RESEARCH ARTICLE

FEATURE EXTRACTION BASED WILDFIRE SMOKE DETECTION USING COMPUTATIONAL INTELLIGENCE TECHNIQUES

Nisha. V S,

Assistant Professor, Department of Computer Science and Engineering, Lourdes Mount of Engineering and Technology, K.K.Dist, TN. India, E.mail: nishamscse@gmail.com

----

ABSTRACT

------------------------------   ---

Corresponding Author: Nisha. V S,

Corresponding Email:

nishamscse@gmail.com

Early wildfire detection is essential in order to assess an effective response to emergencies and damages. In this paper, we propose a low cost approach based on image processing and computational intelligence techniques, capable to adapt and identify wildfire smoke from heterogeneous sequences taken from a long distance. Since the collection of frame sequences can be difficult and expensive, we propose a virtual environment, based on a cellular model, for the computation of synthetic wildfire smoke sequences. The proposed detection method is tested on both real and simulated frame sequences. Results show that the proposed approach obtains accurate results.

I. INTRODUCTION

Forest fires are an important problem in regions which present hot climate and large areas covered with vegetation. It is reported in [1] that, each year, the 0.1% of the world forest surface is lost due to forest fires. The simplest approach for monitoring this problem is the adoption of manned tow-ers in the most critical areas. However, the deployment of surveillance cameras can be a more feasible solution since a great number of cameras can be monitored in a single place. Recently, computer vision techniques for fire monitoring have been researched. These systems allow the vision at a high distance and without latency, and permit to measure more information related to the fires, such as the size and direction [2].

Since the smoke produced by wildfires is visible much before the flames, the approach proposed in this paper focuses on the detection of smoke regions in frame sequences.

Starting from the preliminary works published in [3] and

[4], we propose in this paper a comprehensive methodology capable to detect wildfire smoke plumes in video sequences, which is able to automatically adapt the system configuration (e.g., the choice of the relevant features to process, the possibility to enrich the available training image sequences with different realistic simulated conditions, the final classification structure and topology) to the specific characteristics of the applicative environments.

This approach is designed to work in real-time applications based on low power and low cost hardware. The considered frame sequences are captured by low resolution cameras in visible light conditions. The approach uses image processing algorithms to extract a set of distinctive features, and then applies computational intelligence techniques to detect the presence of smoke clouds. Computational intelligence methods permit to dynamically adapt the system to the environment, thus achieving a greater flexibility and accuracy than tradi-tional techniques.

The training of computational intelligence classifiers re-quires a big number of frame sequences. However, it can be very difficult and expensive to obtain frame sequences of the same environment in normal situations, during a wildfire, and in different weather conditions. For this reason, we also propose a virtual environment for the computation of synthetic smoke sequences, which is able to inject a distant smoke cloud in frame sequences captured in real applicative scenarios. The proposed method is also able to simulate external forces and adverse environmental conditions.

The contribution of the work is then twofold. We propose a wildfire smoke detection method based on image processing techniques that is specifically designed to be applied in a great variety of environments and weather conditions, and we present a virtual environment designed for the computation of the frame sequences useful to enrich the available images used to train the wildfire detection systems. Moreover, with respect to the works presented in [3,4], we studied and implemented also the possibility to automatically select the most relevant features to be processed by the system in every specific environment, we introduced a sensitivity analysis of the most relevant parameters present in the proposed algorithms, and we presented the possibility to tune the robustness of the smoke detection approach to false alarms, by analyzing the ROC curves obtained with different classification thresholds.

The proposed wildfire detection approach includes two distinct algorithms: Algorithm A is designed to segment smoke clouds in every frame of surveillance frame sequences; Algo-rithm B is designed to raise an alarm in the presence of smoke clouds by evaluating the characteristics of single frames. These algorithms can be divided in the same steps. First, a set of features describing different physical characteristics of the smoke clouds (related to the color, shape, and evolution during the time) are extracted. The used features are similar to the ones described in [3]. Then, computational intelligence techniques are used to classify every frame as "smoke" or "non-smoke". Feature selection techniques are also used in order to reduce the computational time.

The proposed smoke simulation environment is based on the method presented in [4], and uses a cellular model that considers the rules of propagation and collision to recreate the basic principles of advection, diffusion, and buoyancy. External forces, such as wind, are simulated by adding pseudo-random variations in the virtual model. The resulting smoke plume is then merged with a real frame sequence. Finally, adverse environmental conditions, such as low illumination, fog, and acquisition noise, are added to the model.

Experimental results show that the proposed smoke detection approach is able to obtain accurate results on datasets of frame sequences describing a greater number of environmental scenarios and weather conditions with respect to other tech-niques in the literature. Moreover, the proposed method for the simulation of synthetic smoke sequences permits to create

realistic datasets, which can be used to train smoke detection systems and increase their accuracy and adaptability.

II. Previous work

A. Machine vision techniques for smoke detection

Smoke detection systems based only on machine vision techniques that use frame sequences captured in visible light conditions can be divided in systems based on single images and systems based on multiple frames. In the first case, the detection systems can be considered as image segmentation techniques. Techniques belonging to the first class are still studied. For example, a method based on the image histogram is presented in [5]. The majority of the wildfire detection systems belong to the second class and are based on dynamic characteristics of the smoke [6]. This class of smoke detection systems includes methods based on texture, shape, color, movement, energy, and frequency.

The use of color features is presented in the method described in [7], which performs the search of the pixels that present similar values on all the RGB channels. The methods presented in [1,8,9] use texture features to detect smoke regions. The method described in [8] extracts the regions of the image with a high variability and then compares them with regions previously extracted from the smoke frame sequences.

A predetermined color model that describes the smoke is presented in [9] and considers the way in which the model blends into the frame. Tamura features, GLCM, gray intergrowth matrix, wavelet-extracted features [10] are used by the system proposed in [1].

Features related to shape and movement, such as area, direction, circumference, growth, and wavelet transforms, are described in [11–13]. Combinations of these features are also used in the works described in [14–17].

Visual and shape features, wavelet transforms, and statistical modeling are combined in the approach described in [18,19], which in particular deals with the problem of detecting long-range wildfire smokes.

Another approach that aims to detect forest smoke is proposed in [13], and uses features based on motion detection and related to moving edges. Shape properties, like growing regions, are also used in order to discriminate false alarms.

Systems for fire and smoke detection can be based on different classifiers. The use of probabilistic classifiers and support vector machines (SVM) for the detection of smoke in open areas at a medium distance is proposed in [8,20]. SVM are used also in [11,16,17], while neural networks are adopted

in [10,15]. Hidden Markov Models and Bayesian classifiers are used respectively in [19] and [21].

A low-cost method for the segmentation of wildfire smoke in low-quality frame sequences is proposed in [3]. This method is based on neural classifiers and uses features related to the analysis of the motion, color, edges, growing and rising regions.

An example of a machine vision system currently deployed for wildfire monitoring is presented in [22]. The system uses a multi-agent architecture, with an image processing stage for the smoke detection. This stage is based on the motion detection, image segmentation, and analysis of the dynamic pattern, color-space, and texture. A dedicated processing step is also used to reduce false alarms [23].

A review of machine vision system for wildfire smoke detection is proposed in [24].

B. Smoke simulation methods

Since it is difficult to collect a big number of smoke frame sequences, it is possible to use simulated data in order to train and validate the classifiers used by the smoke detection systems. In the literature, there are techniques for the simulation of smoke frame sequences designed for different applicative contexts. The two major applicative contexts related to the simulation of smoke frame sequences are the Computational Fluid Dynamics (CFD) and Computer Graphics (CG). All of the methods appertaining to these areas are based on the equations of the fluid flow.

The Navier-Stokes equations describe the physical model used in fluid dynamics by considering the flow of a compress-ible and viscous fluid in terms of a velocity vector field [25]. Considering gaseous fluids, such as smoke, these equations can be simplified and substituted by the Euler equations [26]. The exact solutions of the physical equations of fluid mechanics are computed by the CFD applications in order to achieve a realis-tic understating of the phenomenon evolution. These methods are especially used in engineering and testing applications. The CG applications, instead, focus on the creation of a realistic fluid that can be processed in real time. The sense of reality and the capability of a quick rendering of the scenes (e.g. for computer games) [27] are more important than the rigorous implementation of physical laws. Most of the CG methods are based on approximations of the described physical laws in order to obtain results comparable to real data.

This paper focuses then on the CG approach, since we are interested in using the resulting simulated smoke for computer vision applications. Moreover, the rigorous solution of

physical laws would be complicated and not necessary in order to simulate a distant smoke plume.

The method described in [28] can be considered as a CG approach and starts with an implementation of a numerical malgorithm that solves the Navier-Stokes equations. This method, however, is stable only when the time step t is sufficiently small. An always stable method is described in [29], for which some implementations are even available [30]. The approach defined as *"Vorticity Confinement"*, described in [27] and [31], is designed in order to overcome the problem of the numerical dissipation. The numerical approximation introduced by the finite precision of computers, in fact, results in an attenuation of some features of gaseous substances. This approach consists in feeding the energy back to the systems in the form of vortexes.

Another technique that is often used to model the flow of a fluid is based on the lattice gas cellular automata (LGA). In order to obtain a macroscopic simulation that realistically represents the modeled fluid, this approach does not propose a solution for the Navier-Stokes equations, but it focuses on the interactions of the microscopic particles that compose the fluid. The smoke plume is then modeled as a grid, in which Boolean variables indicate the positions of the particles. Propagation rules are used to describe the evolution of the system that must conserve mass and momentum [32]. The Navier-Stokes equations can be derived from the LGA [33,34]. Examples of clouds and smoke simulation using the LGA are proposed in [35,36].

Lattice Boltzmann methods (LBM) are a derivation and improvement of the LGA, in which the Boolean values are substituted by the particle densities described by real values. The Navier-Stokes equations can also be derived from the LBM model [37]. An implementation of the LBM for gas simulations is described in [38].

A method that uses a lightweight implementation of a LBM model for the wildfire smoke simulation is proposed in [4].

III. WILDFIRE DETECTION METHODS

The proposed approach, compared with many existing smoke detection systems based on different kinds of sensors and IR illumination techniques (e.g. the one proposed in [22]), requires a less expensive hardware setup, based on low-cost cameras (with a resolution equal to 320×240 pixel), operating in visible light conditions.

The feature extraction step, similar to the one described in [3], considers physical properties of the smoke clouds used also by [13,19], such as moving regions, smoke color, growing and rising regions. However, the proposed approach is designed to be used in real-time applications and is based on feature extraction techniques that require limited resources in terms of computational time and memory usage. For example, the rising regions are described using a less complex model than [13]. Moreover, the approach is designed to be used in different applicative contexts. For example, the proposed techniques for the color analysis and the computation of growing and rising regions perform analyses less related to the environmental scenarios with respect to [8,13,20].

Only a small number of frames, ranging from 5 to 10, is kept under consideration during the feature extraction step, then the computed features are reduced using a sequential forward feature selection algorithm, applied separately on every particular scenario. Computational intelligence techniques are then used to classify the features and adapt the system to the considered environment.

The proposed approach includes two distinct algorithms:

- *Algorithm A* - Performs the segmentation of the smokearea for each frame of a frame sequence.
- *Algorithm B* - Uses computational intelligence classifiersin order to detect the frames containing smoke plumes.

First, a set of features are extracted from the frame sequence. The computed features are then used as inputs of classifiers based on computational intelligence techniques, which output

a Boolean value that represents the classes "smoke" and "non-smoke".

A. Feature extraction

In long range wildfire smoke plumes, frequency analysis is not sufficiently distinctive. The focus is then on features related to the movement and shape.

Algorithm A and Algorithm B use the same feature extraction steps:
1) moving region detection;
2) smoke-color analysis;
3) sharp edge detection;
4) growing region detection;
5) rising region detection;
6) perimeter disorder analysis;
7) feature set computation.

The feature set computation is performed in different manners by the two proposed algorithms.

1) Moving region detection: the detection of moving regions consists in the extraction of the candidate smoke regions by estimating the differences between a frame and a reference background image. This step is based on the works proposed in [39] and [18]. A pixel is considered as moving if:

$$|I(x, y, t) - I(x, y, t-1)| > T_I(x, y, t) \quad , \quad (1)$$

$$|I(x, y, t) - I(x, y, t-2)| > T_I(x, y, t) \quad , \quad (2)$$

where $I(x, y, t-1)$ is the intensity of the pixel (x, y) in the $(t-1)$-th frame of I, and $T_I(x, y, t)$ is an adaptive threshold value. The value of $T_I(x, y, t)$ is updated according to:

$$T_I(x, y, t+1) = \begin{cases} D_I T_I, & \text{if } (x, y) \text{ is stationary} \\ (t) + (1-b)(c|I(x, y, t) - B(x, y, t)|) & \text{if } (x, y) \text{ is a moving pixel} \end{cases} \quad (3)$$

where $B(x, y, t)$ is the intensity of the estimated background related to the frame t at the position (x, y), and c, d are two real positive constants. The initial values of these thresholds are fixed numbers greater than 0. The background image is adaptively estimated as follows:

$$B(x, y, t+1) = \begin{cases} aB(x, y, t) + (1-a)I(x, y, t) & \text{if } (x, y) \text{ is stationary} \\ & \text{if } (x, y) \text{ is a moving pixel} \end{cases} \quad , \quad (4)$$

where $I(x, y, t)$ is the intensity of the pixel at the location (x, y) in the t-th frame of the sequence I, and a is a real and

positive constant close to one. Initially, $B(x, y, 0)$ is equal to the first frame $I(x, y, 0)$.

In order to detect the slow moving regions, two different reference background images are estimated at different instants of time. The background image $B_f(x, y, t)$ is updated at every frame, and the image $B_S(x, y, t)$ is updated with a period of 1 s. A matrix $D_M(x, y, t)$, which represents the motion of every pixel (x, y, t), is then computed:

$$D_M(x, y, t) = \begin{cases} 0 & \text{if } |B_F(x, y, t) - B_S(x, y, t)| \leq T_L \\ \frac{(B_F(x,y,t) - B_S(x,y,t)) - T_L)}{(T_H - T_L)}, & \\ 1 & \text{if } \leq T_L |B_F(x, y, t) - B_S(x, y, t)| \leq T_H \end{cases} \quad (5)$$

where $0 < T_l < T_h$ are fixed threshold values. The elements of $D_M (x, y, t)$ with a value that is less than a threshold T_M are set to 0 in order to remove false growing regions related to the noise present in the frame sequence.

2) Smoke color analysis: first, each frame$I(x, y, t)$is converted from the RGB to the YUV color space. Then, a smoke color feature matrix $D_C (x, y, t)$, similar to the one used in [18], is computed for each pixel (x, y, t) by using the equation:

$$D_C (x, y, t) = \begin{cases} 1 - \dfrac{|U (x, y, t) - 128| + |V (x, y, t) - 128|}{128} & \text{if } (Y (x, y, t) - U (x, y, t) > T_1) \\ & \text{and } (Y (x, y, t) - V (x, y, t) > T_2) \\ & \text{and } (T_3 \leq Y (x, y, t) \leq T_4) \\ 0 & \text{otherwise} \end{cases}$$

(6)

where T_1, T_2, T_3, T_4 are fixed thresholds. The elements of $D_C (x, y)$ with a value that is less than a threshold T_C are then set to 0 in order to reduce the areas that do not represent smoke clouds.

3) Sharp edge detection: the sharp edge detection step aims to search high differences in the luma channel of adjacent frames, usually not present in the regions containing smoke clouds. The differences of intensity are computed by considering the Y channel of the frames $I(x, y, t)$ and $I(x, y, t − 1)$:

$$D_Y (x, y, t) = |(Y (x, y, t) − Y (x, y, t − 1)| . \qquad (7)$$

The result is then thresholded in order to remove the values of $D_Y (x, y, t)$ that are less than a fixed threshold T_S and to consider only the values related to sharp edges.

4) Growing region detection: growing regions of the frame sequence are detected by considering the image computed as the difference between the moving regions at the time instants t and $t − 1$:

$$D_{M−diff} (x, y, t) = D_M (x, y, t) − D_M (x, y, t − 1) . \quad (8)$$

The summation of the resulting matrix is then computed. In order to avoid excessive data fluctuations, only positive values of $D_{M−diff}$ are considered:

$$D_G(t) = \sum_{x,y} (D_{M−diff} (x, y, t)) \ \forall \ D_{M−diff} (x, y, t) > 0 ,$$

(9)

where $D_G(t)$ is related to the time instant t and represents the growing of the region. D_G is then normalized between two empirically estimated thresholds T_{GL} and T_{GH} . The range of values between T_{GL} and T_{GH} is defined in order to match all the possible values of $D_G(t)$ related to growing smoke clouds.

5) Rising region detection: similarly to [19], the rising value is computed as the difference between the y coordinates of the moving pixel with the lowest y value at the frames t−n and t. For each frame, the location of the moving pixel with the lowest y coordinate is extracted:

Two empirically estimated thresholds T_{RL} and T_{RH} are then used to normalize the values of D_R in order to better match the values of the previously computed rising smoke regions.

6) Perimeterdisorderanalysis: first, the Matrix D_M (x, y, t), which describes the motions of the Frame regions, is binarized by using the threshold value T_M . Then, the area and perimeter are computed for each 8-connected region of the binary image. The value that describes the perimeter disorder for each region is computed as:

$$R_P (i, t) = \frac{P_i}{A_i} \forall\ 1 < i < N_b\ , \qquad (11)$$

where R_P (i, t) is the perimeter disorder of the i-th 8-connected region at the time instant t, P_i and A_i are the perimeter and area of the i-th region, and N_b is the number of considered regions.

A matrix D_P (x, y, t) that contains the values of R_P (i, t) related to each pixel is then computed. An approach for false alarms reduction based on a similar principle is described in [23].

7) Feature set computation: The features sets used by theAlgorithm A and Algorithm B are computed in different manners.

- *Algorithm A* - The features are extracted for every pixelof each frame of the frame sequence. The features are computed by considering N previous frames. The grow-

$$D_{M0}(x, y, t) = D_M (x, y, t) > 0,$$
$$H(t) = \min_{y} D_{M0}(x, y, t).$$

The rising value $D_R(t)$ is then computed:

$$D_R(t) = H(t) - H(t - n)\ . \qquad (10)$$

ing value D_G (t), rising value D_R (t), and the perimeter disorder value D_P (t) are the same for every moving pixel of the frame. N growing and rising values are then considered.

For each pixel, $4 + (N \times 2)$ features are extracted:

- *Moving value*: $D_M(x, y, t)$;
- *Smoke color value*: $D_C(x, y, t)$;
- *Sharp edge transition value*: $D_Y(x, y, t)$;
- N×*Growing values*: $D_G(t-N, \ldots, t)$.
- N×*Rising values*: $D_R(t-N, \ldots, t)$.
- *Perimeter disorder value*: $D_P(x, y, t)$.

- *Algorithm B* - For each frame, the features are ex-tracted globally by considering data related to N previous frames. The characteristics related to the single pixels (smoke color, and sharp edge transitions) or related to local regions of the frame (the perimeter disorder value) are aggregated considering the mean value.

For each frame, $3 + (N \times 2)$ features are extracted:

- *Global smoke color value*: mean($D_C(t) > 0$);
- *Global sharp edge transition value*: mean($D_Y (t) > 0$);
- N×*Growing values*: $D_G(t-N, \ldots, t)$.
- N×*Rising values*: $D_R(t-N, \ldots, t)$.
- *Perimeter disorder value*: mean($D_P(t) > 0$);

B. Feature selection and classification

In order to extract the most distinctive features and reduce the required computational time, a sequential forward feature selection algorithm, based on kNN, is used. The feature selection is applied separately on the sets of frame sequences depicting distinct scenarios, in order to better adapt the pro-posed approach to different contexts.

The obtained features are the inputs of computational intelligence classifiers and statistical classifiers. For each sample, the output of the classifiers is a binary value "smoke" / "non-smoke". We tested linear and quadratic classifiers, neural networks, and k-nearest neighbor.

IV. SMOKE SIMULATION METHOD

The proposed smoke simulation method aims at the computation of long-range wildfire smoke clouds, which can be used to train classifiers used by wildfire detection methods. The method is based on a lightweight physics-based model, implemented using image processing techniques.

A smoke simulation method should consider the Navier-Stokes equations. In the case of gaseous fluids (such as smoke), these equations can be simplified in the Euler formulas:

$$\nabla u = 0 ; \qquad (12)$$

$$\frac{\partial u}{\partial t} = -(u\nabla)u - \nabla p + f . \qquad (13)$$

where u is the velocity vector field, p is the pressure and f is an external force. The symbol ∇ is the vector of spatial partial derivatives.

The proposed approach is based on [4], and uses the physical model only as a base, since many features of the smoke cannot be seen from a great distance (e.g. > 1 Km).

The main parameters used in the simulation to control the strength of the smoke are the extension E, density D, and speed S of the simulated smoke plume.

The proposed approach can be divided in the following steps:

1) initial plume computation;
2) external forces computation;
3) velocity estimation;
4) merging with the frame sequence;
5) simulation of adverse conditions.

A. Initial plume computation

The first step is based on the LBM technique and consists in the initialization of a two-dimensional grid with a real finite number of particles, related to a space-time function $f_i(x, y, t)$, which describes the density of the particles at the position (x, y), at the time instant t, and direction i. Only four possible

where $f_i'(x, y, t)$ is the new number of particles with direction i, position (x, y), and time t. C_i is the numerical coefficient of diffusion, representing the percentage of particles in the cell with direction i, after the collision.

The propagation rule is defined as follows:

$$f_i(x + e_{ix}, y + e_{iy}, t + 1) = f_i'(x, y, t) , \qquad (16)$$

where e_{ix} and e_{iy} are the x and y velocities components of the i-th direction respectively, expressed in pixel / frame, and $f_i(x, y, t)$ is the new number of particles after the application of the collision rule.

Similarly to the method presented in [36], a matrix C is used to model the coefficients, which represent the interactions with the air particles and the scattering of the particles in the four possible directions:

$$C = \begin{matrix} 0 & 0.20 & 0 \\ 0.20 & 0.32 & 0.20 \\ 0 & 0.08 & 0 \end{matrix} . \qquad (17)$$

Combining the Eq. 16 and Eq. 15, the equation that globally describes the evolution of the model is defined as:

$$f_i(x + e_{ix}, y + e_{iy}, t + 1) = C_i \sum_{\mathbf{X}} f_i(x, y, t) . \qquad (18)$$

Mass and momentum conservation are guaranteed by imposing

$$\sum_{\mathbf{X}} C_i = 1 ; \qquad e_i = \sqrt{e_{ix}^2 + e_{iy}^2} = 1 . \qquad (19)$$

B. External forces

The wind is modeled by including additional pseudo-random movements to the propagation equation (Eq. 16), proportional to the distance of the particles from the smoke source:

$$m_x = Md_x \quad ; \quad m_y = Md_y , \qquad (20)$$

where m_x and m_y are the movements in the x and y directions, d_x and d_y are the directions of the drift, and M is the drift value. M is computed as:

$$M = \frac{L}{H} \cdot f_d , \qquad (21)$$

where L is the distance of the cell from the source of the smoke, H is the height of the smoke plume, and f_d is a multiplying factor. The values of d_x and d_y are randomly chosen from five possible directions ($0°, 45°, 90°, 135°, 180°$). Different movements are simulated by changing the coeffi-

directions (0°, 90°, 180°, 270°) are considered by the used model. Each instant of time t consists in an image, and the evolution of the model generates a series of frames.

At each time instant, new particles are injected at the starting point of the model, according to the chosen density D and extension E:

$$f_i(x_i, y_j, t) = D, \qquad (14)$$

where x_i, y_j represent a set of E positions, chosen as the starting point of the smoke plume.

At each time instant, the rules of collision and propagation are applied in order to determine the evolution of the model. The collision rule is expressed as:

$$f_i'(x, y, t) = C_i \sum_i f_i(x, y, t), \qquad (15)$$

cients of the matrix C described in Eq. 17. Two different versions of the matrix C are used in order to favor the left (C_L) or the right (C_R) movement of the smoke plume:

$$C_L = \begin{array}{ccc} 0 & 0.20 & 0 \\ 0.20 & 0.32 & 0.08 \\ 0 & 0.20 & 0 \end{array}. \qquad (22)$$

The matrix C_R is computed in a similar way.
The effects of the buoyancy force majorly depend on the gas temperature. The equation of the buoyancy force can then be expressed in relation to the temperature of the gas [38]:

$$F_b = Hg(T_k - T_{ambient}), \qquad (23)$$

where F_b is the buoyancy force, g is the gravity force, H is
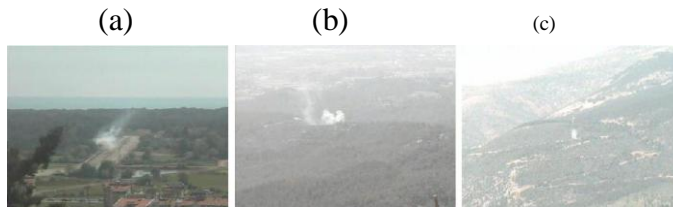
(a)   (b)   (c)

Fig. 1.   Examples of real and synthetic smoke plumes in the same conditions: (a,b,c) real smoke;

the smoke cloud, and $T_{ambient}$ is the ambient temperature. The heat equation determines the evolution of the temperature in the system:

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0, \qquad (24)$$

where u is the function $u(x, y, z, t)$, which describes how the temperature is changing in a determined region over the time. Since the variations of the gas temperature are similar to the way in which the gas particles scatter, our approach considers the temperature as proportional to the concentration of the particles in each cell, similarly to the approach described in [38]. If we assume that the environmental temperature is constant, it is possible to model the buoyancy force (Eq. 23) as a movement along the vertical direction, proportional to the number of particles that are present in the cell:

$$B = \sum_i P_i^{f(x, y, t)} f_b, \qquad (25)$$

where B is the movement of the particles at the location (x, y) and f_b is a proportionality factor.

C. Smoke speed

buoyancy respectively. As a consequence, the greater these intervals are, the slower the resulting simulated smoke is.
It is then possible to choose a particular value for the speed parameter $S = (S_h, S_v)$, and define the values N_c, N_d, N_b accordingly to it.

D. Merging with the frame sequence

The merge of the smoke model with the frame sequence is performed frame-by-frame by adding the intensity of the smoke to the frame:

$$F(x, y, t) = R(x, y, t) + \frac{S(x, y, t)}{f_m} \quad \forall x, y, t, \qquad (28)$$

where R is the real frame sequence, S is the simulated smoke, and f_m is a constant used to control the visibility of the resulting smoke. Positive values of f_m generate a high albedo smoke, and negative values of f_m produce a low albedo smoke.
Examples of real and synthetic smoke plumes are shown in Fig. 1.

The smoke movement speed S is controlled by performing the propagation, drift, and applying the buoyancy force at certain frame intervals. The speed S can be divided in its horizontal and vertical contributions, according to the formulas:

$$S_h = \frac{3}{N_c} + \frac{M}{N_d} \quad ; \qquad (26)$$

$$S_v = \frac{3}{N_c} + \frac{M}{N_d} + \frac{B}{N_b} , \qquad (27)$$

where S_h and S_v are the velocities in the horizontal and vertical directions, 3 represents the propagation rule applied three times, M is the maximum drift value, B is the movement exerted by the buoyancy force, and N_c, N_d, N_b are the three time intervals that determine how many frames need to pass before computing the effects of propagation, drift, and

IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS–PART A: SYSTEMS AND HUMANS

Simulation of synthetic smoke frame sequences - Theparameters were experimentally tuned in order to achieve a good realism. We initialized the cellular model with a density D = 45, and extension E = 5:

$$f(x_i, y_j, 0) = 45 , \ 0 < i < 5, j = 0 . \qquad (29)$$

The threshold used do discard almost-empty cells is $T_P = 0.01$, the

a.b. original image c. smoke region d. segmented smoke region

drift multiplying factor is $f_d = 3$, the buoyancy proportionality factor is $f_b = 5$, the time intervals used for determining the speed S of the simulation are $N_c = 3$, $N_d = 29$, $N_b = 30$. The constant used to control the visibility of the smoke is $f_m = 2$, the parameter used to

increase or decrease the luminance is

· *Smoke detection approach* - The parameters are the samefor the Algorithm A and Algorithm B. The parameters

used for the slow moving region detection are a = 0.7, b = 0.7, c = 5, $T_l = 10$, $T_h = 30$, $T_M = 0.1$. For the smoke color analysis, we used $T_1 = 40$, $T_2 = 40$, $T_3 = 30$, $T_4 = 255$, $T_C = 0.1$. The parameters of the sharp edge detection are $T_S = 30$. For the detection of growing and rising regions, we used $T_{GL} = 5$, $T_{GH} = 10$, $T_{RL} = 0$, $T_{RH} = 10$.

B. Simulation of synthetic smoke frame sequences

E. Simulation of adverse conditions

As a final step, we simulate different adverse conditions, in particular we introduce additive noise, different light conditions (increased and decreased luminance), and the fog effect. A detailed description is presented in [4]. Examples of simulated adverse conditions are shown in Fig. 2.

V. EXPERIMENTAL RESULTS

First, the parameters used by the proposed methods are reported. Then, the experimental results obtained by the method for the simulation of synthetic frame sequences, and by the proposed smoke detection approach are presented. For each method, the description of the procedure used to evaluate.

The first step of the realism validation of the obtained frame sequences consists in the visual examination of the obtained results. Examples of simulated smoke images are shown in Fig. 1, and examples of simulated adverse environmental conditions are shown in Fig. 2.

As visible in Fig. 1 and Fig. 2, the obtained results are very similar to real scenes. Examples of complete simulated frame sequences are available at [40].

In order to obtain a measure of the obtained realism, the simulated frame sequences were compared to real frame sequences by evaluating the results obtained by the proposed smoke detection approach. In particular, the results related to the Algorithm A permitted to evaluate the realism of every pixel of the tested frame sequences.

C. Smoke detection approach: Algorithm A

The tests were performed by using three collections of frame sequences. For each frame, there is a corresponding binary image that describes the segmented smoke plume. Fig. 3 shows an example of segmented smoke plume. The used sets of frame sequences are:

1) *DST1*-5low-quality real frame sequences correspondingto 4 different environments, inluding both smoke and non-smoke samples;

2) *DST2*-5medium-quality real frame sequences corre-sponding to 4 different environments, inluding both smoke and non-smoke samples;

3) *DST3*-5low-quality synthetic frame sequences corre-sponding to 4 different environments, inluding both smoke and non-smoke samples.

From these collections, we created four datasets by considering a number of consecutive frames N equal to 5. The datasets are summarized in Table I.

In order to use the neural networks with the best training for each dataset, we performed an N-fold cross-validation scheme with N = 10 [41]. The used classifiers are two-layer feed-

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}};$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}. \qquad (30)$$

It is possible to observe that the total classification error is always < 0.2%. In particular, the tests performed on the aggregated dataset DTS4 show that the synthetic smoke three different collections of frame sequences:

1) *DSF1* -16low-quality frame sequences corresponding to16 environments, including both smoke and non-smoke frames;

2) *DSF2* -31synthetic frame sequences corresponding to16 environments, including both smoke and non-smoke frames;

3) *DSF3* - composed by both the frame sequences DSF1and DSF2.

The used feature datasets are related to these sets of frame sequences. The datasets are summarized in Table V.

In order to test the generalization capability of the classifiers using the proposed feature set and search the most accurate classification technique, we used the entire feature set to test different classifiers. The topology of the neural networks and the used training algorithm are the same of the ones used for the Algorithm A. Table VI summarize the obtained results by using features extracted considering a number of consecutive frames N = 5. It is possible to observe that feed-forward neural networks obtained the best results. In particular, using a neural classifier with 35 nodes in the hidden layer, the obtained total classification error is always < 0.2%. A similar accuracy was also obtained by using kNN classifiers. Neural networks, however, are more suitable for real-time applications because they are able to obtain better performances in terms of computational time. For example, the time needed by a feed-forward neural network with 10 neurons in the hidden layer to classify the tested feature set is about 1/10 of the time required by a kNN classifier with k = 10.

forward neural networks, tested in different configurations, with different numbers of nodes in the hidden layer: 10, 15, 20, 25. The output node is Boolean and the topology of the hidden layer nodes is tan-sigmoidal.

A summary of the results of the proposed approach is depicted in Table II. We refer to True Positives as the percentage of smoke samples correctly identified as such, and True Negatives as the percentage of non-smoke samples correctly identified as non-smoke. We also reported the values of sensitivity and specificity:

TABLE II
RESULTS OF THE ALGORITHM A

Dataset Name	Hidden L. Size	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
DST1-5	10	1.94	98.06	0.00	0.00	100.00	100.00	0.00
DST2-5	10	1.71	97.84	0.18	0.27	86.31	99.82	0.45
DST3-5	20	1.59	98.37	0.04	0.01	99.33	99.96	0.05
DST4-5	15	1.73	98.08	0.10	0.10	94.73	99.90	0.19

Notes: Hidden L. Size = number of hidden layer nodes of the feed-foreward neaural networks; Sens. = Sensitivity; Spec. = Specificity.

TABLE III
RESULTS OF THE ALGORITHM A VALIDATED USING ONLY REAL DATA

Dataset Name	Hidden L. Size	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
Training: DST4-5 Validation: DST1-5 + DST2-5	20	2.43	97.56	0.01	0.01	99.66	99.99	0.01

Notes: Hidden L. Size = number of hidden layer nodes of the feed-foreward neaural networks; Sens. = Sensitivity; Spec. = Specificity.

plumes obtained by using our approach can be effectively used to increase the performance of the detection system in the case that few real data are available. Detailed results are presented in [3].

In order to test the adherence to real data, we also performed the N-fold cross-validation using the DST4-5 dataset (both real and synthetic data) during the training step, but using only the samples from DST1-5 and DST2-5 datasets (real data) as validation data. The obtained results are depicted in Table III. The total classification error is decreased to 0.01%. For this reason, it is possible to infer that the frame sequences obtained by using the proposed method can effectively improve the generalization capability of the neural classifiers.

The Algorithm A was also evaluated by using frame sequences obtained applying the proposed methods for the simulation of adverse conditions. Table IV summarizes the results of the classification under simulated fog, increased and decreased luminance, and additive Poisson noise. These results show how the adverse conditions do not have a relevant impact on the number of false alarms (Fp), while the number of missed detections can increase, resulting in a lower true positive (Tp) percentage. This is especially verified in the case of fog. However, this is an intrinsic limitation of machine vision systems that operate in visible light conditions.

The proposed techniques for the introduction of simulated adverse conditions are then proved to be useful to adapt and test the smoke detection system in such situations.

D. Smoke detection approach: Algorithm B

TABLE V
DATASETS USED FOR TESTING THE ALGORITHM B

Dataset Name	Frame Sequence	Features × Samples	Non-smoke samples / Smoke samples
DSF1	DSF1	13 × 5895	5536/359
DSF2	DSF2	13 × 5732	5381/351
DSF3	DSF1 + DSF2	13 × 11636	10917/719

1) Results of the smoke detection method on different environmental conditions: we tested the Algorithm B by using

TABLE IV
RESULTS OF THE ALGORITHM A UNDER ADVERSE CONDITIONS

Dataset Name	Hidden L. Size	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
DST1-5	10	1.94	98.06	0.00	0.00	100.00	100.00	0.00
DST1-5 Fog	10	1.95	98.01	0.02	0.02	99.06	99.97	0.04
DST1-5 - Lum	10	1.96	98.04	0.00	0.00	100.00	100.00	0.00
DST1-5 + Lum	10	1.96	98.04	0.00	0.00	100.00	100.00	0.00
DST1-5 + Noise	15	1.94	97.99	0.03	0.04	98.11	99.97	0.07

Notes: Sens. = Sensitivity; Spec. = Specificity; Fog = addition of simulated fog; -Lum: decreased luminance (Y) channel; +Lum = increased luminance (Y) channel; Noise = additive Poisson noise.

frame sequences composed by a large number of frames, with a frame-level classification (smoke / non-smoke). We used

The results obtained by the neural classifier under adverse conditions are summarized in Table VII. The used classifier is a feed-forward neural network and the number of considered consecutive frames is N = 5. These results show that the approach obtained a good accuracy also on data affected by noise. It is possible to observe that the Sensitivity and Specificity obtained on noisy frame sequences are very similar to the ones obtained on the original data. However, in the presence of fog the system obtained a higher detection error. As stated before, this is an intrinsic limitation of machine vision systems operating in visible light conditions.

2) Sensitivity analysis: the robustness of the proposedapproach to improperly tuned parameters has been tested by evaluating the accuracy obtained with different parameter configurations. In particular, the parameter configuration used by the feature extraction process was evaluated by modifying every variable in a range from −20% to +20% of its working point.

The accuracy of the proposed wildfire detection algorithm

TABLE VI
RESULTS OF THE ALGORITHM B

Classifier	Dataset	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
Linear	DSF1	5.19	86.96	6.96	0.90	85.24	92.59	7.85
	DSF2	6.12	90.61	3.26	0.00	100.00	96.52	3.26
	DSF3	4.88	89.22	4.60	1.30	79.00	95.10	5.90
Quadratic	DSF1	6.09	92.20	1.71	0.00	100.00	98.18	1.71
	DSF2	6.12	91.12	2.76	0.00	100.00	97.06	2.76
	DSF3	6.14	68.47	25.35	0.03	99.44	72.98	25.39
kNN-1	DSF1	6.09	93.57	0.34	0.00	100.00	99.64	0.34
	DSF2	6.12	93.60	0.28	0.00	100.00	99.70	0.28
	DSF3	6.18	93.55	0.28	0.00	100.00	99.71	0.28
kNN-3	DSF1	6.09	93.32	0.59	0.00	100.00	99.37	0.59
	DSF2	6.12	93.44	0.44	0.00	100.00	99.54	0.44
	DSF3	6.18	93.34	0.48	0.00	100.00	99.49	0.48
kNN-5	DSF1	6.09	93.15	0.76	0.00	100.00	99.19	0.76
	DSF2	6.12	93.20	0.68	0.00	100.00	99.28	0.68
	DSF3	6.18	93.17	0.65	0.00	100.00	99.30	0.65
kNN-10	DSF1	6.09	92.74	1.17	0.00	100.00	98.75	1.17
	DSF2	6.12	92.83	1.05	0.00	100.00	98.88	1.05
	DSF3	6.18	92.78	1.04	0.00	100.00	98.89	1.04
NN-25	DSF1	6.09	93.67	0.24	0.00	100.00	99.75	0.24
	DSF2	6.12	93.79	0.09	0.00	100.00	99.91	0.09
	DSF3	6.18	93.65	0.17	0.00	100.00	99.82	0.17
NN-30	DSF1	6.09	93.79	0.12	0.00	100.00	99.87	0.12
	DSF2	6.12	93.70	0.17	0.00	100.00	99.81	0.17
	DSF3	6.18	93.61	0.21	0.00	100.00	99.78	0.21
NN-35	DSF1	6.09	93.76	0.15	0.00	100.00	99.84	0.15
	DSF2	6.12	93.81	0.07	0.00	100.00	99.93	0.07
	DSF3	6.18	93.66	0.16	0.00	100.00	99.83	0.16
NN-40	DSF1	6.09	93.74	0.17	0.00	100.00	99.82	0.17
	DSF2	6.12	93.44	0.44	0.00	100.00	99.54	0.44
	DSF3	6.10	93.57	0.25	0.08	98.75	99.73	0.33

Notes. Classification error obtained on the different datasets with the methods: Feed-Forward Neural Network with one hidden layer composed by 25 node (NN-25), 30 nodes (NN-30), 35 nodes (NN-35), 40 nodes (NN-40); k Nearest Neighbor with k = 1 (kNN-1), k = 3 (kNN-3), k = 5 (kNN-5), k = 10 (kNN-10); Linear classifier (linear); Quadratic classifier (Quadratic). Sens. = Sensitivity; Spec. = Specificity.

TABLE VII
RESULTS OF THE ALGORITHM B UNDER ADVERSE CONDITIONS

Dataset Name	Hidden L. Size	Tp (%)	Tn (%)	Fp (%)	Fn (%)	Sens. (%)	Spec. (%)	Total (%)
DSF1	30	6.09	93.79	0.12	0.00	100.00	99.87	0.12
DSF1 Fog	10	5.90	93.08	0.83	0.19	96.94	99.11	1.02
DSF1 - Lum	40	6.09	93.86	0.05	0.00	100.00	99.95	0.05
DSF1 + Noise	30	6.09	93.77	0.14	0.00	100.00	99.86	0.14

Notes: Sens. = Sensitivity; Spec. = Specificity; Fog = addition of simulated fog; -Lum: decreased luminance (Y) channel; Noise = additive Poisson noise.

was then evaluated on all the considered datasets by using neural networks and the 10-fold cross-validation technique. The obtained results are summarized in Table VIII, which reports also the classification error obtained using the initial parameters of the proposed algorithm. It is possible to observe that, in most of the cases, the total error obtained by using modified parameter configurations is similar to the value obtained using the initial configuration. It is then possible to infer that neural networks are able to compensate the presence of noisy data obtained using different parameter configurations of the feature extraction techniques. A similar analysis can also be performed in order to search the best configuration of the proposed approach in specific applicative contexts.

3) Tuning of the classification threshold: the other considered parameter is the classification threshold value C_t, which is applied to the floating point output obtained using neural networks with an output layer composed by a single linear node. For each dataset, we selected the configurations of neural networks which produced the best results with an output layer composed by a Boolean node (reported in

TABLE VIII
RESULTS OF THE SENSITIVITY ANALYSIS OF THE USED THRESHOLDS

Dataset Name	Original Total (%)	Hidden L. size	Threshold Mod. (%)	Resulting Total (%)
DSF1	0.12	10	+20	1.56
DSF1	"	30	−20	0.02
DSF2	0.07	15	+20	0.16
DSF2	"	45	−20	0.07
DSF3	0.16	35	+20	0.91
DSF3	"	30	−20	0.10

DSF1 + Fog	1.02	20	+20	1.88	
DSF1 + Fog	"	25	−20	1.26	
DSF1 - Lum	0.05	30	+20	0.80	
DSF1 - Lum	"	45	−20	0.03	
DSF1 + Noise	0.14	35	+20	0.64	
DSF1 + Noise	"	40	−20	0.10	

TABLE IX
ACCURACY USING THE FEATURE SELECTION PROCESS

Dataset Name	N. of Features	Hidden L. size	Total (%)
DSF1-FFS	3	$1 - 35$	0.14
DSF2-FFS	3	$1 - 35$	0.02
DSF1	13	$1 - 35$	0.13
DSF2	13	$1 - 35$	0.02

Notes: -FFS: dataset computed using the Forward Feature Selection.

Table VI and Table VII) and repeated the accuracy evaluation using an output layer composed by a linear node. We then performed the classification of the datasets using values of C_t equal to [0, 0.1, 0.2, . . . 1]. The obtained receiver operating characteristic (ROC) curves are shown in Fig. 4. The curves show that the method is robust to false negatives. Moreover, it is possible to observe that the number of false positives can be largely tuned according to the applicative situations by varying the classification threshold C_t.

4) Feature selection: in order to decrease the computationaltime needed by the feature selection step and to test the capability of a reduced feature set to characterize different situations, we also evaluated the results obtained by introduc-ing a feature selection step. The best features were searched distinctly for every considered environment. All the frame sequences describing an environment were used to compute a set of features. Then, we searched the 3 most distinctive features by applying the 10-fold cross-validation technique and a forward feature selection algorithm based on a 1-NN classifier. Finally, the accuracy of the proposed smoke detection approach was evaluated using the reduced feature set. We evaluated neural networks with different numbers of nodes in the hidden layer, using the 10-fold cross-validation technique. The results obtained by applying the described procedure to the datasets DSF1, DSF2 are shown in Table IX. It is possible to observe that the total error obtained by using the feature selection strategy is very similar to the one obtained with the complete set of features. Considering that this strategy permits to limit the number of computed features

without reducing the obtained accuracy, the feature selection step performed on the sets of frame sequences related to same environmental scenario can effectively be useful to reduce the computational time and hardware costs.

5) Comparison with methods in the literature: as a comparison, we applied the proposed wildfire detection method on two publicly available collections of frame sequences:

(a)   (b)

(c)   (d)

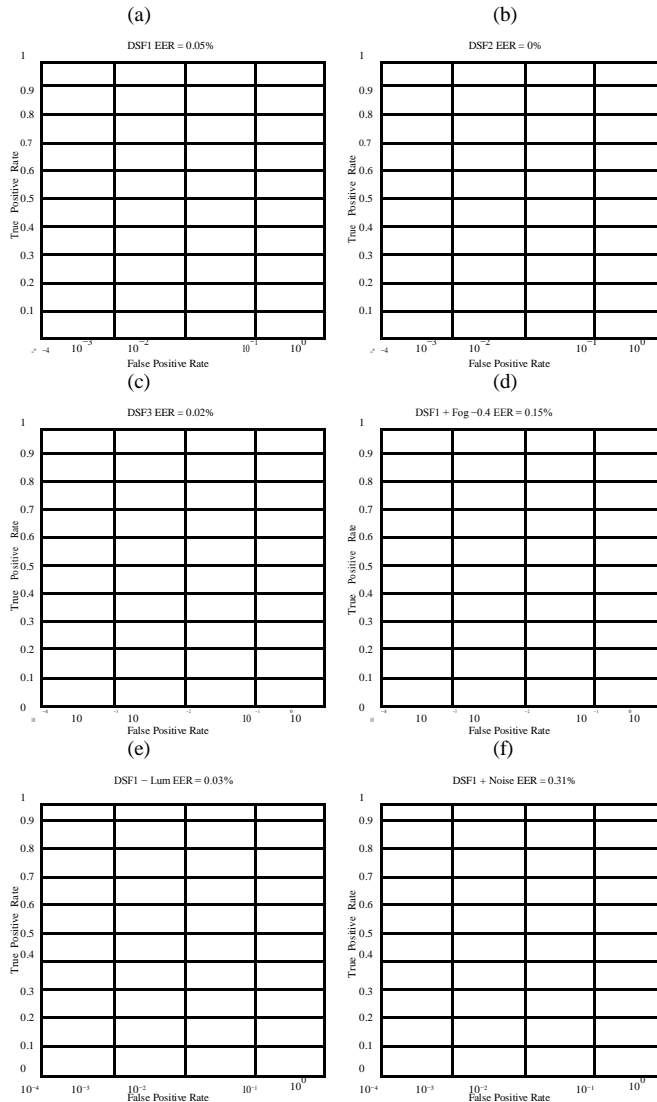(e)   (f)

Fig. 4. ROC curves obtained by varying the classification threshold from 0 to 1: (a) DSF1; (b) DSF2; (c) DSF3; (d) DSF1 + Fog (Parameters modified by −40%); (e) DSF1 - Lum; (f) DSF1 + Noise. The FN number has been minimized, while the FP number can be largely tuned according to the applicative situations.

• *DSPub1*-4frame sequences corresponding to4environments, available at [42], including both smoke and non-smoke frames. A larger dataset, containing these frame sequences, was used to evaluate the methods described in [18]. The frame-level evaluation obtained a number of false alarms equal to 0.1%.

TABLE X
RESULTS OF THE ALGORITHM B ON THE PUBLIC COLLECTIONS OF FRAME SEQUENCES

Dataset Name	Number of Frames	Non-Smoke Frames	Smoke Frames	Hidden L. Size	Sens. (%)	Spec. (%)	Total (%)
DSPub1	6880	6455	425	25	100.00	99.85	0.15
DSPub2	119	60	59	1	100.00	100.00	0.00

Notes: Sens. = Sensitivity; Spec. = Specificity.

The best obtained results are depicted in Table X.

It is possible to observe that the False Positive value Fp corresponding to the DSPub1 dataset is similar to the number of false alarms described in [18]. Moreover, the Sensitivity and Specificity corresponding to the DSPub2 dataset are greater than the corresponding values reported in [6].

VI. CONCLUSIONS

The paper presented a new approach for the detection of wildfire smoke clouds from low quality frame sequences, together with a new technique for the simulation of smoke frame sequences. The smoke detection approach is designed in order to work in real-time applications with off-the-shelf hardware and to be robust to different environmental conditions. It is based on computational intelligence techniques and can classify every pixel (Algorithm A) or frame (Algorithm B) in "smoke" or "non-smoke". The proposed technique for the simulation of smoke frame sequences is designed to compute big datasets of simulated wildfires in different environments and weather conditions, which should be used to train and test visual fire detection methods. The technique is based on the lattice Boltzmann method, is able to introduce the effects of the wind and buoyancy, and permits to simulate adverse environmental conditions.

The performance evaluation of the proposed smoke detection approach was carried out by using both real and simulated smoke frame sequences. Moreover, we evaluated the robustness of the approach to different parameter configurations, the possibility to tune the sensibility to false alarms, and the accuracy obtained by applying feature reduction techniques. Compared with other methods in the literature on datasets composed by real frame sequences, the proposed smoke detection approach obtained accurate results on a greater number of environments and weather conditions. The obtained results are satisfactory and suggest that the approach can be effectively used in different applicative contexts.

• *DSPub2*-4frame sequences corresponding to1environment, available at [43], including both smoke and non-smoke frames. The results obtained by testing the method described in [44] on a larger dataset that contains these frame sequences are reported in [6]. The results obtained by performing a frame-level evaluation are: Sensitivity equal to 50.55%, Specificity equal to 99.62%.

Since no ground truth is publicly available for the datasets DSPub1 and DSPub2, we manually performed a classification of every frame. We then evaluated the accuracy of the proposed wildfire detection method on these datasets by using neural networks with different number of nodes in the hidden layer.

We used two procedures to validate the method proposed for the simulation of synthetic smoke frame sequences. First, we visually compared the obtained results with real wildfire frame sequences. Then, we compared the performances of the proposed smoke segmentation algorithm on both simulated and real frame sequences. The obtained results showed that the simulation method is able to obtain realistic smoke clouds in all the evaluated environmental scenarios. Moreover, we experimentally observed that the use of simulated frame sequences can effectively increase the accuracy and generalization capability of the proposed wildfire smoke detection approach when tested on real frame sequences.

REFERENCES

[1] F. Luming, X. Aijun, and T. Lihua, "A study of the key technology of forest fire prevention based on a cooperation of video monitor and GIS," in *Fourth International Conference on Natural Computation (ICNC)*, vol. 5, October 2008, pp. 391–396.

[2] Z. Liu and A. Kim, "Review of recent developments in fire detection technologies," *Journal of Fire Protection Engineering*, vol. 13, no. 2, pp. 129–149, May 2003.

[3] A. Genovese, R. Donida Labati, V. Piuri, and F. Scotti, "Wildfire smoke detection using computational intelligence techniques," in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA)*, September 2011, pp. 1–6.

[4] ——, "Virtual environment for synthetic smoke clouds generation," in *IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS)*, September2011, pp. 1–6.

[5] D. Krstiniˊc, D. Stipaniˇcev, and T. Jakovˇceviˊc, "Histogram-based smoke segmentation in forest fire detection system," *Information Technology and Control*, vol. 38, no. 3, pp. 237–244, 2009.

[6] T. Jakovˇceviˊc, L. Seriˊc, D. Stipaniˇcev, and D. Krstiniˊc, "Wildfire smoke-detection algorithms evaluation," in *International Conference on ForestFire Research*, 2010.

[7] B.-H. Cho, J.-W. Bae, and S.-H. Jung, "Image processing-based fire detection system using statistic color model," in *International Conferenceon Advanced Language Processing and Web Information Technology*,2008.

[8] H. Maruta, Y. Kato, A. Nakamura, and F. Kurokawa, "Smoke detection in open areas using its texture features and time series properties," in *IEEE International Symposium on Industrial Electronics (ISIE)*, July2009, pp. 1904–1908.

[9] P. Piccinini, S. Calderara, and R. Cucchiara, "Reliable smoke detection in the domains of image energy and color," in *IEEE InternationalConference on Image Processing (ICIP)*, October 2008, pp. 1376–1379.

[10] Y. Cui, H. Dong, and E. Zhou, "An early fire detection method based on smoke texture analysis and discrimination," in *Congress on Imageand Signal Processing (CISP)*, vol. 3, May 2008, pp. 95–99.

[11] J. Yang, F. Chen, and W. Zhang, "Visual-based smoke detection using support vector machine," in *Fourth International Conference on NaturalComputation (ICNC)*, vol. 4, October 2008, pp. 301–305.

[12] D. K. Kim and Y.-F. Wang, "Smoke detection in video," in *WorldCongress on Computer Science and Information Engineering*, 2009, pp.759–763.

[13] I. Grubisic, D. Kolaric, and K. Skala, "Intelligent algorithm for smoke extraction in autonomous forest fire detection," *Microelectronics, Electronics and Electronic Technology. GRID and Visualization System*, pp.363–367, 2009.

[14] J. Gubbi, S. Marusica, and M. Palaniswami, "Smoke detection in video using wavelets and support vector machines," *Fire Safety Journal*, vol. 44, no. 6, November 2009.

[15] Z. Xu and J. Xu, "Automatic fire smoke detection based on image visual features," in *International Conference on Computational Intelligenceand Security Workshops (CISW)*, December 2007, pp. 316–319.

[16] C.-Y. Lee, C.-T. Lin, and C.-T. Hong, "Spatio-temporal analysis in smoke detection," in *IEEE International Conference on Signal andImage Processing Applications (ICSIPA)*, November 2009, pp. 80–83.

[17] C.-B. Liu and N. Ahuja, "Vision based fire detection," in *Proceedingsof the 17th International Conference on Pattern Recognition (ICPR)*,vol. 4, August 2004, pp. 134–137.

[18] B. U. T¨oreyin, "Fire detection algorithms using multimodal signal and image analysis," Ph.D. dissertation, Bilkent University, 2009.

[19] B. U. T¨oreyin and A. Cetin, "Wildfire detection using LMS based active learning," in *IEEE International Conference on Acoustics, Speech andSignal Processing, (ICASSP)*, April 2009, pp. 1461–1464.

[20] H. Maruta, A. Nakamura, and F. Kurokawa, "A new approach for smoke detection with texture analysis and support vector machine," in *IEEEInternational Symposium on Industrial Electronics (ISIE)*, July 2010,pp. 1550–1555.

[21] S. Calderara, P. Piccinini, and R. Cucchiara, "Smoke detection in video surveillance: A MoG model in the wavelet domain," *Lecture Notes in Computer Science*, vol. 5008, pp. 119–128, 2008.

[22] D. Stipaniˇcev, M. Stula, D. Krstini´c, L. Seri´c, T. Jakovˇcevi´c, and M. Bugari´c, "Advanced automatic wildfire surveillance and monitoring network," 2010.

[23] M. Stula, D. Krstini´c, and L. Seri´c, "Intelligent forest fire monitoring system," *Information Systems Frontiers*, vol. 14, pp. 725–739, 2012, 10.1007/s10796-011-9299-8.

[24] T. Jakovˇcevi´c, M. Braovic, D. Stipaniˇcev, and D. Krstini´c, "Review of wildfire smoke detection techniques based on visible spectrum video analysis," in *2011 7th International Symposium on Image and Signal Processing and Analysis (ISPA)*, September 2011, pp. 480–484.

[25] A. J. Chorin and J. E. Marsden, "A mathematical introduction to fluid mechanics," *Springer-Verlag. Texts in Applied Mathematics 4, Second Edition*, 1990.

[26] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics,* 2ND*edition*. Butterworth - Heinemann, Oxford, 1998.

[27] D. Wanning, J. Limin, and J. Hongqiang, "A new method of smoke simulation," in *International Conference on Educational and Network Technology (ICENT)*, June 2010, pp. 267–270.