detection performance.

**Table 5**. The comparison results of flame detection performance, video frame level, frame number (K) for MRAD calculation 5, frame acquisition interval 0.5 s, flame decision interval 3 s

| Test Video No. | No. of flame decision (every 3s) | Proposed block-based processing | | Pixel-based processing | |
|---|---|---|---|---|---|
| | | No. of correct decision | No. of false decision | No. of correct decision | No. of false decision |
| Video 1* | 10 | 10 | 0 | 9 | 0 |
| Video 2* | 10 | 10 | 9 | 8 | 7 |
| Video 3* | 10 | 7 | 2 | 4 | 0 |
| Video 4 | 10 | 0 | 0 | 0 | 0 |
| Video 5 | 10 | 0 | 0 | 0 | 2 |
| Video 6* | 10 | 10 | 0 | 9 | 0 |
| Video 7* | 10 | 6 | 0 | 7 | 0 |
| Video 8* | 10 | 5 | 0 | 1 | 0 |
| Video 9 | 10 | 0 | 8 | 0 | 8 |
| Video10* | 10 | 10 | 0 | 9 | 0 |

*Video contains fire scenes

Next, the real-time processing capability of the DSP system based on pixel-based image processing was evaluated, as shown in **Fig. 9**. The reference image processing time was 152 ms (**Fig. 9-(a)**) and input image processing time was 208 ms (**Fig. 9-(b)**) for pixel-based flame detection algorithm.




(a) (b)

**Fig. 9.** Mersured real-time processing capability of pixel-based algorithm for test video-1, (a) oscilloscope picture of DSP operation timing, upper signal is reference image processing time (Δ=152 ms) (b) oscilloscope picture of DSP operation timing, bottom signal is input image processing time (Δ=208 ms)

The comparison results of real-time processing capability are listed in **Table 6.** Compared with pixel-based algorithm, the proposed block-based algorithm showed excellent real-time processing capability to suitable implementation of embedded DSP system.

**Table 6**. The comparison results of real-time processing capability

| **Proposed block-based processing** | | **Pixel-based processing** | |
|---|---|---|---|
| **Reference image processing time** | **Input image processing time (equivalent frame processing rate)** | **Reference image processing time** | **Input image processing time (equivalent frame processing rate)** |
| 30 ms | 44 ms (22 fps) | 152 ms | 208 ms (4 fps) |

## 4. Conclusion

In this paper, we propose a block-based simple flame detection algorithm optimized with an embedded device for IR camera-based real-time fire surveillance DSP systems. To satisfy early detection requirements, block-based threshold and MAD functions were integrated in an efficient manner. To realize this efficient system, the block-based MBI was filtered to select FB, and the selected FB-based FR segmentation was performed. In the last step, the temporal RAD of the FR was checked to identify the occurrence of a flame. The real-time functions and performances of the proposed algorithm were evaluated through testing using the real-time DSP system. The flames of test videos were detected within 20 s, and the practicality was confirmed by a flame alarm ratio of 100% and an acceptable false alarm ratio in video sequence level. The proposed algorithm is thus comparatively effective for real-time fire detection, and will be a suitable embedded algorithm for a real-time video surveillance DSP system. Future studies to reduce false alarms related to movements of the surveillance camera and flame-like motions of fixed high-temperature moving objects will be performed to develop an even more reliable fire alerting system.

## References

[1] TJokorda Agung Budi W., Iping Supriana Suwardi, "Fire alarm system based on video processing," in *Proc. of International Conference on Electrical Engineering Informatics*, pp. 1-7, 2011. Article (CrossRef Link).

[2] Turgay celik, "Fast and efficient method for fire detection using image processing," *ETRI journal*, volume 32, number 6, pp. 881-890, 2010. Article (CrossRef Link).

[3] Ramzmi S.M., Saaad N., Asirvadam V.S., "Vision-based flame detection: motion detection & fire analysis," in *Proc. of IEEE Student Conference on Research and Development*, pp. 187-191, 2010. Article (CrossRef Link).

[4] Mei Zhibin, Yu Chunyu, Zhang Xi, "Machine vision based flame detection using multi-features," in *Proc. of 24th Chinese Control and Decision Conference,* pp. 2844-2848, 2012. Article (CrossRef Link).

[5] Zhengwen Xie, Qiang Wang, "Large space fire detection in laboratory-scale based on color image segmentation," in *Proc. of ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science*, pp. 572-575, 2010. Article (CrossRef Link).

[6] Wirth M., Zaremba R, "Flame region detection based on histogram back-projection," in *Proc. of*

*Canadian Conference on Computer and Robot Vision*, pp. 167-174, 2010. Article (CrossRef Link).
[7] S.Briz, A.J.de Castro, J.M. Aranda, J. Melendz, F. Lopez, "Reduction of false alarm rate in automatic forest flame infrared surveillance systems," *Remote Sensing of Environment*, Elsevier Science Inc., pp. 19-26, 2003. Article (CrossRef Link).
[8] A.Ollero, B.C.Arrue, J.R.Martinez, J.J. Murillo, "Techniques for reducing false alarms in infrared forest-flame automatic detection systems," *Control Engineering Practice,* Vol.7, pp. 123-131, 1999. Article (CrossRef Link).
[9] B.C. Arrue, A. Ollero, J.M de Dios, "An intelligent system for false alarm reduction in infrared forest flame detection," *IEEE intelligent system*, pp. 64-73, 2000. Article (CrossRef Link).
[10] Ignacio Bosch et al, "Infrared image processing and its application to forest flame surveillance," in *Proc. of IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 283-288, 2007. Article (CrossRef Link).
[11] W. Philips, M.shah, N.V. Lobo, "Flame recognition in video," in *Proc. of the 5th IEEE workshop on application of computer vision,* pp. 224-229, 2000. Article (CrossRef Link).
[12] J.R. Martinez-de Dios et al, "Computer vision techniques for forest flame perception," *Image and Vision Computing*, Vol.26, pp. 550-562, 2008. Article (CrossRef Link).
[13] Rafael C. Gonzalez. *Digital image processing using MATLAB*, Prentice Hall, 2004.
[14] Kenneth ayala. *Intel microcontroller*. Thomson Delmar learning, 2005.

**Won-Ho Kim** received the B.S. and M.S. degrees in electronics engineering from Kyung-Pook National University, Daegu, Korea, in 1985, 1987, respectively. He received Ph.D. degree in electronics engineering from Chung-Nam National University, Daejeon, Korea, in 1999. From 1989 to 1999, he was with Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, as a senior researcher. In 1999, he joined Kongju National University where he is now a full professor. His research interests are areas of image analysis, video surveillance system, and image and communication signal processing.