# An Integrated Framework for 24-hours Fire Detection

Jongwon Choi and Jin Young Choi(✉)

ASRI, Department of Electrical and Computer Engineering,
Seoul National University, Seoul, South Korea
jwchoi.pil@gmail.com, jychoi@snu.ac.kr

**Abstract.** In this paper, the integrated framework for 24-hours fire detection with a camera is proposed. The framework consists of four novel modules: an integration module, a flame detector with a visible-light camera, a flame detector with an infrared-ray camera, and a smoke detector. According to the state decided by the integration module, different detectors are selected to find fires. The flame detector with a visible-light camera determines flame patches from candidates through the cascaded classifiers, based on the color, shape, and randomness of flames. The flame detector with an infrared-ray camera finds flames, using the random movement of blob candidates. The smoke detector recognizes the smoke regions by utilizing the colors and the transparent property of smoke. The three detectors and the integrated framework are tested with numerous videos, which validates the generality and the robustness of the proposed framework.

**Keywords:** Integrated framework · Fire detection · Flame detection · Smoke detection · 24 hours

## 1 Introduction

According to the reports from National Fire Protection Association [1], 15 % of home fire victims have been caused by the physical disability, which ranked second among the fatal factors of home fires. Most of the victims could not avoid the death because of delayed escapes from a fire. However, because conventional fire warning systems, such as a water sprinkler and a manual warning lever, are operated only by strong fires, it becomes too late for the physically disabled people to escape from the fire.

Therefore, the fire warning system for all day is essential for the physically disabled people. Among the various fire warning systems, the systems based on a vision sensor have been spotlighted due to its low price and easy installation. Therefore, there has been various research for the early fire detection with cameras.

The most general fire detection algorithm is a flame detection based on a visible light (VL) camera, which can be categorized into three types: pixel-level,


G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part II, LNCS 9914, pp. 463–479, 2016.
DOI: 10.1007/978-3-319-48881-3_32

**(a)** Flame and smoke by visible light cameras **(b)** Flame by infrared-ray cameras

**Fig. 1. Various Images of Fire.** The shapes of fire are various according to scenes and material. (a) shows the fire images captured by visible light cameras. (b) shows the fire images captured by infrared ray cameras.

blob-level, and patch-level algorithms. The pixel-level algorithms find flames by utilizing pixel-wise features including colors and flickers [2,3]. The pixel-level algorithms work very fast, but they show low performance because the shape of flame cannot be considered and the classifiers with the simple pixel-wise features can be easily biased by training data. The blob-level algorithms detect flames by extracting features from blob-level candidates [4,5,7]. The blob-level algorithms show better performance than the pixel-level algorithms, but their classifiers are hard to be trained due to the various shapes of flame blobs as shown in Fig. 1(a). For complementing the limitations of the pixel-level and blob-level algorithms, the patch-level algorithm is developed recently [6]. The patch-level algorithm showed good performance by considering local appearance, but the algorithm gave too many outliers yet to be applied in real scenes.

Furthermore, there have been flame detection algorithms based on an infrared-ray (IR) camera to detect fires at night [8–11]. The algorithms utilized the properties of flame captured by IR camera, including high intensities and frequent flickers. However, as shown in Fig. 1(b), because the flames from IR cameras have no color information and no distinctive appearance shape, the algorithms show unsatisfactory performances.

Smoke detection algorithms also have been developed [12–15]. Because smoke appears before the flame becomes strong, the algorithms work as an important role for the fire warning. However, due to the less visual distinctiveness of smokes than flames, the methods issued many outliers and the performance can be easily biased by training samples. The previous works are described in detail in Sect. 2.

In addition to the limitations of the individual algorithms, the algorithms are improper for 24-hours surveillance environments because general surveillance cameras selectively use a VL camera and an IR camera. The two cameras are alternatively utilized according to the current visual state such as daytime and nighttime. As shown in Fig. 1, the visual characteristics of flames are very different between VL and IR camera images, which makes it inefficient to detect the flame by a single algorithm.

To solve the problems, we propose novel algorithms for fire detection and an integrated framework of the algorithms working for all day. The proposed framework consists of four modules: integration module, VL flame detector, smoke

detector, and IR flame detector. The integration logic controls and optimizes the operation of the detectors by automatically recognizing the current state of the camera. The VL flame detector works by the patch-level flame detection scheme, showing higher performance with fewer parameters to be set than the previous method. The IR flame detector considers the temporal randomness of a flame region, which is defined as an irregular shape change of blobs. The smoke detector works based on the transparent property of smoke, meaning that the background appearance becomes dim but remains in smoke regions. The detectors and the integrated framework are validated by numerous videos including generally used videos [16] and newly captured videos, and the results verify the robustness and the generality of the proposed framework.

## 2 Related Works

For the pixel-level flame detection algorithms based on a VL camera, Phillips *et al.* [2] proposed a framework to detect flames by the colors and the temporal variation of candidate pixels. In addition, Chenebert *et al.* [3] utilized color values in HSV domain and the texture features.

Most flame detection algorithms with a VL camera are based on blob-level classifiers. Toreyin *et al.* [4] analyzed the flickering property of flames and the irregularity of flame boundary. Ham *et al.* [5] utilized fuzzy finite automation to classify the irregular patterns of candidate blobs. Morerio *et al.* [7] obtained the color information, motion models, and temporal dynamics of the blobs in parallel to detect flames.

Choi *et al.* [6] proposed the patch-level VL flame detection algorithm which extracts candidate patches from an input image and classifies the patches as a flame by their appearance and randomness.

For detecting flames with an IR camera, Maoult *et al.* [8] analyzed the property of flames with various gas types and materials, which was utilized to reject flame-like objects captured by IR cameras. Owrutskya *et al.* [9] combined the images captured by a VL camera and the ones by an IR camera to obtain high performance of the flame detector. Even though Tasdemir *et al.* [10] used a VL camera, they tried to use the properties of night flames such as a slow movement, high intensity, periodic motion, and a self-motion. Toreyin *et al.* [11] detected flames with an IR camera by utilizing the spatial and temporal changes of the wavelet information on flame boundaries.

To detect smoke, Ho [13] tried to fuse the spectral, spatial, and temporal probability densities on candidate regions. Kim and Wang [14] focused on the smoke captured in outdoor and utilized the temporal information of colors and shapes extracted from the smoke. Chen *et al.* [15] proposed a smoke detector working with two rules of a chromaticity-based static decision and a diffusion-based dynamic characteristic decision. Toreyin *et al.* [12] classified blob-level candidates by recognizing a decrease in the energy content of edges and analyzing the blob boundaries. The algorithm showed good performance by utilizing the transparent property of smoke as the proposed algorithm. However, the algorithm estimated the transparency just by comparing the energy content of edges,

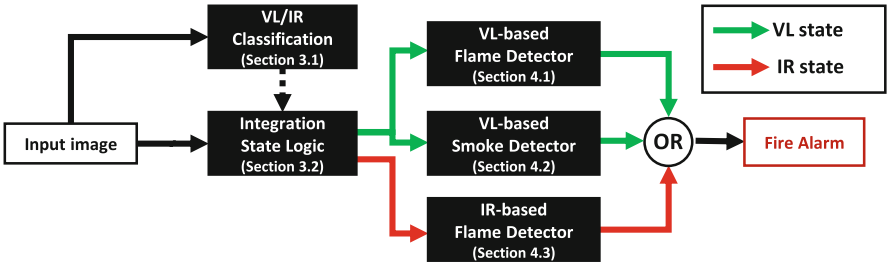



**Fig. 2. Integrated Framework.** In the integrated framework, there are four modules: an integration logic, a VL flame detector, a smoke detector, and an IR flame detector. The final fire alarm is given with any alarm from the detectors.

while the proposed algorithm represents the transparency robustly by comparing the normalized gradients of a background and a smoked region.

## 3 Integrated Framework

The integration module recognizes the current state of a camera and selectively executes the detectors which are proper to the state, as described in Fig. 2.

### 3.1 VL/IR Classification

From some cameras, the state between VL and IR camera can be obtained by a switching signal. However, for the other cameras not sending the switching signal, it is necessary to recognize the state from input frames. In this paper, the state is recognized by estimating the difference among R, G, B channel images represented by $\mathbf{F}_r$, $\mathbf{F}_g$, and $\mathbf{F}_b$, respectively. For IR cameras, R, G, B channel values of each pixel should be equivalent. Therefore, the current state is classified as IR state when

$$\begin{aligned} (1/WH)\textstyle\sum_{x,y=1}^{W,H} \|\mathbf{F}_r^{(x,y)} - \mathbf{F}_g^{(x,y)}\|_2 < \epsilon_{sw}, \\ (1/WH)\textstyle\sum_{x,y=1}^{W,H} \|\mathbf{F}_r^{(x,y)} - \mathbf{F}_b^{(x,y)}\|_2 < \epsilon_{sw}, \\ (1/WH)\textstyle\sum_{x,y=1}^{W,H} \|\mathbf{F}_g^{(x,y)} - \mathbf{F}_b^{(x,y)}\|_2 < \epsilon_{sw} \end{aligned} \tag{1}$$

are all satisfied where $W \times H$ is the image size. If one or more conditions are not satisfied, the current state is classified as VL state. The predefined threshold $\theta_{sw}$ is used to consider the noise of input frames.

When the state is switched right after satisfying the condition, the state switching can recur at evening due to the ambiguous brightness at that time. For overcoming the problem, the state is switched after a different state is continuously detected during 1000 frames.

### 3.2 Integration Logic

When the three detectors operate simultaneously, a significant computational resource is needed and many outliers can be issued. Therefore, only the detectors proper to the current state are selected to detect fire. When the camera state is VL state, the VL flame detector and smoke detector work. The alarms of the two detectors are integrated by OR operation. On the other hand, when the camera state is IR state, the IR flame detector runs.

When a fire occurs in the nighttime, the state can be changed to VL state by the bright fire. Then, the VL flame detector works even with the flame which is easy to be detected by the IR flame detector. For overcoming the problem, working detectors run continuously even with the state switching if one or more maximum scores of the detectors are larger than a predefined threshold $\epsilon_{ct}$. At that time, the other detectors do not start working to conserve the computational load.

## 4 Individual Detectors

### 4.1 VL Flame Detector

The VL flame detector is improved from Choi *et al.* [6] which shows the state-of-the-art performance. As shown in Fig. 3, the VL flame detector works with three subsequent steps: a candidate extraction, an appearance classifier, and a randomness classifier. After extracting the patch candidates from a fire-like probability map, the candidates are classified by the two classifiers in a cascade scheme. A detection map is obtained from the candidates classified as flames, and the fire alarm is determined after the spatiotemporal association of the consecutive detection maps.

**Candidate Extraction:** At first, the temporal change is estimated for every pixel to consider the temporal movement of flames. The moving pixel is determined by

$$\sum_{c=r,g,b} \|\mathbf{F}_c^{(x,y,t)} - \mathbf{F}_c^{(x,y,t-1)}\|_1 > \epsilon_{vl}, \tag{2}$$

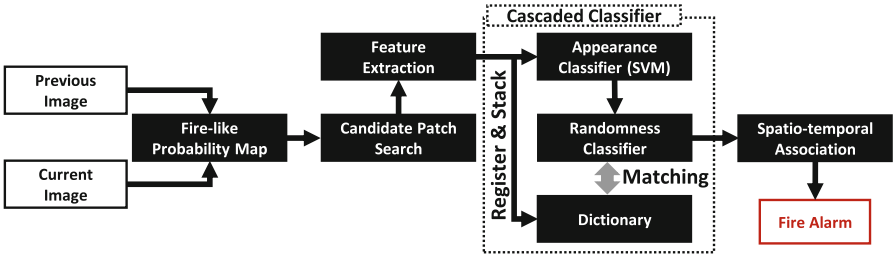



**Fig. 3. Framework of VL flame detector.** The detector extracts the patch-level candidates from an input image. Then, the candidates are classified as a flame by cascaded classifiers checking their shapes and randomnesses. After the spatiotemporal association of the results, a fire alarm is decided.

where $\epsilon_{vl}$ is a predefined threshold. For the moving pixels, the fire-like probability is estimated by using pixel-wise features, while the probability is set to 0 for the remaining static pixels. In the input frame of time $t$, a pixel at $(x, y)$ has six pixel-wise features $\mathbf{Z}^{(x,y,t)} = \{\mathbf{F}_r^{(x,y,t)}, \mathbf{F}_g^{(x,y,t)}, \mathbf{F}_b^{(x,y,t)}, \Delta_1\mathbf{F}_r^{(x,y,t)}, \Delta_1\mathbf{F}_g^{(x,y,t)}, \Delta_1\mathbf{F}_b^{(x,y,t)}\}$, which are the RGB values and the magnitude of derivative for each of RGB channels, respectively. The magnitude of derivative for each channel is estimated as

$$\Delta_1\mathbf{F}_c^{(x,y,t)} = \|\mathbf{F}_c^{(x-1,y,t)} - \mathbf{F}_c^{(x+1,y,t)}\|_1 + \|\mathbf{F}_c^{(x,y-1,t)} - \mathbf{F}_c^{(x,y+1,t)}\|_1, \quad (3)$$

where $c \in \{r, g, b\}$. The estimation is based on 1-norm for a low computational load.

From, the pixel-wise features, the fire-like probability is calculated by

$$\mathbf{P}^{(x,y)} = \mathbf{W}^T\mathbf{F}^{(x,y)}, \quad (4)$$

where $\mathbf{W} \in I\!R^6$. The fire-like probability is built to be high on the boundary of a flame because the pixels on the boundary move actively and the patches on the boundary have more distinctive appearance than the ones inside the flame. Contrary to Choi *et al.* [6] setting the weights manually, the weight vector $\mathbf{W}$ is trained by linear regression algorithm [19]. The ground truth of the probability map for the regression is obtained by blurring the binary map with Gaussian kernel, which is set to 1 on the boundary of flame.

From the estimated probability map, the local maximums are extracted as the center points of patch candidates. The local maximums are used to reduce the computational load by rejecting the neighboring patches analogous to each other. Also, to reject the outliers with the colors far from flames, the detector chooses only the candidates where the probability is above the predefined threshold $\theta_{pr}$. When the number of the selected candidates is over $N_c$, only the candidates with top $N_c$ probability values are used to limit the computational load.

**Appearance classifier:** The patch-level features are obtained by concatenating three SURF features [20] each extracted from the candidate in one of RGB channels. Contrary to Choi *et al.* [6] which extracts the features only in R channel, the distinctiveness of appearance feature would be improved by utilizing all the RGB channels.

To handle the non-linearity of classification problem, the dimension of the patch-level features are expanded by a Chi-squared kernel, following Vedaldi and Zisserman [21]. With the expanded features, the classifier is trained by linear SVM. For the training, the positive samples are extracted from the patches where the ground truth probability is above 0.8, while the negative samples are obtained randomly from the outside of flame. In the test sequence, the appearance classifier determines the candidate as a flame if the SVM margin is over 0.

**Randomness classifier:** Because the shape of a flame changes randomly by air convection, the candidates can be rejected by classifying the randomness of the shape. The method to detect the randomness is same as Choi *et al.* [6], but the used feature is different because the patch-level feature is extracted from three

channels in the proposed detector. The detail description of the randomness classifier can be referred by [6].

**Temporal association:** A current likelihood map $\mathbf{L}_{vl}$ has the same size of the input image, and all the pixels located in the detected fire parts are set to one, while the other pixels are set to zero. In order to reject remaining outliers, consecutive $\mathbf{L}_{vl}$ are temporally associated. Therefore, the final detection map $\mathbf{D}_{vl}$ with the same size of $\mathbf{L}_{vl}$ is estimated by

$$\mathbf{D}_{vl}^{(x,y,t)} = \alpha \mathbf{D}_{vl}^{(x,y,t-1)} + (1-\alpha)\mathbf{L}_{vl}^{(x,y)}, \tag{5}$$

where $\mathbf{D}_{vl}^{(x,y,0)} = 0$. When the maximum value of $\mathbf{D}_{vl}$ becomes over a predefined threshold $\zeta_{vl}$, the fire alarm is given.

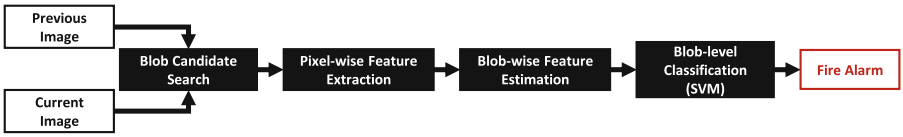



**Fig. 4. Framework of Smoke Detector.** After extracting the pixel-wise features from candidate blobs, the blob-level features are obtained from the distribution of the pixel-wise features and classified by SVM. A fire alarm is decided after the temporal association of the classification results.

### 4.2 Smoke Detector

The proposed smoke detector determines blob-level candidates as a smoke by utilizing the transparency property of smoke in addition to the general color and shape features of smoke. The transparency property means that the background is dimmed but remains at the region of smoke. Figure 4 describes the entire framework of the proposed smoke detector. At first, the blob candidates are extracted by background subtraction. Then, the pixel-wise features are extracted from the pixels of one blob candidate, which include colors, shapes, and transparency features. The blob-level feature is obtained from the distribution of the pixel-wise features, which is used for the smoke classification. Finally, the fire alarm resulted from the temporal association of the smoke regions.

**Candidate Blob Search:** Because the transparency property is weak on thick smoke, we should find candidate blobs where the smoke just expands to use the transparency property efficiently. The expanded region can be easily found by a temporal difference image. To consider the variety of expanding speed of smoke, the difference image is obtained between a current image and a previous image before $N_s$ frames as

$$\mathbf{B}_{s}^{(x,y,t)} = \sum_{c\in\{r,g,b\}} \|\mathbf{F}_{c}^{(x,y,t)} - \mathbf{F}_{c}^{(x,y,t-N_s)}\|_2. \tag{6}$$

From the binary image obtained by thresholding $B_s$ by $\epsilon_{sm}$, blob candidates are extracted by connected component algorithm [22]. To limit the computational load, only top three candidates are chosen in order of volume.

**Feature Extraction:** There are three types of pixel-wise features: colors, shapes, and transparency features. The color feature consists of six channel including L, a, b in Lab color space and R, G, B in RGB domain. The magnitude of the derivative on each of RGB channels is used as the shape feature. The magnitude is estimated by

$$\Delta_2 \mathbf{F}_c = \|(\Delta_x \mathbf{F}_c, \Delta_y \mathbf{F}_c)\|_2, \quad c \in \{r, g, b\}, \tag{7}$$

where $\Delta_x \mathbf{F}_c$ and $\Delta_y \mathbf{F}_c$ are obtained by filtering $\mathbf{F}_c$ with x-directed and y-directed Sobel window [23], respectively.

The transparency feature is obtained by two channels: a gradient correlation and a dark channel difference. The gradient correlation is based on the characteristic of the transparency where the shape of the background remains under smoke. The gradient correlation is estimated by the difference between the normalized derivative vectors on the current and previous images as

$$\mathbf{F}_{gc}^{(x,y,t)} = \sqrt{\sum_{c=\{r,g,b\}} \left[ \left( \frac{\Delta_x \mathbf{F}_c^{(x,y,t)}}{\Delta_2 \mathbf{F}_c^{(x,y,t)}} - \frac{\Delta_x \mathbf{F}_c^{(x,y,t-N_s)}}{\Delta_2 \mathbf{F}_c^{(x,y,t-N_s)}} \right)^2 + \left( \frac{\Delta_y \mathbf{F}_c^{(x,y,t)}}{\Delta_2 \mathbf{F}_c^{(x,y,t)}} - \frac{\Delta_y \mathbf{F}_c^{(x,y,t-N_s)}}{\Delta_2 \mathbf{F}_c^{(x,y,t-N_s)}} \right)^2 \right]}. \tag{8}$$

Therefore, $F_{gc}$ becomes small for the smoke because the normalized gradients of the background should be preserved through the transparent smoke.

The dark channel difference is based on dark channel prior [25]. Following the dark channel prior, air lights diffused by fog or smoke are colorless, so the minimum value of R, G, B channels becomes high on smoke regions. Following the property, the dark channel difference is estimated by

$$F_{dc}^{(x,y,t)} = \min_{c \in \{r,g,b\}} \left( F_c^{(x,y,t)} \right) - \min_{c \in \{r,g,b\}} \left( F_c^{(x,y,t-N_s)} \right). \tag{9}$$

Then, the pixel-wise features are obtained for every pixel contained in a target blob candidate. However, because the numbers of pixels in blob candidates are various, the distribution of the pixel-wise features is utilized as a blob-wise feature to represent blobs in the same dimension. The distribution is represented by a covariance matrix and a mean vector of the whole pixel-wise features contained in the target blob. Therefore, 72 channel vector composed of 64 values from the covariance matrix and 8 values from the mean vector is used as the blob-level feature.

**Classification:** The classification method is same as the method used for the VL flame detector, which utilizes the homogeneous kernel mapping [21] and linear SVM. Contrary to the previous smoke detectors [12–15] which need the labeled region of smoke to train their classifiers, the proposed algorithm do not need the labeled region if only smoke is moving in the training video. When only the smoke moves in the training video, positive samples can be obtained by applying

the two previous steps including the candidate search and feature extraction on the video. Negative samples are obtained from non-smoke videos by the same method. In the test sequence, a blob candidate is classified as smoke if its SVM margin is over 0.

**Temporal Association:** After the classification step, for a binary detection map $\mathbf{L}_{sm}$, the pixels of the blobs classified as smoke are set to one and the other pixels are zero. To remove remaining outliers, the consecutive detection maps are temporally associated as

$$\mathbf{D}_{sm}^{(x,y,t)} = \alpha \mathbf{D}_{sm}^{(x,y,t-1)} + (1-\alpha)\mathbf{L}_{sm}^{(x,y)}, \tag{10}$$

where $\mathbf{D}_{sm}^{(x,y,0)} = 0$. Then, the values of $\mathbf{D}_{sm}$ are between 0 and 1, and the detector gives a fire alarm when the maximum value of $\mathbf{D}_{sm}$ is over a predefined threshold $\zeta_{sm}$.

### 4.3 IR Flame Detector

When the signal can be represented by a repeat of basic signal, the signal shows weak randomness so that the blob would be a fire-like outlier such as a turn signal and a neon sign. Therefore, the randomness of a signal can be defined as the absence of repeated basic signal in the signal, which is utilized in the proposed IR flame detector.

The entire framework of the IR flame detector is described in Fig. 5. At first, blob candidates are extracted by grouping bright pixels. Then, blob-wise shape features are estimated from the shapes of the blobs. After tracking the blobs temporally, the temporal shape change of a blob is expressed by a blob shape signal obtained by stacking the shape features of the tracked blobs. Based on the randomness of the signal, a classifier determines the candidate blobs as a flame. Finally, the classified flame blobs are temporally associated, and a fire alarm is given according to the temporally associated result.

**Candidate Blob Search:** Because the IR flame detector works on an IR camera, the input image is transformed to a gray image $\mathbf{I}$ by averaging the whole

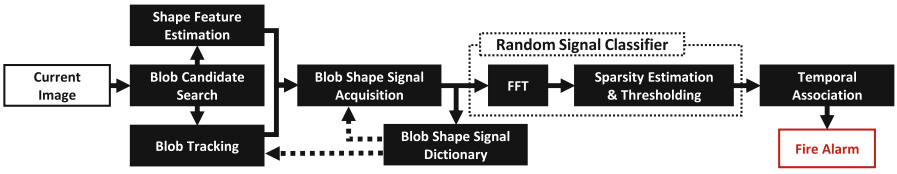



**Fig. 5. Framework of IR Flame Detector.** After extracting the bright regions as candidate blobs, a shape feature is obtained for each blob. The candidate blobs are tracked continuously, and the shape features are stacked to acquire a blob shape signal. By testing the randomness of the shape signal, the flame candidate is determined. After the temporal association of the flame candidates, a fire alarm is given.

channels. In order to regard the brightness of flames in an IR camera, a binary map $\mathbf{B}_{ir}$ is estimated by thresholding $\mathbf{I}$ as

$$\mathbf{B}_{ir}^{(x,y)} = \begin{cases} 1 & \text{if } \mathbf{I}^{(x,y)} > \epsilon_{ir} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $\epsilon_{ir}$ is a predefined threshold. After applying a denoising sequence with dilation and erosion filters, the blob candidates are obtained by connected component algorithm [22]. Among all the blobs, the small blobs are rejected, which contain fewer pixels than a predefined number $N_{sb}$.

**Shape Feature Estimation:** From $l$-th blob candidate, a shape feature $V^l$ is estimated by summing the variances of x and y positions of the containing pixels as

$$\begin{aligned} V^l &= \textstyle\sum_{i=1}^{N^l} (x_i^l - m_x^l)^2 + (y_i^l - m_y^l)^2, \\ \mathbf{m}^l &= (m_x^l, m_y^l) = (\textstyle\sum_{i=1}^{N^l} x_i^l / N^l, \sum_{i=1}^{N} y_i^l / N^l), \\ \mathbf{X}^l &= \{(x_1^l, y_1^l), ..., (x_{N^l}^l, y_{N^l}^l)\}, \end{aligned} \tag{12}$$

where $\mathbf{X}^l$ is a position set of pixels labeled as $l$-th blob and $N^l$ is the number of the pixels.

**Blob Shape Signal Acquisition:** For tracking the blobs on consecutive frames, a tracking algorithm based on the distances among the blobs is used. A previous blob at time $(t-1)$ is connected to a current blob at time $(t)$ if the current blob is closest to the previous blob and the distance between the center points of the two blobs is less than a predefined threshold $\epsilon_{di}$. When the previous blob has no connection, the previous blob becomes a finished blob. On the other hand, the current blob becomes an initial blob when the blob is not connected to any previous blob.

For the connected blobs, the shape features are stacked temporally to build the blob shape signal. The size of the blob shape signal for one blob is limited by $N_M$, and the oldest blob value is removed in a first-in-first-out scheme when the size goes over the limitation. Algorithm 1 describes the method acquiring the blob shape signal in detail.

$\mathbf{M}$ is a set of blob center points, $\mathbf{S}$ is a set of blob shape signals, $\mathbf{V}$ is a set of shape features, and $K$ and $L$ are the number of candidate blobs at $(t-1)$ and $(t)$, respectively.

**Classification:** The blobs tracked over $N_M$ frames are applied to the classification. For verifying the absence of a repeated basic signal, the sparsity of the signal in Fourier domain is estimated. When the entire signal $\mathbf{s}(t)$ with length $T$ is repeated $N$-times by a basic signal $\mathbf{s}^o(t)$, the signal in time domain can be simplified as $\mathbf{s}(t) = \mathbf{s}^o(t) * \sum_{k=0}^{N-1} \delta\left(t - \frac{kT}{N}\right)$, which can be represented in Fourier domain as

$$\mathcal{F}\left(\mathbf{s}(t)\right) = \mathcal{F}\left(\mathbf{s}^o(t)\right) \sum_{k=-\infty}^{\infty} \delta\left(w - k\frac{2\pi N}{T}\right), \tag{13}$$

**Algorithm 1.** Blob Shape Signal Acquisition

$\mathbf{M}^{(0)} = \emptyset$, $\mathbf{S}^{(0)} = \emptyset$,
$\mathbf{M}^{(t-1)} = \{\mathbf{m}^{1,(t-1)}, ..., \mathbf{m}^{K,(t-1)}\}$, $\mathbf{S}^{(t-1)} = \{\mathbf{S}^{1,(t-1)}, ..., \mathbf{S}^{K,(t-1)}\}$
$\mathbf{M}^{(t)} = \{\mathbf{m}^{1,(t)}, ..., \mathbf{m}^{L,(t)}\}$, $\mathbf{V}^{(t)} = \{V^{1,(t)}, ..., V^{L,(t)}\}$
**for** $l = 1...L$ **do**
  $r = arg_k \min_{k \in \{1,...,K\}} (\|\mathbf{m}^{l,(t)} - \mathbf{m}^{k,(t-1)}\|_2)$
  **if** $\|\mathbf{m}^{l,(t)} - \mathbf{m}^{r,(t-1)}\|_2 < \theta_d$ **then**
    $\mathbf{S}^{l,(t)} = [\mathbf{S}^{r,(t-1)}, V^{l,(t)}]$
    **if** $length\left(\mathbf{S}^{l,(t)}\right) > N_M$ **then**
      $\mathbf{S}^{l,(t)} = \mathbf{S}^{l,(t)}(2 : end)$
    **end**
  **else**
    $\mathbf{S}^{l,(t)} = [V^{l,(t)}]$
  **end**
**end**

where $\mathcal{F}(\bullet)$ returns a magnitude of an input signal in Fourier domain. Therefore, the signal in Fourier domain becomes sparse due to the multiplication of delta functions. On the contrary, a random signal has no sparse element in Fourier domain because the entire frequency elements are necessary for representing the random signal [24].

According to the property, the temporal randomness of a blob candidate is measured by the sparsity of the blob shape signal in Fourier domain. The sparsity of $l$-th blob $F_{sp}^{l}$ is estimated by

$$F_{sp}^{l} = \left\| scad\left(\mathcal{F}\left(\mathbf{S}^{l,(t)}\right)\right)\right\|_1 \tag{14}$$

In the equation, the scad function [26] approximates the sparsity when the input signal contains noises. Finally, the $l$-th candidate blob is determined as a flame if $F_{sp}^{l}$ is larger than a predefined threshold $\theta_{sp}$.

**Temporal Association:** A binary detection map $\mathbf{L}_{ir}$ is obtained by setting one for the pixels on the flame blobs and zero for the other pixels. The detection map $\mathbf{D}_{ir}$ is temporally associated to remove remaining outliers as

$$\mathbf{D}_{ir}^{(x,y,t)} = \alpha \mathbf{D}_{ir}^{(x,y,t-1)} + (1-\alpha)\mathbf{L}_{ir}^{(x,y)}, \tag{15}$$

where $\mathbf{D}_{ir}^{(x,y,0)} = 0$. Finally, when the maximum value of $\mathbf{D}_{ir}$ is over a predefined threshold $\zeta_{ir}$, the fire alarm is decided to result.

## 5 Experiments

### 5.1 Implementation

The used parameters are represented in Table 1 and decided experimentally. The size of SURF used for the VL flame detector was $32 \times 32$. The orders of

**Table 1.** Summary of parameters

| | Parameter | Value | Description |
|---|---|---|---|
| Integration module | $\epsilon_{sw}$ | 0.002 | Threshold to decide VL/IR state |
| | $\epsilon_{ct}$ | 0.05 | Min. detector score to stop switching the state |
| VL flame detector | $\epsilon_{vl}$ | 0.3 | Threshold for moving pixels |
| | $\theta_{pr}$ | 0.5 | Threshold for color classifier |
| | $N_c$ | 20 | Max. number of candidates |
| | $N_d$ | 10000 | Max. size of dictionary |
| Smoke detector | $N_s$ | 30 | Frame gap for a difference image |
| | $\epsilon_{sm}$ | 0.2 | Threshold for candidate extraction |
| IR flame detector | $\epsilon_{ir}$ | 0.98 | Threshold for candidate pixels |
| | $N_{sb}$ | 800 | Min. size of blobs |
| | $\epsilon_{di}$ | 300 | Min. distance for blob tracking |
| | $N_M$ | 100 | Max. length of shape signal |
| | $\theta_{sp}$ | 0.005 | Threshold for classification |
| Etc. | $\alpha$ | 0.99 | Weight for temporal association |

homogeneous kernel mapping were set to 3 for VL flame detector and 5 for VL smoke detector.

The entire framework was implemented in C++ with OpenCV and VLFeat library. With one core of 3.40 GHz CPU and 16 GB memory, the computational speed was 35.71 fps in VL state and 47.62 fps in IR state for $1280 \times 720$ videos, so that the proposed framework worked in real-time.

### 5.2 Experimental Details

The experiments were executed by three datasets of which all the positive videos are different. The first dataset is 'General Dataset' consisting of 41 fire videos and 54 fire-like outlier videos, which were gathered from previous papers [16] and other uploaded datasets [17,18]. The videos were captured only by VL camera in various environments including indoor and outdoor scenes. The second dataset is 'Indoor Dataset' consisting of 99 fire videos and 141 fire-like outlier videos, which were newly captured indoor. Among the fire videos, 54 videos were captured by VL cameras and the other videos were by IR cameras to confirm the 24-hours performance of the proposed framework. Among the fire-like outlier videos, 102 videos were captured by VL cameras and the other 39 videos were by IR cameras. For verifying the robustness, the videos were captured in numerous environments, such as a house, a storehouse, and an office. The third dataset is 'Smoke Dataset' consisting of 60 smoke videos, which were captured by ourselves with the various environment. In all the videos of the dataset, smoke occurs to test the performance of the proposed smoke detector, because smoke

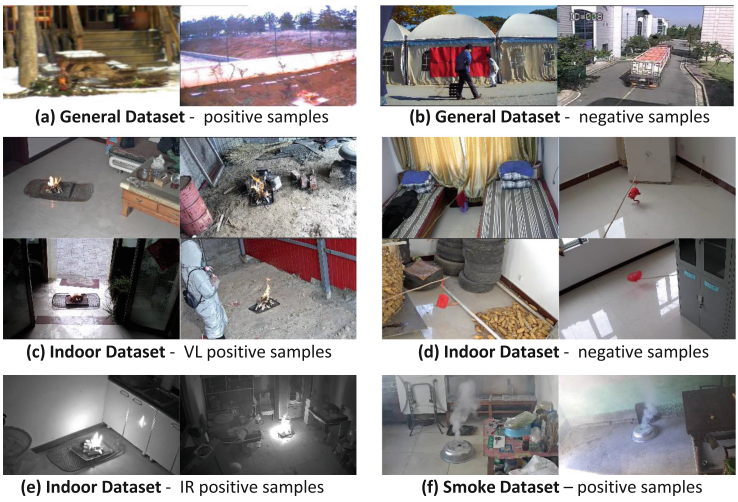



**Fig. 6. Sample images of used datasets.** We used three datasets: (a-b) General Dataset, (c-e) Indoor Dataset, and (f) Smoke Dataset.

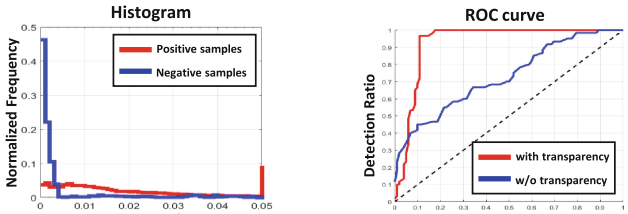
is not shown in many fire videos of the other two datasets. The fire-like outlier videos captured by a VL camera among Indoor Dataset were used as the negative videos of Smoke Dataset (Fig, 6).

General Dataset videos are utilized as training samples for the experiments using Indoor Dataset. On the other hand, the framework is trained by General Dataset for the experiments of General Dataset.

The performances of algorithms are estimated by two measures: detection ratio and false positive ratio. The detection ratio is obtained by dividing the number of correctly detected fire videos by the number of the tested fire videos. The detection of a video is determined when one or more fire alarms are detected from the fire video. The false positive ratio is evaluated by dividing the number of wrongly detected negative videos by the number of the tested negative videos. Because a negative video becomes a wrongly detected negative video even with one false alarm, the measure is very challenging for fire detection frameworks. As the algorithm shows high detection ratio with a low false positive ratio, the performance of the algorithm becomes good. With a ROC curve of detection ratio and false positive ratio, the performances of algorithms can be compared by the area under the curve. In the following experiments, the ROC curves were obtained by varying the thresholds $(\zeta_{vl}, \zeta_{sm}, \zeta_{ir})$ which determine the fire alarm from the temporally associated detection maps.

### 5.3 Self-comparison

**Effectiveness of Sparsity Estimation in IR Flame Detector.** To show the effectiveness of sparsity estimation used in the IR flame detector, we compared the distributions of the sparsity values estimated from flame videos and




(a) Histogram of sparsity values (b) Effectiveness of transparency features

**Fig. 7. Results of self-comparison experiments.** (a) shows the difference in the distribution of the sparsity values from IR flames and the ones of flame-like objects. (b) presents the effectiveness of the proposed transparency features in the smoke detector.

flame-like outlier videos. A flame histogram was built from the sparsity values of all the blob candidates extracted from the flame videos captured by IR cameras among Indoor Dataset. An outlier histogram was obtained by the sparsity values estimated from the negative videos captured by IR cameras among Indoor Dataset.

The flame histogram and outlier histogram are compared in Fig. 7(a). As shown in the histograms, about 80 % of sparsity values from outliers were located in 0–0.005 region, while about 90 % of sparsity values from flames were distributed out of the region. Therefore, as suggested in this paper, most of the flame-like outliers with IR cameras can be rejected by the sparsity of a shape signals in Fourier domain.

**Effectiveness of Transparency Features in Smoke Detector.** For validating the effectiveness of transparency features used for the smoke detector, we compared the proposed detector with a naive detector which utilizes same features excepting for the two transparency features. Smoke Dataset was used as a test set, and the two detectors were trained by five positive videos with smoke and all the negative videos of General Dataset.

The experimental results are shown in Fig. 7(b). As shown in the results, the proposed detector utilizing the transparency features showed much better performance than the naive detector. From the results, the proposed transparency features can be verified as robust features to distinguish smokes from outliers.

### 5.4 Performance

**General Dataset.** General Dataset was used to show the improvement of the VL flame detector from the previous flame detection algorithms. The flame detector was compared with Choi *et al.* [6], which has shown the state-of-the-art performance. The performance comparison is shown in Fig. 8(a), verifying that the proposed detector shows better performance than the state-of-the-art algorithm in general environments.

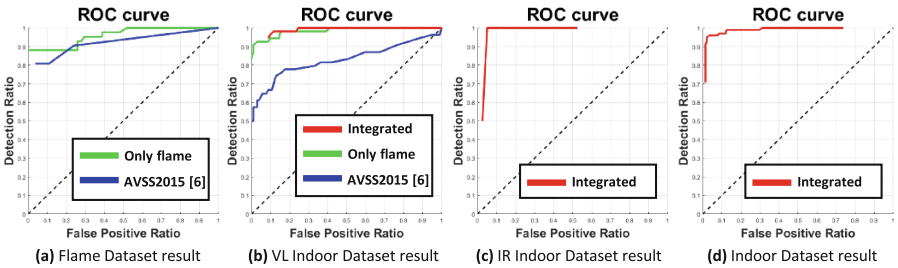



**Fig. 8. Performance Evaluation.** (a) shows the performance of VL flame detector with General Dataset. (b) presents the performance of integrated framework with the VL videos of Indoor Dataset. (c) is the results of integrated framework with the IR videos of Indoor Dataset. (d) shows the performance of integrated framework with the entire videos of Indoor Dataset.

**Indoor Dataset.** Indoor Dataset can be divided by VL Indoor Dataset and IR Indoor Dataset according to the used camera. The performance estimated by VL Indoor Dataset and IR Indoor Dataset shows the respective performance of the proposed framework for each state.

In the experiment using VL Indoor Dataset, we compared three algorithms including Choi *et al.* [6], the VL flame detector, and the integrated framework, as shown in Fig. 8(b). The ROC curve of the integrated framework was obtained by changing $\zeta_{vl}$, while $\zeta_{ir}$ and $\zeta_{sm}$ were fixed by 0.9 and 0.6, respectively. The proposed algorithm shows better performance than Choi *et al.* [6]. Also, it can be confirmed that the proposed algorithm shows much better performance in indoor scenes than in general scenes by comparing the performance gap between [6] and the proposed framework. When the performances of the VL flame detector and the integrated framework are compared, it can be checked that the smoke detector is helpful for fire detection.

Only the proposed algorithm was tested for IR Indoor Dataset, and the results are shown in Fig. 8(c). In the experiment, IR flame detector wrongly detects outliers from only two of 39 negative videos, while all the positive videos are correctly detected.

By testing the integrated framework for Indoor Dataset, the 24-hour performance of the framework was evaluated as shown in Fig. 8(d). Because the switching signal from a camera cannot be used in the experiment, the proposed VL/IR classification method was applied to recognize the current camera state. Following the results, the proposed integrated framework showed good performance in various indoor scenes.

## 6 Conclusion

In this paper, we proposed the framework for 24-hours fire detection, including the integration module and the three novel fire detectors of VL flame detector, smoke detector, and IR flame detector. The integration module selectively

operates the detectors proper to the current state of VL/IR cameras. The VL flame detector was improved from the previous patch-level flame detector, showing higher performance than the previous one. The smoke detector was newly developed, which considered the transparency property of smoke by gradient correlations and dark channel differences. Also, the IR flame detector was built to find flames with an IR camera by testing the randomness in the temporal changes of a blob shape. The three detectors and the integrated framework were tested by three datasets consisting of numerous videos, showing satisfactory performances to be applied to real scenes even in real-time.

**Acknowledgment.** This work was partly supported by the ICT RD program of MSIP/IITP[B0101-15-0552, Development of Predictive Visual Intelligence Technology], the Brain Korea 21 Plus Project, and EU FP7 project WYSIWYD under Grant 612139.

## References

1. Ahrens, M.: Characteristics of home fire victims. National Fire Protection Association (2014). http://www.nfpa.org/
2. Phillips III, W., Shah, M., Lobo, N.V.: Flame recognition in video. PRL **23**, 319–327 (2002)
3. Chenebert, A., Breckon, T.P., Gaszczak, A.: A non-temporal texture driven approach to real-time fire detection. In: ICIP (2011)
4. Toreyin, B.U., Dedeoglu, Y., Gudukbay, U., Cetin, A.E.: Computer vision based method for real-time fire and flame detection. PRL **27**, 49–58 (2006)
5. Ham, S., Ko, B., Nam, J.: Fire-flame detection based on fuzzy finite automation. In: ICPR (2010)
6. Choi, J., Choi, J.Y.: Patch-based fire detection with online outlier learning. In: AVSS (2015)
7. Morerio, P., Marcenaro, L., Regazzoni, C.S., Gera, G.: Early fire and smoke detection based on colour features and motion analysis. In: ICIP (2012)
8. Maoult, Y.L., Sentenac, T., Orteu, J.J., Arcens, J.P.: Fire detection, a new approach based on a low-cost CCD camera in the near infrared. Process Saf. Environ. Prot. **85**, 193–206 (2007)
9. Owrutsky, J.C., Steinhurst, D.A., Minor, C.P., Rose-Pehrsson, S.L., Williams, F.W., Gottuk, D.T.: Long wavelength video detection of fire in ship compartments. Fire Saf. J. **41**, 315–320 (2006)
10. Tasdemir, K., Gunay, O., Toreyin, B.U., Cetin, A.E.: Video based wildfire detection at night. In: Signal Processing and Communications Applications Conference (2009)
11. Toreyin, B.U., Cinbis, R.G., Dedeoglu, Y., Cetin, A.E.: Fire detection in infrared video using wavelet analysis. Opt. Eng. **46**, 067204 (2007)
12. Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: Contour based smoke detection in video using wavelets. In: European Signal Processing Conference (2006)
13. Ho, C.: Machine vision-based real-time early flame and smoke detection. Meas. Sci. Technol. **20**, 045502 (2009)
14. Kim, D., Wang, Y.: Smoke detection in video. In: World Congress on Computer Science and Information Engineering (2009)

15. Chen, T., Yin, Y., Huang, S., Ye, Y.: The smoke detection for early fire-alarming system base on video processing. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing (2006)
16. Ko, B., Ham, S., Nam, J.: Modeling and formalization of fuzzy finite automata for detection of irregular fire flames. Trans. Circuits Syst. Video Technol. **21**, 1903–1912 (2011)
17. Foggia, P., Saggese, A., Vento, M.: Real-time fire detection for video surveillance applications using a combination of experts based on color, shape and motion. Trans. Circuits Syst. Video Technol. (2015)
18. Lascio, R.D., Greco, A., Saggese, A., Vento, M.: Improving fire detection reliability by a combination of videoanalytics. In: International Conference on Image Analysis and Recognition (2014)
19. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics **12**, 55–67 (1970)
20. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (SURF). CVIU **110**, 356–359 (2008)
21. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. TPAMI **34**, 480–492 (2012)
22. Samet, H., Tamminen, M.: Efficient component labeling of images of arbitrary dimension represented by linear bintrees. TPAMI **10**, 579–586 (1988)
23. Sobel, I.: History and Definition of the Sobel Operator, Report (2014)
24. Oppenheim, A.V., Schafer, R.W.: Discrete-Time Signal Processing, 3rd edn. Pearson, Upper Saddle River (2010)
25. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. TPAMI **33**(12), 2341–2353 (2011)
26. Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**, 1348–1360 (2001)