Volume 58, Number 1, 2017

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

STATE OF THE ART: METHODS FOR VIDEO BASED FIRE DETECTION VIA STATIONARY CAMERA

Elena Roxana BUHUS[1], Anca APĂTEAN[2]
[1]*Technical University of Cluj-Napoca, Basis of Electronics Department*
[2]*Technical University of Cluj-Napoca, Communications Department*

Abstract: **Fire outbreak is a common issue happening in open spaces (like forest areas) and the damage caused by these type of incidents is significant toward nature and human interest in matter of economy. This paper provides an analysis and comparison between several implementation methods from the literature meant to detect fire pixels regions from video input. The fundamental goal of this research is to develop an improved algorithm for fire pixel classification in video sequences.**

Keywords: *fire detection, RGB color model, YCbCr color model, HSV, L*a*b*, k-means, image processing.*

I. INTRODUCTION

Uncontrolled fire, flames and smoke are one of the leading hazards affecting everyday life around the world. Fires and burns are one of the most common cause of death, especially to small children. In addition, the entire ecosystem can be affected by such unwanted phenomenon. Smoke detectors have been developed and they have been widely used to an extent to raise alarms. However, some smoke detectors cannot be used reliably to detect fires in indoor, large spaces or outdoor ones.

Several highly dependable fire detection systems use infrared cameras. Although expensive, this type of cameras reached an increased popularity by their own or by combining with the use of standard visible spectrum surveillance cameras.

Computer vision algorithms for automatic video fire or smoke detection have been developed for applications in tunnels, aircraft hangars, ships, etc. being mostly dedicated to small, cluttered spaces. Also, a significant amount of research has been focused on developing reliable video fire detection systems in large or open environment.

Motion and color are two important features for detecting fire and smoke from video sequences. Several related computer-vision based fire detection systems are briefly reviewed in what follows: [1] identifies fire using only color clues; [2] uses RGB color channel information and create fire rules for the three channels separately.

Color and motion clues with fire flickers, later analyzed on the wavelet domain to detect fire is performed in [3] by considering video sequences. Previously, the same authors [4] applied hidden Markov models (HMM) to extract spatial color variations and to improve detection accuracy.

In [5] and [6], the authors used adaptive Gaussian Mixture Model to approximate the background modeling process. In [7] a fire detection rule is used after extracting foreground pixels by means of a change detection algorithm based on the classification of fire/smoke pixels.

In [8] a model where data is represented and analyzed in a YUV space is presented. Some systems may consider that fire flickers with a certain range of frequency, e.g. [9], [10], which performs temporal variations analysis of fire to improve detection performance.

Some authors try to represent shapes of fire regions by Fast Fourier Transform (FFT) of temporal object boundary pixels - in [9]. Also, temporal variation of fire intensity is considered in [10], to select candidate fire regions - represented further by features and they determined the presence of fire or non-fire patterns.

More of the developed systems are constrained by the following conditions: complicated lighting conditions resulting from day and night, artificial lights, light reflection, or shadows and complex scene with objects and/or people moving in velocities and sizes similar to fire. To face these problems, the method presented in [11] proposed a method incorporating both pigmentation values of the RGB color and the saturation and the intensity properties in the HSV color space. They used both spatial and temporal color variations of fire; they also analyze the grouping of fire regions to further eliminate false alarms.

Generally, the methods proposed in literature work with a video stream by extracting a short clip, analyzing it for fire, and then extracting another clip for analysis. If a fire is detected within the clip, an alarm may be further issued.

In section II, we provide an overview of the analyzed methods, while in section III we present some details and the results of the experiments performed on each method. Finally, the conclusion and possible improvements are presented in section IV.

II. ANALYZED METHODS

A number of five methods were analyzed and tested for fire pixel classification. These methods are well explained in [11-15]; the experimental results as concerns their

Manuscript received February 8, 2017; revised March 20, 2017

comparison we provided in section III.
Each method is briefly described in subsections II.1 – II.5. In subsection II.1 we present the approach used in [12], but with a different context: instead of face recognition, fire detection was aimed. In subsection II.2 method from [13] is presented, while subsection II.3 presents the method proposed in [11]. Subsections II.4 and II.5 present methods from [14] and respectively [15]. We have used the standard deviation and mean value presented by Fedias and Saigaa in [12], as an own experiment for fire detection. Using this approach, we defined the experiments, to measure accuracy in fire detection pixels for RGB fire color model approach, HSV, YCbCr and L*a*b as presented by the corresponding authors in their works [11-15].

II.1 Standard deviation and mean value for fire

Generally, statistical classifiers such as mean value and standard deviation are used for face recognition. The generic vector of characteristics or the feature vector, as it is generally called in literature, is created from images that contain the face desired to be detected [12]. Once this vector is extracted, the next step is to compare it with a trained vector and obtain the minimum distance vector between the current image and the trained data base.

According to an error rate, it is decided if the face from the current image is detected or not. This approach, which is based on a classifier and an extracted features vector was adapted for a set of images that contained fire and smoke regions in open and closed spaces, and the detection performance was measured. In order to train the model, a data base of 50 images was considered from a stationary camera, having mostly positive scenarios but also few negative ones (meaning that no fire/smoke was present). At block level, the method is described in Figure 1.

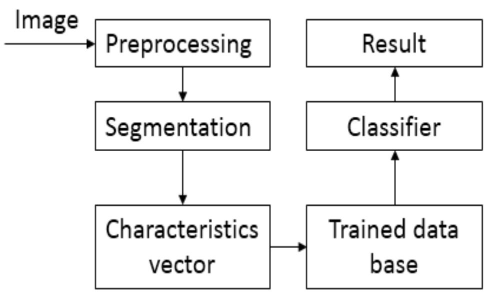

Figure 1. Model diagram for statistical object detection

In the preprocessing stage, the image is filtered in order to remove noise before applying other processing steps, using the gray version of the image. For image segmentation and features vector extraction, the mean value and standard deviation were used. For the mean value, as presented in [12], the following equation was used:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \qquad (1)$$

Where x_i is a variable vector and n represents the number of scalar observations.
The standard deviation is the square root of the variance according to [12] with the below formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}} \qquad (2)$$

where μ is the mean value computed previously.

If we consider the image as a matrix, where every row and column has a numerical value, then the features vector will be a combination between the mean value and the standard deviation (for every row and column in the image). The aproach considers that after extracting the features of an object and comparing with the ones corresponding to the others from the trained data set, using a minimum distance threshold T determined experimentally, the sistem should identify the images that contain the objects of interest. In our case fire pixel regions.

II.2 The RGB fire color model

One of the most used methods for fire pixel detection is based on the RGB fire color model. There are three components that determine the color of a pixel: R(red), G(green) and B(blue). Each of these components are extracted from the frame.

Using the fire color model, fire pixels are identified. In this sense, according to method [13], it is considered that a pixel is a fire pixel if the values for each component respect the relations: R>G>B. For fire detection, the R component is considered the dominant one; in this sense, it is imposed that R should have a determined threshold RTH. Thus, a mask can be created and it may be used to mark the fire pixels in an image using the following rules, as presented in Table I:

Table 1. Rules for fire pixel detection in RGB space

R1	R1(x,y) = (R(x,y) > G(x,y)) && (G(x,y) > B(x,y))
R2	R2(x,y) = (R(x,y) > 190) && (G(x,y) > 100) && (B(x,y) <140)

The resulting mask is M = R1 && R2 and contains all the pixels from the image that presents the fire characteristics. The performances of this approach are shown in section III.

II.3 RGB and HSV fire color model

Another method proposed by [11], take into consideration the flicker of the flame, as part of the fire detection process. In this sense, the method is defined by three stages, as presented in Table II.

Table 2. Stage description for method from [11]

Stage	Description
1.Detect Flicker	Detect pixel regions with a high flicker frequency by luminance time derivate matrix
2.Segment Color	pixel regions with fire characteristics are identified using RGB and HSV fire color model over the color cumulative matrix
3.Merge Regions	the previous identified pixel regions are merged
4. Measure Spatial color variation	on the identified regions, spatial color variation is measured to filter out objects with similar characteristics as fire
5.Measure Time color variation	on the identified regions time color variation is measured to filter out objects with similar characteristics as fire, but a constant time color variation
6. Decide	the result is provided as marked: positive or

final verdict	negative detection

For flicker detection, two main properties of the fire are used, meaning that first, the flickering frequency is between 1 and 10 Hertz and second, the luminance of the fire tends to be maximum [11]. Briefly described, the flicker detection algorithm has several steps, as presented in Table III.

The color segmentation stage has also several processing steps that aim to provide another mask, a matrix that can be used to the next stage. For the region merging stage, a series of morphological operations are applied upon the previous obtained *Fire* matrix [11].

During the next stages of the method, spatial and time color variation are measured on the merged matrix to filter out objects that have similar color characteristic as fire. A detailed description of these stages can be consulted in [11]. In the last stage, according to the identified fire pixel rate, it is decided if there is fire in the current frame or not.

Table 3. Flicker detection algorithm details as shown in [11]

Step	Description
1.Transform RGB to YUV	$Yi(x,y)$ is the luminance of pixel (x,y) from frame Fi.
2. Compute luminance derivate	$D_i(x,y) = \lvert Y_i(x,y) - Y_{i-1}(x,y)\rvert$
3. Compute cumulative time derivate	$CT_i(x,y) = \alpha CT_{i-1}(x,y)$ $+ (1-\alpha) w_i(x,y) D_i$ where: α means the cumulative power set to (N-1)/N, frames from a video, while wi means the weight of luminance. $CT1(x,y)$ is set to 0, initially.
4.Normalize $CT(x,y)$	$CT(x,y)$ is normalized to 8 bit values and the mean value Ave is calculated
5.Create Mask	A mask that contains all the pixels from normalized $CT(x,y) > Ave$ is created

II.4 The RGB and YCbCr fire color model

Within [14], authors proposed a fire pixel detection using the RGB but also the YCbCr fire color model. In this sense, the components in the YCbCr space, are Y for luminance (luma), Cb for blue minus luma (B-Y) while Cr is red minus luma (R-Y). As processing steps, this method is trying to reach a fast and efficient implementation: the image is captured via a stationary camera; next, within the preprocessing stage, filters are used for noise reduction together with resizing of the current capture, for a further faster processing.

The fire color segmentation algorithm is presented in Table IV. In the segmentation stage, the mathematical equations are used to create the possible fire pixel mask. The potential fire pixel mask is used in the classification stage to extract the detected fire blob (in the current captured image) and these steps are applied for each frame. One important aspect for a detection algorithm functioning in real time is the processing speed. Depending on the applications need and context constraints, this may vary within different acceptable margins.

The method is implemented considering the bloc schematic presented in Figure 2, using block level representation, in a sequential manner: each stage provides processed information as a matrix; this is further processed, until it reaches the final scope, which is classification in one of the possible classes, meaning objects that have fire pixel characteristics.

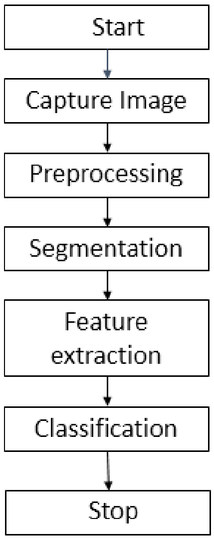

Figure 2.RGB and YCbCr color model fire detection

According to the experimental results obtained in [14], the values for Rmed, Gmed, Bmed and Th have been settled to 190,100,140 and 70 respectively.

Table 4. Fire segmentation using RGB/YCbCr color model

Color space	Fire detections rules
RGB	1. $R > G > B$
	2. $R > Rmed \cap G > Gmed \cap B < Bmed$
YCbCr	3. $Y(x,y) \geq CB(x,y)$
	4. $CR(x,y) \geq CB(x,y)$
	5. $Y(x,y) \geq Ymed \cap$ $CB(x,y) \leq CBmed \cap CR(x,y) \geq Crmed$ Where Ymed, CBmed, CRmed are: $Ymed(x,y) = \frac{1}{MxN}\sum_{x=1}^{M}\sum_{y=1}^{N} Y(x,y)$ $CBmed(x,y) = \frac{1}{MxN}\sum_{x=1}^{M}\sum_{y=1}^{N} CB(x,y)$ $CRmed(x,y) = \frac{1}{MxN}\sum_{x=1}^{M}\sum_{y=1}^{N} CR(x,y)$
	6. $CB(x,y) - CR(x,y) \geq Th$
	7. $CB(x,y) \leq 120 \cap CR(x,y) \geq 150$

In order to classify the fire pixel, the rules presented in Table IV were applied to the current frame to create a mask marking the fire pixels.

II.5 The L*a*b and YCbCr fire color model

One of the most complex color space specified by International Commission on Illumination (CIE) is considered to be L*a*b. The L*a*b color space provides all the colors visible to the human eye and it is independent of any device, this being the reason why it is used as a reference. Within research presented in [15], fire detection method uses the L*a*b and YCbCr color space, together with the k-mean clustering method for best results. Authors claim this implementation is very stable and accurate with great results in real time detection. Since the L*a*b color space contains colors that cannot be perceived by the human eye as in the CIE XYZ color space, it is necessary to convert from RGB to CIE XYZ and from CIE XYZ to L*a*b. The block level description for implementing the method from [15] is presented in Figure 3.

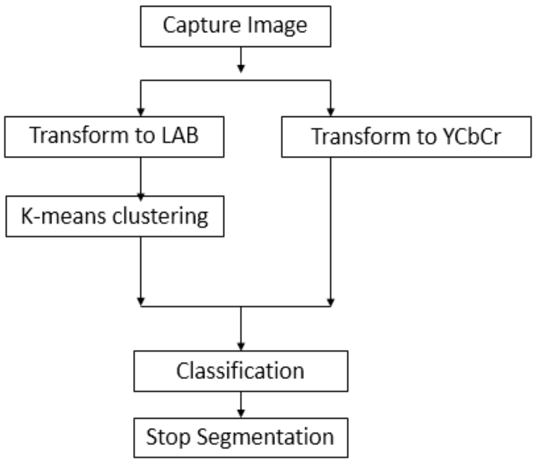

Figure 3. L*a*b and *YCbCr color model fire detection*

The captured image is provided in parallel for two processing paths, each providing input for the classification stage. The color segmentation for YCbCr uses the normal bivariate distribution function, having as variables the values for Cb and Cr of each pixel. In this sense, the formula for implementing the algorithm from [15] for the function considering the YCbCr color space is:

$$f(Cr,Cb) = \frac{1}{2\pi S_{Cr} S_{Cb}} * \exp\left\{\frac{1}{2}\left[\left(\frac{Cr-Crm}{S_{Cr}}\right)^2 - 2\frac{(Cr-Crm)(Cb-Cbm)}{S_{Cr}S_{Cb}} + \left(\frac{Cb-Cbm}{S_{Cb}}\right)^2\right]\right. \quad (3)$$

The Crm, Cbm, SCr and SCb represent the simple variance and standard deviation corresponding to red and blue channels. According to the method from [15], if f(Cr,Cb) > T, then we have a fire pixel. Statistical analysis showed that the values for Crm, Cbm and T are 0.246431, 0.682332 and 3*exp(-5) as suggested in [15].

The k-means clustering algorithm is implemented by several steps, as presented in Table V and implies the use as input of the outcome from the transform to L*a*b color space. L*a*b color space is a 3-axis color system with dimension L for lightness, while a and b are for the color dimensions. Working with the L*a*b color space includes all of colors from the visible spectrum, as well as colors outside of human perception. This color space is the most exact means of representing color; in addition, it is device independent as stated in [15].

Table 5. The K-means clustering algorithm

Step	Description
1	Transform to L*a*b color space, using a threshold value T; if the distance between the old and the new mean < then the center -> consider it stable
2	Compute cluster center: K pixels from the image are initially considered the center of the cluster
3	For each pixel it is computed the distance between the pixel and the mean value of the cluster (allocating to the cluster only the pixel with the minimum distance)
4	Compute the new mean value for each cluster using the formula: $\bar{v} = \frac{\sum_{i=1}^{N} v_i}{N}$ Where v_i is a variable vector and N represents the number of scalar observations.
5	Compare the distance between the new and the old mean value; if the distance is > T, then repeat step 3 and 4, otherwise continue with step 6
6	Check each cluster if the mean value is not different than the value of the pixels belonging to the color space, then those pixels from the cluster are considered fire pixels

III. EXPERIMENTAL RESULTS

In order to measure the performance of the methods presented in this article, we used a data base of 39 images, from which 34 ones are for open space (source from internet), such as forests, while the other five were provided by the main author. An example of results, for positive scenarios, for testing the method from [14] is represented in Figure 4. For a positive scenario, in which the fire region is present, a reliable method should mark the identified fire region. Still, due to the error rate, it seems that for a closed-door scenario (third image from Figure 4), the method from [11] misses the present fire region from the lighter.

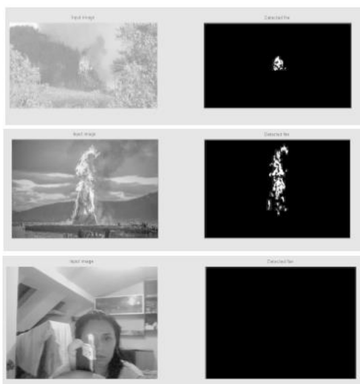

Figure 4. Results method [14] for scenarios with fire present

Other example of the results for negative scenarios are presented in Figure 5. For a negative scenario, in which the fire region is absent, a reliable method should mark the absence of the fire region.

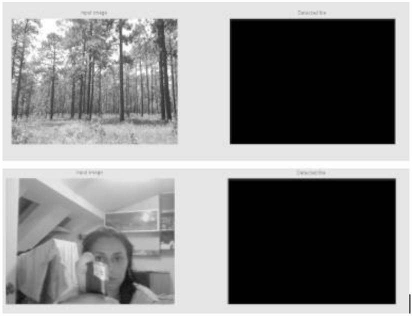

Figure 5. Results for the method from [14] for scenarios with fire absent

The used images for measurements were numbered from F1 until F39, however in the graphical representation for a better overview only the i-th index of the Fi image is used.

Both scenarios of fire present or absent was covered by the data base. This process of validation uses a golden model as suggested in [16], provided by the human decision, that is compared with the output of the method. The notion of True Positive (TP) and True Negative (TN) means a correct classification.

A False Positive (FP) means that the method provided an erroneous classification, and that the object of interest is present in the image but was not identified, while a False Negative (FN) means that the object was identified in the image although it was absent.

The accuracy of a method is defined as the ability to detect the region of interest (ROI) according with the specifications. At this point the accuracy of the methods is a more important aspect versus efficiency and response time. In this sense, the accuracy was defined as: accuracy = TP/(TP+TN)*100%.

In Table VI the results for each presented method in Section II are presented, numbered from M1 to M5 and according to each subsection number.

For the golden model provided by the human decision, if the region of interest in present in the image we marked it with ROI-P equal to 1, while if the region of interest is absent we marked it with ROI-A equal to 0.

In Figure 6, we represented the golden model for the mentioned data base.

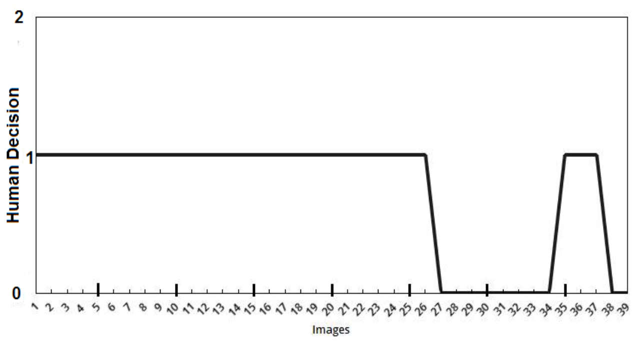

Figure 6. Human decision – golden model for the data base

The results for each method from M1-M5, was marked with 0 for FP and 1 for TP. In this sense, Figure 7 provides a graphical representation of the results.

Considering the provided results, together with the provided formula for accuracy, it was able to establish the accuracy of each method, as presented in Table VI.

Table 6. Accuracy results for methods M1-M5

Method	**Accuracy**
M1	74%
M2	82%
M3	94%
M4	89%
M5	87%

Considering our experimental results, together with the obtained results of the authors that first proposed methods M1-M5, one can notice that method M3-M5 would offer a satisfying accuracy on one hand. On the other hand, not all of them ensure a small error rate.

It seems that method M3 might provide a small error rate due to the complexity of the implemented algorithm. In this sense, the response time is larger as compared with other methods, but at this point accuracy is much important.

Based on the results highlighted in this section, we can conclude that a reliable fire detection method, providing a small error rate implies a complex implementation and multiple aspects to consider. Also, one of the authors goal is to create a proper method to detect both fire regions and smoke regions.

One possible approach is to perform a further analysis of the research presented in [17] and [18], to evaluate the performances of each implementation and to create a fusion method that would identify both fire and smoke, from the same video sequence.

Authors have already approached this domain, i.e. that of fusion performed at multiple different levels, as presented in [19]. Pixel level, features level, classifier or even deeper, at that of classifier kernel (as for the SVM) but also decision-level fusion is aimed. At the classifier level, the SVM is also intended to be tested, as already approached by the authors in [20] and [21].

IV. CONCLUSION

In this paper, five methods were analyzed for performing fire pixel detection.

We provided details regarding the implementation algorithms, together with few results of their application on a data base created from 39 images.

Also, a new approach was tested in the context of fire detection, meaning the mean value and standard deviation that is commonly used for face recognition. The reasoning for this choice was due to the low complexity in the algorithm implementation.

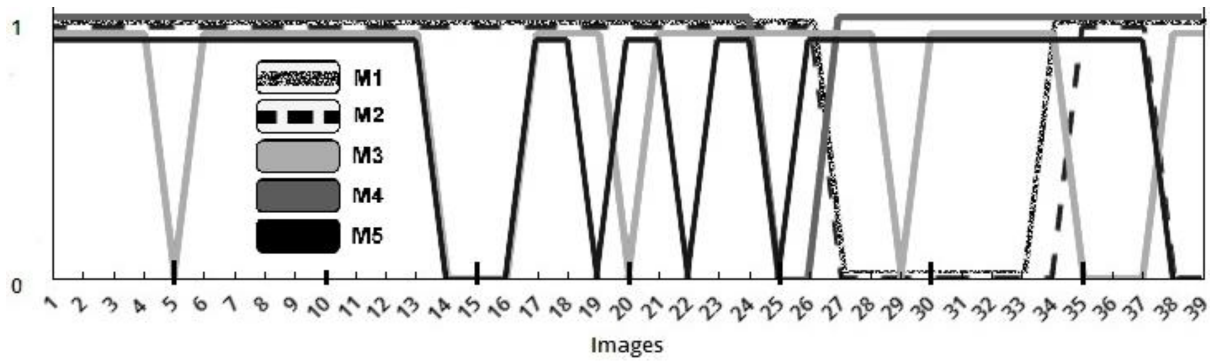

Figure 7. Graphical representation of results for M1-M5

REFERENCES

[1] Healey, G., Slater, D., Lin, T., Drda, B., and Goedeke, D., 1993, A System for Real-Time Fire Detection, Computer Vision and Pattern Recognition, 605-606.
[2] T-H. Chen, Ping-Hsueh Wu, and Yung-Chuen Chiou, "An early fire-detection method based on image processing," in Image Processing, 2004. ICIP '04. 2004 International Conference on, oct. 2004, vol. 3, pp. 1707 – 1710 Vol. 3.
[3] Töreyin, B. U., Dedeoğlu, Y., Gudukbay, U., Çetin, A. E., 2006, Computer Vision Based Method for Real-Time Fire and Flame Detection, Pattern Recognition Letters, 27, 49-58.
[4] Töreyin, B. U., Dedeoğlu, Y., Çetin, A. E., 2005, Computer Vision Based Method for Real-Time Fire and Flame Detection. Bilkent University.
[5] P. Piccinini, S. Calderara, R. Cucchiara. Reliable smoke detection system in the domains of image energy and color. 15th International Conf. on Image Processing, (2008), pp.1376–1379.
[6] B.Ugur Toreyin et al. Wavelet based real-time smoke detection in video.Signal Processing: Image Communication, EURASIP, Elsevier (20) (2005), pp. 255–256.
[7] T. Celik, Hasan Demirel, Huseyin Ozkaramanli, and Mustafa Uyguroglu, "Fire detection using statistical color model in video sequences," Journal of Vis. Comun. Image Represent, vol. 18, pp. 176–185, April 2007.
[8] G. Marbach, Markus Loepfe, and Thomas Brupbacher," An image processing technique for fire detection in video images," Fire Safety Journal, vol. 41, no. 4, pp. 285 – 289, 2006.
[9] Liu, C., Ahuja, N., 2004, Vision Based Fire Detection, Proc. of the 17th International Conf. on Pattern Recognition, 134-137.
[10] Marbach, G., Loepfe, M., Brupbacher, T., 2006, An Image Processing Technique for Fire Detection in Video Images, Fire Safety Journal, 4, 285-289.
[11] Xiaojun Qi, Jessica Ebert, "A Computer Vision-Based Method for Fire Detection in Color Videos", International Journal of Imaging, Volume 2 – 2009
[12] M.Fedias, D. Saigaa, "A new approach based in mean and standard deviation for authentication system of face", International Review on Computers and Software, Vol. 5, n.3 -2010
[13] Punam Patel, Shamik Tiwari, "Flame Detection using Image Processing Techniques", International Journal of Computer Applications, Volume 58 – November 2010
[14] Norsyahirah Izzati binti Zaidi, Nor Anis Aneza binti Lokman, Mohd Razali bin Daud, Hendiyawan Achmad, Khor Ai Chia, "Fire Recognicion using RGB and YCbCr Color Space", ARPN Journal of Engineering and Applied Sciences, Volume 10 – 2015
[15] Rui Chen, Yuanyuan Luo, Mohanmad Reza Alsharif, "Forest Fire Detection Algorithm Based on Digital Image", Journal of Software, Volume 8, Number 8 , Aug. 2013
[16] Kumarguru Poobalan, Siau-Chuin Liew, "Fire Detection Algorithm using image processing techniques", Proceeding of the 3rd International Conference on Artificial Intelligence and Computer Science (AICS2015), 12 - 13 October 2015, Penang, MALAYSIA
[17] Yashar Deldjoo, Fatemef Nazary, Ali M. Fotoului, "A Novel Fuzzy – Based Smoke Detection System Using Dynamic and Static Smoke Features", Iranian Conference on Electrical Engineering, Volume 23, 2015
[18] N.Brovko, R. Bogush, S.Ablameyko, "Smoke detection algorithm for intelligent video surveillance system", Computer Science Journal of Moldova Volume 21, 2013
[19] Apatean,A. , Rogozan,A., Bensrhair,A., "Information Fusion for Obstacle Recognition in Visible and Infrared Images", IEEE International Symposium on Signal, Circuits and Systems (ISSCS2009), July, 9-10, Iasi, Romania, pp. 1-4, ISBN: 978-1-4244-3785-6, 2009
[20] Apatean (Discant), A., Rogozan, A., Bensrhair, A., "Kernel and Feature Selection for Visible and Infrared based Obstacle Recognition", 11th Int. IEEE Conf. on Intelligent Transportation Systems, ITSC2008, pp. 1130-1135, 12-15 Oct. 2008
[21] Apatean, A., Rogozan, A., Bensrhair, A., "SVM-based Obstacle Classification in Visible and Infrared Images", Proceedings of the 17th European Signal Processing Conference (EUSIPCO2009), Glasgow, Scotland, 2009