Fire Detection in Video Sequences Based on Static and Dynamic Features

Rykhard Bohush¹, Nadeja Brouka¹ and Sergey Ablameyko²

1. Department of Computer Systems and Networks, Faculty of Information Technologies, Polotsk State University, Novopolotsk 211440, Belarus

2. Department of Information Management Systems, Faculty of Applied Mathematics and Computer Science, Belarussian State University, Minsk 220050, Belarus

Received: July 9, 2013 / Accepted: November 28, 2013 / Published: December 25, 2013.

Abstract: In this paper we propose the efficient smoke and flame detection algorithms for intelligent video surveillance systems. Our algorithms consider dynamic and static features of smoke and flame including contrast, color, texture features and motion information. For smoke detection, the approach uses the motion and contrast as the two key features of smoke. Motion is a primary sign and it is used at the beginning for candidate areas extraction from a current frame. Furthermore, the direction of smoke distribution is also considered as an estimated movement based on the optical flow technique. For flame detection, we use the color image segmentation technique. Firstly, the temporal and spatial wavelets are analyzed. Then, color and texture features for candidate flame regions are extracted. Texture features are defined based on the normalized gray level co-occurrence matrix after computation of local binary pattern. Experimental results asserted the advantages of the proposed fire detection in the diversity of test video sequences.

Key words: Flame and smoke detection, background subtraction, contrast analysis, wavelet transform, texture segmentation.

1. Introduction

Early fire detection on open spaces, in buildings, in territories of the industrial enterprises is an important component for making any system of fire safety. Traditional fire detectors which have been widely applied in the buildings are based on infrared sensors, optical sensors, or ion sensors that depend on certain characteristics of fire, such as smoke, heat or radiation. Such detection approaches require a position of sensor in very close proximity to fire or smoke and often give out false alarms. So they may be not reliable and cannot be applied into open spaces and larger areas. Due to the rapid developments in digital camera technology and video processing techniques, intelligent video surveillance systems have been

installed in various public places for monitoring. Therefore, there is a noticeable trend to use such systems for early fire detection with special software applied.

We propose algorithms for smoke and flame detection on color video sequences obtained from a stationary camera. Smoke detection algorithm consists of the following steps: (1) preprocessing; (2) slowly moving areas and pixels segmentation in a current input frame based on adaptive background subtraction; (3) merge slowly moving areas with pixels into blocks; (4) classification of the blobs obtained before. Flame detection algorithm is composed of basic steps: (1) candidate flame regions detection; (2) color and texture features extraction; (3) object classification.

2. Related Work

Generally, the video processing-based fire detection

Corresponding author: Rykhard Bohush, Ph.D., research fields: image processing, intelligent video analysis and pattern recognition. E-mail: rbohush@gmail.com.

algorithms are carried out using two principal characteristics of fire, which are flame and smoke [1]. Video processing based on fire detection still has challenges. This is mainly due to the following reasons: diverse background, lighting and none of the primitive image features such as intensity, motion, edge and texture characterizes smoke and flame.

Motion information provides a key as the pre-condition to locate the possible smoke regions. The algorithm of background subtraction is traditionally applied to movement definition in video sequence [2-5]. Common techniques use the adaptive Gaussian mixture model to approximate the background modeling process such as Refs. [2, 3]. In Ref. [6], optical flow calculation is applied to detection of movement of a smoke. The drawbacks of the given approach are high sensitivity to noise and high computational cost. Algorithms based on color and dynamic characteristics of a smoke are applied for classification of the given moving blocks. In Ref. [7], the algorithm comparative evaluation of the histogram-based pixel level classification is considered. In this algorithm, the training set of video sequences with smoke is applied to the analysis. Methods based on the preliminary training highly depend on the quality of training set classification. It demands much qualitative characteristics of the processed images. The area of decreased high frequency energy component is identified as smoke using wavelet transforms [2, 3]. However, change of scene illumination can be contours degradation reason. Therefore, such approach requires additional estimations. Color information is also used for identifying smoke in video. Smoke color at different stages of ignition and depending on a burning material is distributed in a range from almost transparent white to saturated gray and black. In Ref. [2] decrease in value of chromatic components U and V of color space YUV is estimated.

Phillips et al. [8] use color model based on Gaussian distribution to the color histogram for flame

detection. This algorithm employs information gained through both color and temporal variation to detect fire. In Ref. [9], flame is detected by analyzing the video in the wavelet domain in addition to ordinary motion and color clues. The algorithm checks flicker in flames using 1-D temporal wavelet transform and color variation in fire-colored moving regions using 2-D spatial wavelet transform. Töreyn et al. [10] use the hidden Markov models and the wavelet transform for flickering pixels detection that indicate the presence of flames. The hidden Markov models are used temporally and spatially to determine if flame colored pixels flicker or not. In Ref. [11], video fire detection algorithm based on the covariance texture representation method is considered. The algorithm does not use a background subtraction method and therefore it can be used with moving cameras, but he works well when the fire is clearly visible and in close range so that the flicker and irregular nature of flames are observable. Chenebert et al. [12] use a combined texture and color based feature descriptor as an input to a trained classifier based detection boundary. Feature classification is performed based on isolation of candidate fire pixel regions using a basic color spectroscopy approach and combined color-texture classification of these regions using a trained classification approach.

3. Smoke Detection

The proposed algorithm uses motion and contrast as the two key features for smoke detection. The preprocessing block applies some methods of image processing which increases the performance of the proposed detection algorithm and reduces false alarms. Frame preprocessing block comprises three steps: grayscale transformation, histogram equalization and the discrete wavelet of the current input frame. Cameras and image sensors must usually not only deal with the contrast on a scene but also with the image sensors exposure to the resulting light on that scene. Histogram equalization is a most commonly used

method for improvement of contrast image characteristics. To resize the image and to remove high frequencies on horizontal, vertical and diagonal details, the discrete wavelet transform to Haar basis is applied. Wavelet transform to Haar basis is the simplest and the fastest algorithm that is important for systems of video processing.

In the course of the distribution, a smoke is being gradually blended to the background. Our adaptive algorithm of background subtraction considers this characteristic of a smoke and is based on the ideas stated in works of Refs. [3, 13]. A background image B_t at time instant t is recursively estimated from the image frame I_{t-1} and the background image B_{t-1} of the video as follows [13]:

$$B_t(x, y) = \begin{cases} \alpha B_{t-1}(x, y) + (1 - \alpha) I_{t-1}(x, y), & \text{if } (x, y) \text{ is moving;} \\ B_{t-1}(x, y), & \text{if } (x, y) \text{ is stationary.} \end{cases} \quad (1)$$

where (x, y) represent a pixel video frame and α is a adaptation parameter between 0 and 1. As the area of a smoke frame by frame grows slowly that the pixels belonging to a smoke, quickly did not fix in a background, value α should be close to 1.

At the initial moment of time $B_0(x, y) = I_0(x, y)$. Pixel (x, y) belongs to moving object if the following condition is satisfied [13]:

$$\begin{aligned} & (|I_t(x, y) - I_{t-1}(x, y)| > T_t(x, y)) \& \\ & (|I_t(x, y) - I_{t-2}(x, y)| > T_t(x, y)) \end{aligned} \quad (2)$$

where $I_{t-2}(x, y), I_{t-1}(x, y), I_t(x, y)$ values of intensity of pixel (x, y) at time instant $t-2, t-1$ and t respectively; $T_t(x, y)$ is adaptive threshold for pixel (x, y) at time instant t calculated as follows:

$$T_t(x, y) = \begin{cases} \alpha T_{t-1}(x, y) + (1 - \alpha)(5 \times \\ |I_{t-1}(x, y) - B_{t-1}(x, y)|) - \\ \text{if } (x, y) \text{ is moving;} \\ T_{t-1}(x, y), & \text{if } (x, y) \text{ is stationary.} \end{cases} \quad (3)$$

At the initial moment of time $T_0(x, y) = \text{const} > 0$.

Accurate separating of a foreground object from the background is the main task of digital matting. Porter and Duff [14] introduced the blending parameter (so-called alpha channel) as a solution of this problem

and a mean to control the linear combination of foreground and background components. Mathematically the current frame I_{t+1} is modeled as a combination of foreground F_{t+1} and background B_t components using the blending parameter β :

$$I_{t+1}(x, y) = \beta F_{t+1}(x, y) + (1 - \beta) B_t(x, y) \quad (4)$$

For opaque objects value of β is equal to 1, for transparent objects value of β is equal to 0 and for the semitransparent objects, such as smoke, value of β lays in a range from 0 to 1. As it is shown further in this section, we have experimentally established the optimum value for β , to be equal to 0.38.

So, as soon as we have obtained B_t component on background update step, current frame I_{t+1} and set β to 0.38, we can estimate the foreground component F_{t+1} . Then we apply the threshold processing to receive the binary foreground F_{bin} :

$$F_{bin} = \begin{cases} 1, & \text{if } (F_{t+1} > 245) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

At the current step of algorithm, we have 2 parameters α and β which are necessary to be estimated. Optimum values of α and β can be estimated using ROC (receiver operating characteristic) analysis. For estimation implementation the training set from five video sequences of the 200 frames length which contain and do not contain smoke were used. Using the ground truth regions which have been online marked as a smoke in the training frames, rates of true and false detection were calculated for the whole frame set. We received a background for each value of α within a range of (0, 1). After that we applied a background subtraction and thresholding to each frame from a training set, and then calculated TPR (true positive rate) and FPR (false positive rate). For each value of α , the average TPR and FPR is evaluated on a training frame set and used in the ROC curve. Using the ROC curve, an optimum value for α can be easily selected for the smoke detection algorithm based on a pre-defined correct detection versus false detection rates. It is necessary to choose such value of α that slowly

moving objects will not join a background too quickly, i.e. that a smoke will not fixed in a background too fast. At the given stage of algorithm high TPR is important and high enough FPR is acceptable as it is necessary to receive as much as possible pixels for the analysis and incorrectly classified pixels should be excluded at the following stages. Therefore we have established an α value equals to 0.95. Similarly using the training frame set, receiving a foreground component F_{t+1} and after that the foreground F_{bin} and counting FPR and TPR for all values β from a range (0, 1) with the step 0.001 we build a ROC curve for β . Value of β has been chosen to be equal to 0.38, because at such value of β provides high TPR and low FPR.

On the next step of smoke detection algorithm to clear of noise and to connect of moving blobs the connected components analysis is used. This form of analysis takes in a noisy input foreground. Morphological operations are applied to reduce the noise:

- morphological opening to shrink areas of small noise:

$$S \circ M = (S(-)M) \oplus M \quad (6)$$

where, S is image, M is structuring element 3×3 ;

- morphological closing to rebuild the area of surviving components that was lost in opening:

$$S \bullet M = (S \oplus M)(-)M \quad (7)$$

where, M is structuring element 3×3 .

After that, edge analysis is applied and contours with a small perimeter are removed.

Blocks matching approach for optical flow calculation assumes that the frame is divided into small regions called blocks. It considers a primary direction of smoke propagation. It shows in Ref. [15] that global direction of smoke is 0-45°. This statement allows to simplify procedure of blocks matching detection and, hence, to considerably reduce number of calculations. Blocks are typically squares and contain some number of pixels. These blocks do not overlap. In our implementation frames in the size 320×240 pixels divided into blocks 2×2 pixels. Block

matching algorithm attempt to divide both the previous and current frames into such blocks and then compute the motion of these blocks. Each block of size 2×2 can move in eight possible directions. Our implementation searches in three directions of the original block $q_{x,y}^{prev}$ (in the previous frame) and compares the candidate new blocks $q_{x-1,y-1}^{curr}$, $q_{x,y-1}^{curr}$ and $q_{x+1,y-1}^{curr}$ (in the current frame) with the original. This comparison is calculated as follows:

$$F(q_{x,y}^{prev}, q_{x+k,y-1}^{curr})_{k \in \{-1,0,1\}} = \left(\frac{\min(I_{i,j}^{prev}, I_{i,j}^{curr})}{\max(I_{i,j}^{prev}, I_{i,j}^{curr})} \right) \quad (8)$$

where, $I_{i,j}^{prev}$ is the intensity value of pixel on the previous frame, belonging to the block $q_{x,y}^{prev}$; $I_{i,j}^{curr}$ is the intensity value of pixel on the current frame, belonging to the block $q_{x,y}^{curr}$; N is count of blocks into which divided the previous and current frame. The block $q_{x,y}^{prev}$ in the previous frame will correspond to the block in the current frame if function F has the maximum value. Optical flow calculation (function F) is done only for the blocks belonging to the foreground (Fig. 1b). The result of this step is the set of vectors c_s having a direction corresponding to primary propagation of smoke (Fig. 1c).

From each blob from the previous steps we calculate percentage ρ of blocks which have moved in primary direction of smoke:

$$\rho = \frac{c_s}{c} \cdot 100 \frac{dy}{dx} \% \quad (9)$$

where c is the total number of blocks on a current frame, and Weber contrast C_w :

$$C_w = \frac{1}{n} \sum_{i=1}^n \frac{F_{t+1}(x,y) - B_t(x,y)}{B_t(x,y)} \quad (10)$$

where, $F_{t+1}(x,y)$ -value of pixel intensity (x,y) at time instant t , belonging to a blob; $B_t(x,y)$ -value of background pixel intensity (x,y) at time instant t under blob; n -number of the pixels belonging to a blob. If the blob has successfully checked out that we classify it as a smoke. Experimentally established values $C_w > 0.5$ and $\rho > 20\%$ allow efficient distinguishing a smoke from objects with similar

Fig. 1 (a) The current frame, (b) the clean up foreground by the connected components analysis and (c) the results of optical flow calculation.

behavior: a fog, shadows from slowly moving objects and patches of light.

4. Flame Detection

CFR (candidate flame regions) are initially isolated based on color image segmentation, wavelet analysis and post-processing for binary mask of CFR. For image segmentation and flame-colored pixels we use RGB space and rule-set [16]:

$$\begin{cases} R(x, y) > R_{mean} \\ R_{mean} = \frac{1}{K} \sum_{i=1}^K R(x_i, y_i) \\ R(x, y) > G(x, y) > B(x, y) \\ 1.1403g - 0.0759 \leq r \leq -0.9889g + 0.9913 \\ 0.8459b + 0.0482 \leq g \leq -0.4608b + 0.4954 \end{cases} \quad (11)$$

where, $R(x, y)$, $G(x, y)$, $B(x, y)$ are Red, Green and Blue values of pixel respectively; K is the total number of pixels in image; R_{mean} is the mean of Red channel of pixels where a change is not detected and rgb are normalized color space for RGB. The rgb

Fig. 2 (a) The input frame and (b) binary mask after image segmentation.

space can be represented:

$$\begin{cases} r = R / R + G + B \\ g = G / R + G + B \\ b = B / R + G + B \end{cases} \quad (12)$$

We set a binary mask with include fire-colored pixels (Fig. 2).

After that, we use one-dimensional temporal wavelet transform for analysis of checks flicker in flames and two-dimensional spatial wavelet transform for analysis of color variation in flame-colored regions. For temporal wavelet analysis each red component of RGB space fire-colored pixel is fed to a two stage-filter bank. If the pixel is part of a flame boundary at some time, then there will be several spikes in one second due to transitions from background colors to flame colors and vice versa [3]. In contrast to work on fire detection [3], for reducing the time we do not use moving region detection.

Spatial wavelet analysis using low-high (SI_{lh}), high-low (SI_{hl}) and high-high (SI_{hh}) wavelet sub-images are added to obtain these images for parameter calculation of pixel (x, y) [3]:

$$v = \frac{1}{M \times N} \sum_{x,y} \left((SI_{lh})^2 + (SI_{hl})^2 + (SI_{hh})^2 \right) \quad (13)$$

where $M \times N$ is the number of pixels in the fire-colored moving region.

As a result we receive a mask for CFR. On a final steps clearing of noise based on morphological opening and closing, area filtering by size for mask is used. If the area perimeter is equal or more than 1/8 perimeters of a frame, this area is CFR.

After that, texture features for candidate flame regions are extracted. Haralick [17] suggested the use

of GLCM (grey level co-occurrence matrices) to extract second order statistics from an image. This method is based on the joint probability distributions of pairs of pixels. Textural features are calculated with use of co-occurrences matrix:

$$P_d^\theta(i, j) = \{((r, s), (t, v)) : I(r, s) = i, I(t, v) = j\} \quad (14)$$

where, $(r, s), (t, v) \in N_x \times N_y$; $(t, v) = (r + d, s + d)$; d -distance between two next pixels in a direction θ .

In our work, we will use a set of offsets sweep through 180 degrees ($0^\circ, 45^\circ, 90^\circ$ and 135°) and $d = 1$, obtaining four GLCM matrices. From the co-occurrence matrices, nine Haralick texture descriptors (entropy, energy, contrast, sum average, variance, correlation, maximum probability, inverse difference moment, and cluster tendency) are computed [17]. We use five such summary statistics based on normalized GLCM N_d^θ :

$$\text{Entropy: } E = -\sum_i \sum_j N_d^\theta(i, j) \log_2 N_d^\theta(i, j) \quad (15)$$

$$\text{Energy: } E_n = \sum_i \sum_j (N_d^\theta(i, j))^2 \quad (16)$$

$$\text{Contrast: } C = \sum_i \sum_j (i - j)^2 N_d^\theta(i, j) \quad (17)$$

Inverse difference moment

$$IDM = \sum_i \sum_j \frac{N_d^\theta(i, j)}{1 + |i - j|} \quad (18)$$

$$\text{Correlation } Cor = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d^\theta(i, j)}{\sigma_i \sigma_j} \quad (19)$$

where $\mu_i = \sum_j i \cdot N_d^\theta(i, j)$, $\mu_j = \sum_i j \cdot N_d^\theta(i, j)$,
 $\sigma_i = \sum_j N_d^\theta(i, j)(i - \mu_i)$, $\sigma_j = \sum_i N_d^\theta(i, j)(j - \mu_j)$.

In contrast to work on fire detection [12], texture features are calculated for a gray-scale CFR after applying a local binary pattern operator. Ojala et al. [18] introduced the LBP (local binary pattern) operator in 1996 as a means of summarizing local gray-level structure. It should be noted that the basic LBP features have performed very well in various applications, including texture classification and segmentation, image retrieval and surface inspection [19]. The operator takes a local neighborhood around each pixel, the pixels of the neighborhood at the value

of the central pixel and uses the resulting binary valued image patch as a local image descriptor. It was originally defined for 3×3 neighborhoods, giving 8 bits codes based on the 8 pixels around the central one. LBP code for center pixel p is defined as equation:

$$LBP_p = \sum_{i=0}^7 2^i S(g_i - g_c) \quad (20)$$

where, g_c is gray value of p , g_i is gray value of i pixels.

$S(x)$ is the threshold function that can be defined as equation:

$$S(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

For color features CFR, we use HSV space which allows us to isolate the illumination component of the scene. Hue and Saturation channels are quantized as 20 element normalized histograms for input as features for classification [12].

In the first step, the classifier is trained by using 50 positive and 50 negative flame images. Sample images from the training set are shown in Fig. 3. Calculated texture features (entropy, energy, contrast, inverse difference moment and correlation) after applying a LBP operator and color features in HSV space were used for object classification.

Fig. 3 Examples for (a, b) positive and (c, d) negative flame images.

For objects classification k-nearest neighbor algorithm is used. This algorithm is a method for classifying objects based on closest training examples in the feature space. In the second step, it is performed to assess the quality of training based on test sample images. There are no overlap between training and testing images. Training and testing phases are performed once for classifier configuration.

5. Experimental Results

The developed algorithms were tested on the real cases. Test were run on a PC (AMD Athlon (tm) II X2 245, 2.91 GHz, RAM 2 GB). Our program was implemented using Visual C++ and an open source computer vision library Open CV. The proposed algorithm has been evaluated using data set publicly available at the web address <http://signal.ee.bilkent.edu.tr/VisiFire/Demo/SampleClips.html>, <http://www.openvisor.org> and <http://cvpr.kmu.ac.kr>. Test video sequences contain a smoke, moving people, moving transport, a complex dynamic background, and also a number of video sequences without any smoke/flame. Fig. 4 shows some examples for smoke and flame detection. Detection results for some of the test sequences are presented in Table 1.

For smoke detection processing time of a current frame depends on the block sizes and frequency of changes occurring in a background. If the background is stable and few blocks are detected then processing time decreases. The smoke has been found successfully out on all test video sequences with a smoke.

Therefore in this case, we cannot directly find out a smoke and detection time is increasing. The performed experiments have shown that the algorithm quickly finds out a smoke on a complex dynamic scene. Smoke detection is achieved practically in real time. The processing time per frame is about 15 ms. for frames with sizes of 320 by 240 pixels. The algorithm considers both dynamic and static features of

Fig. 4 Smoke and flame detection in video.

a smoke. The algorithm has a low false alarm level. False alarms on objects with properties similar to a smoke are sometimes possible. Tracing of smoky properties during some frames can solve this problem.

Table 1 Detection results for some of the test video sequences.

Video sequences (Fig. 4)	The processing time per frame (ms)	The flame or smoke was present with / is found with (number of frame)	The number of frames on which there was false alarm / Total of frames
(a)	12.67	10 / 12	0 / 900
(b)	14.79	20 / 112	0 / 244
(c)	14.97	80 / 87	0 / 483
(d)	15.00	30 / 117	0 / 630
(e)	16.35	-	0 / 1073
(f)	14.57	-	0 / 1179
(g)	63.00	1 / 125	0 / 439
(h)	56.00	1 / 223	0 / 1201
(i)	45.00	1 / 123	0 / 411
(k)	48.00	-	0 / 530
(l)	42.00	-	0 / 160
(m)	47.00	-	0 / 171

6. Conclusions

We have presented in this paper algorithms for smoke and flame detection in video sequences. Our algorithms consider dynamic and static features of smoke and flame: contrast, color, texture features and motion. For smoke detection the algorithm uses motion and contrast as the two key features of smoke. Motion is a primary sign and is used at the beginning for extraction from a current frame of candidate areas. In addition to consider a direction of smoke distribution the movement estimation based on the optical flow is applied. For flame detection we use color image segmentation, temporal and spatial wavelet analyses on the first step. After that color and texture features for candidate flame regions are extracted. Texture features are defined based on normalized gray level co-occurrence matrix after computation of local binary pattern. We use Hue and Saturation channels HSV space for color features.

References

- [1] M. Garcia Leonardo, G. Sanchez Perez, M. Nakano, K. Toscano Medina, H. Perez Meana, L. Rojas Cardenas, An early fire detection algorithm using IP cameras, *Sensors* 12 (5) (2012) 5670-5686.
- [2] P. Piccinini, S. Calderara, R. Cucchiara, Reliable smoke detection system in the domains of image energy and color, in: *ICIP*, San Diego, CA, 2008.
- [3] B.U. Toreyin, Y. Dedeoglu, A. E. Cetin, Wavelet based real-time smoke detection in video, *Signal Processing: Image Communication* (20) (2005) 255-256.
- [4] D. Kim, Y. Wang, Smoke Detection in Video, in: *WCCSIE*, Los Angeles, CA, 2009.
- [5] B.U. Toreyin, Y. Dedeoglu, A. Cetin, Contour based smoke detection in video using wavelets, in: *14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, 2006.
- [6] F. Comez Rodriuez, B.C. Arrue, A. Ollero, Smoke monitoring and measurement using image processing, application to forest fires, in: *Proc. of SPIE ATR*, 2003, pp. 404-411.
- [7] D. Kristinić, T. Jakovcevic, D. Stipanicev, Histogram-based smoke segmentation in forest fire detection system, *Information Technology and Control* 38 (3) (2009) 237-244.
- [8] W. Phillips III, M. Shan., V. Lobo, Flame recognition in video, *Pattern Recognition Letters* 23 (1-3) (2002) 319-327.
- [9] B.U. Töreyn, Y. Dedeoglu, U. Gudukbay, A.E. Cetin, Computer vision based method for real-time fire and flame detection, *Pattern Recognition Letters* 27 (1) (2006) 49-58.
- [10] B.U. Toreyin, Y. Dedeoglu, Y. Enis-Çetin, Flame detection in video using hidden markov models, in: *Proceedings of IEEE International Conference on Image Processing*, 2005, pp. 1230-1233.
- [11] Y. Habiboglu, O. Gunay, A.E. Cetin, Flame detection method in video using covariance descriptors, in: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, 2011.
- [12] A. Chenebert, T.P. Breckon, A. Gaszczak, A non-temporal texture driven approach to real-time fire detection, in: *ICIP*, Brussels, Belgium, 2011.

- [13] R.T. Collins, A system for video surveillance and monitoring, in: 8th TMRRS, USA, 1999.
- [14] T. Porter, T. Duff, Compositing digital images, *CG* 18 (3) (1984) 253-259.
- [15] R. Yasmin, Detection of Smoke Propagation Direction Using Color Video Sequences, *International Journal of Soft Computing* 4 (1) (2009) 45-48.
- [16] T. Celik, H. Demirel, H. Ozkaramanli, Automatic fire detection in video sequences, in: 14th European Signal Processing Conference [online], Florence, Italy, Sept. 4-8, 2006, [http://www.eurasip.org/Proceedings/Eusipco/](http://www.eurasip.org/Proceedings/Eusipco/Eusipco2006/papers/1568981622.pdf)
- [17] R. Haralick, Statistical and structural approaches to texture, in: *Proceedings of the IEEE*, 1979, pp. 786-804.
- [18] T. Ojala, M. Pietikainen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition*. 29 (1) (1996) 51-59.
- [19] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (6) (2007) 915-928.