Available online at www.sciencedirect.com
SCIENCE DIRECT®
Fire Safety Journal 41 (2006) 285–289
FIRE SAFETY JOURNAL
www.elsevier.com/locate/firesaf

An image processing technique for fire detection in video images

Giuseppe Marbach*, Markus Loepfe, Thomas Brupbacher

Siemens Switzerland Ltd, CH-8708 Männedorf, Switzerland

Available online 29 March 2006

Abstract

This paper presents an image processing technique for automatic real time fire detection in video images. The underlying algorithm is based on the temporal variation of fire intensity captured by a visual image sensor. The full image sequences are analyzed to select a candidate flame region. Characteristic fire features are extracted from the candidate region and combined to determine the presence of fire or non-fire patterns. Fire alarm is triggered if the fire pattern persists over a period of time.

Keywords: Video fire detection; Real time image processing; Pattern recognition

1. Introduction

The progress on video surveillance and monitoring equipment technology in the last decade has increased the presence of Closed Circuit Television (CCTV) cameras in many public and private areas. Their presence has opened the CCTV market to the opportunity to perform, besides monitoring and storage, automatic event detection, e.g. real time video fire detection.

Image processing algorithms for automatic video fire or smoke detection have been developed in the past for applications in tunnels, aircraft hangars, fighting ships, etc. [1–3]. But none of the algorithms presented in the past are so robust and flexible as to face all of the problems typical in the CCTV automatic video fire detection. These problems are:

- Lighting conditions (day and night, artificial lights, light reflections, shadows).
- Image quality (poor camera resolution, poor camera contrast, poor signal transmission, dirty lens, vandalism affecting the image quality).
- Scene complexity (moving objects and people: different velocities and sizes).
- Processor performance (real time detection, processor speed and memory).
- System installation (friendly configuration and parametrisation).

Great flexibility and high reliability are required from the fire detection algorithms to reduce the false alarm rate and to decrease the alarm reaction time. Moreover the detection algorithms must not disturb the performance or reduce the quality of the monitoring and storage task.

The method presented in this paper uses the temporal accumulation of time derivative images to extract the *best candidate fire region*. The temporal accumulation and the candidate fire region are described in Section 2. The subsequent analysis for the detection of fire is evaluated with the data of the best candidate fire region. Characteristic fire features are extracted from the image data of this region, as described in Section 3. These features are used to compute the *fire indicator*, see Section 4, whose pattern describes the presence of fire or non-fire in the video sequence. A fire alarm is triggered if this fire pattern persists for a critical time. In Section 5, the sensitivity parameters of the algorithm are introduced and their effect on the false alarm rate and alarm reaction time are explained. Finally, in Section 6, the response of the algorithm towards a series of tests in different environments is discussed.

*Corresponding author.
E-mail address: giuseppe.marbach@siemens.com (G. Marbach).

0379-7112/$ - see front matter
doi:10.1016/j.firesaf.2006.02.001

2. Candidate fire region

Fire has the property to flicker, increasing and decreasing the intensity of the emitted light. From the point of view of a camera, this flicker causes an increase and decrease in the luminance of the video images. The typical fire flicker frequency is in the 1–10 Hz range [4]. Moreover, fire is, typically, the strongest source of light, thus the luminance of the pixels near the fire tends toward the maximal value allowed by the camera, reaching in most cases the saturation level. These two properties of the fire, flickering and reaching maximal luminance, are used to model the algorithm presented in this paper.

The "YUV" representation of the video data is assumed here. The luminance component is represented by $Y_{ik}(t)$ and the chrominance, i.e. the colour information, by its two components $U_{ik}(t)$ and $V_{ik}(t)$, where t is the time and the indices i, k are the horizontal and vertical pixel position.

The time derivative of the luminance $Y_{ik}(t)$ is zero for the stationary scene regions, and is non-zero for moving objects. Thus the time derivative of the video images will track a moving object. The sum of the absolute value of the derivatives increases if the object moves periodically around a region. In case of a fire scene, the property of the fire to flicker increases permanently the pixels value near the fire region. This sum of derivatives is represented by:

$$M_{ik} = \sum_t D_{ik}(t), \quad t_0 \leqslant t < t_n,$$

where $[t_0,\ t_n)$ is the discrete summation interval and $D_{ik}(t) = |\partial_t Y_{ik}(t)|$ is the absolute value of the luminance time derivative, e.g. the discrete approximation is:

$$D_{ik}(t) = |Y_{ik}(t) - Y_{ik}(t-1)|.$$

A more efficient and robust way to express the sum $M_{ik}(t)$ is to introduce the *cumulative time derivative matrix* $A_{ik}(t)$, expressed by the recursive formula

$$A_{ik}(t) = \alpha A_{ik}(t-1) + (1-\alpha) D_{ik}(t),$$

where α represents the cumulative strength, $0 \leqslant \alpha \leqslant 1$. The matrix $A_{ik}(t)$ describes approximately the mean of the time derivatives $D_{ik}(t)$ in the time interval $[t-N, t]$ with $N+1 = N/\alpha$. Contrary to the sum M_{ik}, the cumulative time derivative matrix $A_{ik}(t)$ has the advantage of being recursive and the values of $A_{ik}(t)$ increase or decrease exponentially according to the cumulative strength α, converging to finite values:

$$\min\{D_{ik}(t)\} \leqslant A_{ik}(t) \leqslant \max\{D_{ik}(t)\}.$$

Fig. 1(b) shows a typical cumulative time derivative matrix $A_{ik}(t)$ of a fire video sequence. The values of $A_{ik}(t)$ are scaled to fit an 8 bits luminance image.

In the fire scene, the cumulative matrix $A_{ik}(t)$ will have high values at the borders of the flame region and otherwise values nearly zero. For persistent fire, the values of $A_{ik}(t)$ near the flame region converge as a geometric series to the temporal mean of $D_{ik}(t)$. In the center of the fire, most luminance pixels are saturated, their time derivative is zero, and they do not contribute to $A_{ik}(t)$.

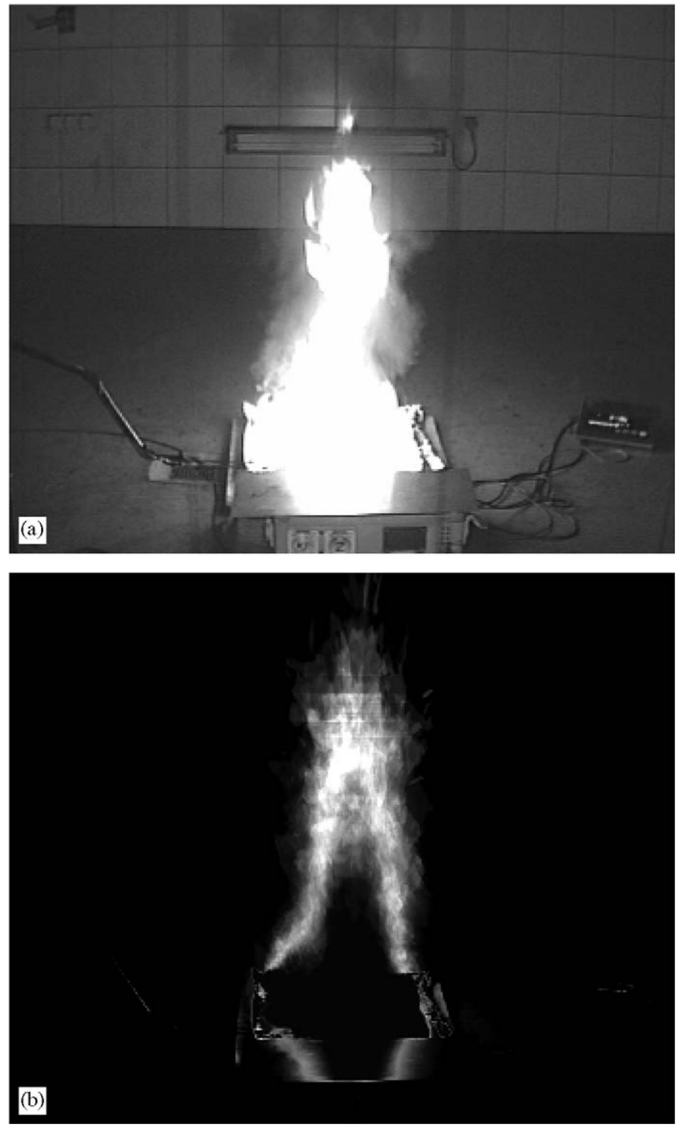

Fig. 1. Typical fire video image (a) and its cumulative matrix (b).

In order to improve robustness of the cumulative time derivative matrix towards false alarms, the time derivative $D_{ik}(t)$ is multiplied by a weight matrix $W_{ik}(t)$:

$$A_{ik}(t) = \alpha A_{ik}(t-1) + (1-\alpha) D_{ik}(t) W_{ik}(t).$$

To enhance the property of the fire of having pixels with maximal luminance and to suppress those having low luminance, the weight matrix $W_{ik}(t)$ is chosen to be proportional to the luminance. This condition is expressed by

$$\text{if } Y_{ik}(t) \geqslant \delta \quad \text{then } W_{ik}(t) = Y_{ik}(t), \quad \text{else } W_{ik}(t) = 0,$$

where the luminance threshold $\delta(\lambda_1,\lambda_2)$ depends from two empirical constants λ_1 and λ_2. These empirical constants ensure that the threshold δ is always between the maximal fire luminance and the scene mean luminance, suppressing most of the non-fire pixels of the scene.

Pixels in the cumulative time derivative matrix $A_{ik}(t)$ with high value represent pixels with high probability of being fire pixels. Thus, to extract the *best candidate fire region* Ω_{ROI}, the pixel with maximal value in the cumulative matrix is chosen:

$$(i_{\mathrm{ROI}}, k_{\mathrm{ROI}}) = \{(i,k) | \max\{A_{ik}(t)\}\}.$$

The region Ω_{ROI}, is defined in the algorithm implementation as a 32×32 pixels neighborhood of $(i_{\mathrm{ROI}}, k_{\mathrm{ROI}})$.

The method presented here uses only one Ω_{ROI} region to detect the presence of fire. Obviously, it is possible to choose more than one non-overlapping candidate fire region.

3. Features extraction

In the second part of the detection, the analysis is focused only on the candidate fire region Ω_{ROI}: characteristic fire features are extracted and combined to evaluate the presence of fire or non-fire patterns.

To extract the six features used for fire detection, the notion of the *active pixels* of the Ω_{ROI} region is introduced: it is defined as the pixels (i,k) of Ω_{ROI} whose values $A_{ik}(t)$ are greater than or equal to a threshold η_1, where $0 \leqslant \eta_1 \leqslant 255$ for $A_{ik}(t)$ quantized to 8 bits. The set of active pixels of Ω_{ROI} is thus

$$\pi_{\mathrm{ROI}} = \{(i,k) \in \Omega_{\mathrm{ROI}} | A_{ik}(t) \geqslant \eta_1\}.$$

The luminance of the active pixels in the Ω_{ROI} region provides the basis for the extraction of three main features:

The luminance of the active pixels: $I_{\mathrm{ROI}}(t)$.
The frequency of $I_{\mathrm{ROI}}(t)$: $f_{\mathrm{ROI}}(t)$.
The amplitude of $I_{\mathrm{ROI}}(t)$: $a_{\mathrm{ROI}}(t)$.

The luminance of the active pixels is thus

$$I_{\mathrm{ROI}}(t) = \{\mathrm{mean}\{Y_{ik}(t)\} | (i,k) \in \pi_{\mathrm{ROI}}\}.$$

The features $f_{\mathrm{ROI}}(t)$ and $a_{\mathrm{ROI}}(t)$ are estimated analyzing the luminance curve $I_{\mathrm{ROI}}(t)$ over the time t.

The second set of three features is related to the numbers of pixels:

The number of active pixels: $r_{\mathrm{ROI}}(t)$.
The number of saturated pixels: $s_{\mathrm{ROI}}(t)$.
The number of fire-color pixels: $c_{\mathrm{ROI}}(t)$.

The number of active pixels $r_{\mathrm{ROI}}(t)$ is defined as the size of the set π_{ROI}:

$$r_{\mathrm{ROI}}(t) = \|\pi_{\mathrm{ROI}}\| = \{\text{number of } (i,k) \in \Omega_{\mathrm{ROI}} | A_{ik}(t) \geqslant \eta_1\}.$$

Thus $r_{\mathrm{ROI}}(t)$ is in the range $0 < r_{\mathrm{ROI}}(t) < N_{\mathrm{ROI}}$, where N_{ROI} is the total number of pixels in the region Ω_{ROI}. The feature $r_{\mathrm{ROI}}(t)$ acts as a measure of the number of pixels in Ω_{ROI} that fulfill the fire properties described by the cumulative time derivative matrix $A_{ik}(t)$.

The second feature of this group, $s_{\mathrm{ROI}}(t)$, represents the number of luminance pixels that are saturated, i.e. pixels whose value is bigger than or equal to a threshold η_2:

$$s_{\mathrm{ROI}}(t) = \{\text{number of} (i,k) \in \Omega_{\mathrm{ROI}} | Y_{ik}(t) \geqslant \eta_2\},$$

where $\eta_2 < 255$ and $\eta_2 \geqslant 0$. This feature acts as a measure of the number of pixels in Ω_{ROI} that fulfill the condition that fire pixels have maximal luminance value.

The feature $c_{\mathrm{ROI}}(t)$ is defined as the number of chrominance pixels in the region Ω_{ROI} falling in a chrominance sector Ω_{c} divided by all the *active chrominance pixels* in Ω_{ROI}:

$$c_{\mathrm{ROI}}(t) = \{\text{number of} (i,k) \in \Omega_{\mathrm{ROI}} | (V_{ik}(t), U_{ik}(t)) \in \Omega_{\mathrm{c}} \text{ and } \|(V_{ik}(t), U_{ik}(t))\| \geqslant \varepsilon \text{ and } A_{ik}(t) \geqslant \eta_1\} / \rho(t),$$

where (U, V) represents the chrominance vector, and $\rho(t)$ is the number of active chrominance pixels in the region Ω_{ROI}:

$$\rho(t) = \{\text{number of } (i,k) \in \Omega_{\mathrm{ROI}} | \|(V_{ik}(t), U_{ik}(t))\| \geqslant \varepsilon \text{ and } A_{ik}(t) \geqslant \eta_1\},$$

where ε is typically $0 < \varepsilon < 20$. The first inequality in $\rho(t)$, $\|(V_{ik}(t), U_{ik}(t))\| \geqslant \varepsilon$, ensures that the pixels near the grey scale are eliminated. The second inequality ensures that chrominance pixels that are not active pixels are not considered. This feature measures the presence of fire pixels in Ω_{ROI} using the color propriety of fire. For monochrome cameras and poor color cameras, this feature can be switched off, as described in Section 5, or its contribution to the final fire pattern can be attenuated.

The chrominance sector Ω_{c} is in the algorithm implementation represented by two lines, dividing the chrominance space in two sectors, see Fig. 2. The fire sector is

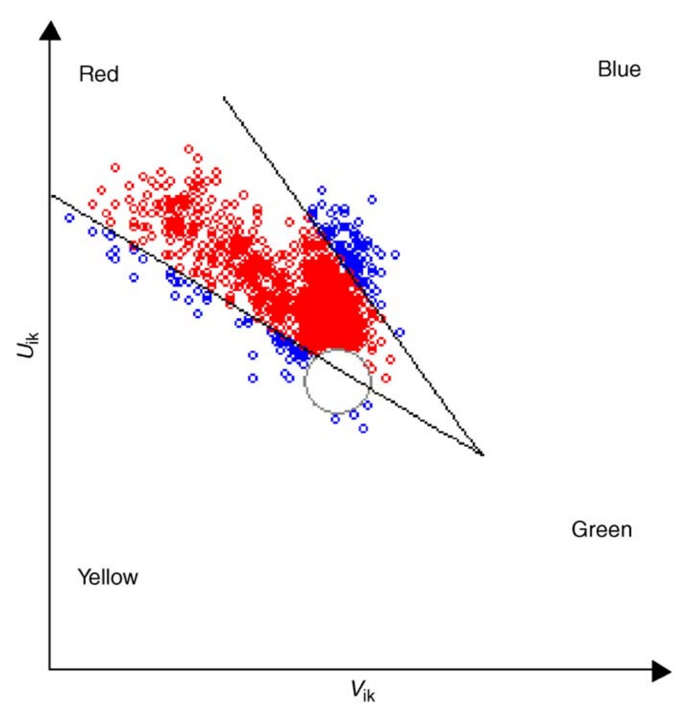

Fig. 2. Qualitative representation of the chrominance sector Ω_{c} and the chrominance pixels ($V_{ik}(t)$, $U_{ik}(t)$) for a typical fire video sequence.

around the color red and is chosen wide enough, so that chrominance analysis for poor quality color cameras or cameras with slight color shift can be still done.

In order to have robust and continuous feature values, each feature is accumulated over the time using the efficient recursion method:

$$\underline{x}_{\mathrm{ROI}}(t) = \beta \underline{x}_{\mathrm{ROI}}(t-1) + (1-\beta) x_{\mathrm{ROI}}(t),$$

where $x_{\mathrm{ROI}}(t)$ is the feature value, e.g. the active pixels feature $r_{\mathrm{ROI}}(t)$. β is the accumulation strength and $\underline{x}_{\mathrm{ROI}}(t)$ is the resulting *cumulative feature* value. The strength β is chosen so that it reproduces a time integration over 1 s, i.e. $\beta = 0.96$ for a video frame rate of 25 fps.

4. Fire pattern

Each of the six features, $\underline{x}_{\mathrm{ROI}}(t)$, is associated with an indicator, $I(\underline{x},t)$, as follows:

$$I(\underline{x}, t) = 1 \quad \text{if } \mu_{\mathrm{low}}(\underline{x}) \geqslant \underline{x}_{\mathrm{ROI}}(t) \geqslant \mu_{\mathrm{high}}(\underline{x})$$

$$I(\underline{x}, t) = 0 \quad \text{else.}$$

The values of the thresholds $\mu_{\mathrm{low}}(x)$ and $\mu_{\mathrm{high}}(x)$ are determined empirically. All these indicators are combined together to build the *fire indicator* $I_{\mathrm{F}}(t)$, which describes the presence of fire or non-fire. An intuitive and easy way to represent $I_{\mathrm{F}}(t)$ is to multiply all the six indicators $\underline{x}_{\mathrm{ROI}}(t)$:

$$I_{\mathrm{F}}(t) = I(\underline{l}, t) I(\underline{f}, t) I(\underline{a}, t) I(\underline{r}, t) I(\underline{s}, t) I(\underline{c}, t).$$

Obviously, more sophisticated representations of $I_{\mathrm{F}}(t)$ are possible, e.g. the combination of the features $\underline{x}_{\mathrm{ROI}}(t)$ to a neural network.

The fire pattern is recognized if the fire indicator $I_{\mathrm{F}}(t)$ is equal to 1. A fire alarm is triggered if this fire pattern persists for a critical time. This is expressed by the integrator $Q(t)$, which increments or decrements according to the value of $I_{\mathrm{F}}(t)$:

$$Q(t) = Q(t-1) + v(2I_{\mathrm{F}}(t) - 1),$$

where v represents the decrement and increment strength. The integrator $Q(t)$ increases by a factor v if $I_{\mathrm{F}}(t)$ is equal to 1, else it decreases.

In order to prevent an endless decrement or increment, $Q(t)$ is saturated to 0 on the bottom and by the threshold Q_{T} on the top. Alarm is then triggered if the integrator $Q(t)$ reaches the threshold $Q_0 : Q(t) \geqslant Q_0 \Rightarrow$ fire alarm, where $0 \leqslant Q_0 \leqslant Q_{\mathrm{T}}$. Fig. 3 shows a typical $Q(t)$ pattern, where fire has been detected.

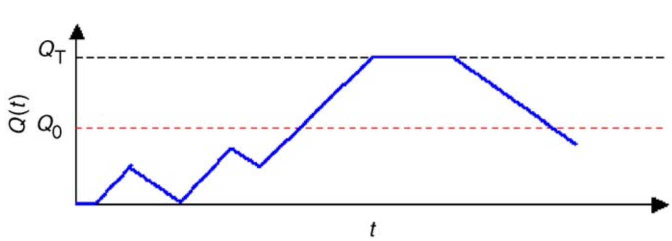

Fig. 3. Typical $Q(t)$ for fire pattern, alarm is triggered if $Q(t) \geqslant Q_0$.

5. Sensitivity

The scene environment and the application requirements—e.g. outdoor scene with low false alarm rate—demand high flexibility from the algorithm. Internal parameters need to be adapted without affecting seriously the detection performance and reliability.

Four sensitivity parameters have been introduced to modify the algorithm's internal parameters according to the requirements and needs:

S_{r} time reaction sensitivity,
S_{l} luminance sensitivity,
S_{m} motion sensitivity,
S_{c} chrominance sensitivity.

The S are in the range: $0 \leqslant S \leqslant 1$ and the default value, or standard configuration, of S, with the exception of S_{c}, whose default value is 1, is 0.5. The change of one or more sensitivity parameters has an immediate impact on a set of corresponding internal parameters.

The time reaction sensitivity $S_{\mathrm{r}} = S_{\mathrm{r}}(v, Q_0, Q_{\mathrm{T}})$ affects the parameters v, Q_0 and Q_{T}, defined in Section 4. Increasing the sensitivity parameter S_{r}, decreases the reaction time of the algorithm: less time is needed to trigger the fire alarm.

The luminance sensitivity $S_{\mathrm{l}} = S_{\mathrm{l}}(\lambda_1, \lambda_2, \eta_1, \eta_2, \beta)$ influences the internal parameters λ_1, λ_2, η_1, η_2 and β, defined in Sections 2 and 3. It is coupled to the light condition of the environment scene: luminance, contrast, saturation, etc.

Increasing the sensitivity S_{l} increases the algorithm's reaction to small changes of the scene's light condition.

The motion sensitivity $S_{\mathrm{m}} = S_{\mathrm{m}}(\alpha)$ controls the build-up of the cumulative matrix $A_{ik}(t)$ by affecting the internal parameter α. Higher values of S_{m} mean that the algorithm is more sensitive to moving objects, e.g. tree's leaves, snow flakes, light reflections, etc.

The chrominance sensitivity S_{c} weights the chrominance indicator $I(\underline{c},t)$, in such a way that if S_{c} is equal to 0, no color analysis is considered; if it is equal to 1, the chrominance indicator is weighted with the same strength as the other indicators. In the case of a monochrome camera or a poor color camera, S_{c} is set to 0.

The sensitivity parameters—except S_{m}—can be adjusted by analyzing offline the values of the features for a representative time interval. A special software simulator tool has been developed to analyze offline the features data and to estimate good values for the sensitivity parameters. False alarm rate and reaction time can be tuned by changing the sensitivity parameters directly on the software tool.

6. Tests and conclusions

The algorithm has been implemented on an Equator MAP-CA™ digital signal processor (DSP). CCTV cameras have been installed in different indoor and outdoor environments and connected to the DSP board. The video images have been down-sampled to the CIF format (288×352 pixels) before being analyzed at 25 fps.

The tests have been subdivided in two phases: a training or learning phase—where the algorithm's parameters are tuned to the scene environment—and an operating phase—where the detection was operative.

During the training phase, which lasted between 15 to 20 days, the algorithm ran with the default sensitivity parameters. Every ca. 30 s (ca. 750 frames) all the features values of one frame were written on a file. Thereafter, the software simulator tool, which delivers the false alarm statistics, analyzed the collected data. It simulates offline the change of the false alarm statistics according to the change of the sensitivity parameters.

In our tests, the sensitivity parameters have been changed in order to have at least 2 false alarms in 10 days. First the time reaction sensitivity, S_r, was adjusted to have 60 s maximum reaction time. Then the luminance sensitivity, S_l, was adjusted according to the environment light conditions, with the intent of reducing the remaining false alarms. Motion and chrominance sensitivity, S_m and S_c, were not changed. If some false alarms were still present, the time reaction sensitivity was decreased again to reach the desired compromise between the false alarm rate and the reaction time.

In the operative phase, the algorithm has been reconfigured with the new sensitivity parameters. The tests in the operative phase lasted for ca. 25 to 40 days.

The algorithm has been tested in different environments. As expected, in the operative phase, the algorithm detected less than one false alarm per week in almost all environments. In one case, the particular environment as well as the position of the camera and the type of the moving object have generated an unfavorable set of conditions so that the false alarm rate was very high under specific lighting conditions.

In general, the tests showed that the method proposed here works under a variety of conditions. It has high reliability and a strong robustness towards false alarm in most critical environments. Moreover, the reaction time and the sensitivity of the algorithm can be adjusted according to the scene complexity and light condition, increasing the flexibility of the method. Tests with true fires in the laboratory showed a fast reaction of the algorithm. Other tests with true fires in non-laboratory environments—working rooms, high rack warehouses—are in progress.

References

[1] Lloyd D. Video smoke detection (VSD-8). Fire Safety Eng, January 2000.

[2] Wieser D, Brupbacher T. Smoke detection in tunnels using video images. 12th international conference on automatic fire detection, Gaithersburg, USA; March 2001.

[3] Foo SY. A rule-base machine vision system for fire detection in aircraft dry bays and engine components. Knowledge-Based Syst 1996;9.

[4] Hamins A, Yang JC, Kashiwagi T. An experimental investigation of the pulsation frequency of flames. Proc Combust Inst 1992;24.