International Journal of Advanced Robotic Systems

OPEN ACCESS
ARTICLE

A Vision-based Approach to Fire Detection

Regular Paper

Pedro Gomes[1], Pedro Santana[2,*] and José Barata[1]

1 CTS-UNINOVA, Universidade Nova de Lisboa, Portugal
2 ISCTE - Instituto Universitário de Lisboa (ISCTE-IUL), Instituto de Telecomunicações, Portugal
* Corresponding author E-mail: pedro.santana@iscte.pt

Received 24 Apr 2014; Accepted 18 Jun 2014

DOI: 10.5772/58821

Abstract This paper presents a vision-based method for fire detection from fixed surveillance smart cameras. The method integrates several well-known techniques properly adapted to cope with the challenges related to the actual deployment of the vision system. Concretely, background subtraction is performed with a context-based learning mechanism so as to attain higher accuracy and robustness. The computational cost of a frequency analysis of potential fire regions is reduced by means of focusing its operation with an attentive mechanism. For fast discrimination between fire regions and fire-coloured moving objects, a new colour-based model of fire's appearance and a new wavelet-based model of fire's frequency signature are proposed. To reduce the false alarm rate due to the presence of fire-coloured moving objects, the category and behaviour of each moving object is taken into account in the decision-making. To estimate the expected object's size in the image plane and to generate geo-referenced alarms, the camera-world mapping is approximated with a GPS-based calibration process. Experimental results demonstrate the ability of the proposed method to detect fires with an average success rate of 93.1 % at a processing rate of 10 Hz, which is often sufficient for real-life applications.

Keywords Vision Systems, Fire Detection, Smart Cameras, Computer Vision, Object Detection & Tracking

1. Introduction

The safety of people and goods is a topic of great concern to society. The use of video surveillance systems is common practice when safety is to be ensured. These systems generate a high volume of video data that needs to be parsed continuously by human operators. To ease such a tedious and error-prone task in the context of fire detection, this paper proposes an automated vision-based method. Vision-based fire detection assists in coping with the limitations of contemporary smoke detectors, whose operation is constrained to indoor environments. Furthermore, in opposition to smoke detectors, vision-based systems are expected to generate sufficiently detailed data for the estimation of the fire's outline, location, and dynamics. Thermal cameras can do this in an extremely robust way. However, their high cost renders them practically non-existent in the vast majority of surveillance applications. Therefore, fire detection from low-cost surveillance cameras operating within the visible spectrum is expected to generate the highest practical impact.

The classical approach to fire detection by surveillance cameras is to classify the image pixels according to an appearance model of the fire, which can be devised to operate on the RGB [1–4], YCbCr [5], CIE L*a*b* [6] or HSI [7] colour spaces. To lower the false alarm rate, potential fire regions can be discarded when they do not comply with an expected deformation model [3, 6, 8–10]. Checking the dynamic characteristics of the potential fire's

outline is also good practice for the reduction of false positives [11, 12]. In order to also take into account the typical fire's dynamic texture, spatio-temporal wavelet analysis can also be applied [3, 13]. The idea is to exploit the well-known flickering and textured characteristics of flames [14] for their detection.

Despite all the developments in fire detection, there is a lack of reports on integrated solutions ready for deployment in real-life scenarios. To be properly fielded, vision systems need to: (1) handle exceptions; (2) manage the speed-accuracy trade-off; (3) avoid perceptual aliasing situations; and (4) be embedded with seamless calibration procedures. In the case of fire detection, these challenges are related to: (1) handling sudden background changes; (2) determining when a computationally intensive frequency analysis is worth applying; (3) detecting and tracking potential distractors, such as people with fire-coloured clothing; and (4) automatically learning the camera-world coordinates mapping. All these challenges demand the proper selection, adaptation and integration of key previous work, as constrained by robustness and computational parsimony requirements.

In addition to offering an integrated solution, this paper also proposes the following adaptations to key elements of the fire detection system: (1) an attentive mechanism to focus the application of expensive yet accurate frequency analysis; (2) an object detection and tracking pipeline for the reduction of object-induced false fire alarms; (3) a context-based gating of the background learning processes for reducing the chances of erroneously learning moving objects (which is vital for proper object detection and tracking); (4) a GPS-based learning mechanism to automatically approximate the camera-world transformation and, thus, provide the vision system with scale-awareness and enable geo-referenced alarm reporting; (5) a new colour-based model of fire's appearance for enhanced detection accuracy; and (6) a new wavelet-based model of fire's spatio-temporal frequency signature.

Experimental results on a dataset of videos obtained from the Internet show the ability of the proposed method to detect fires with an average success rate of 93.1 % 10 Hz. These results show that, with the proposed method, the activity of human operators becomes less error-prone and less tedious.

This paper is organized as follows. First, Section 2 provides a general overview of the proposed method. Afterwards, Section 3 and Section 4 describe the fire detection and confirmation pipelines, respectively. Following this, the experimental results are presented in Section 5. Finally, Section 6 provides a set of conclusions and highlights future work.

2. General Overview

Figure 1 depicts the proposed method's processing pipeline. The method starts by detecting which regions of the input image correspond to objects in motion.

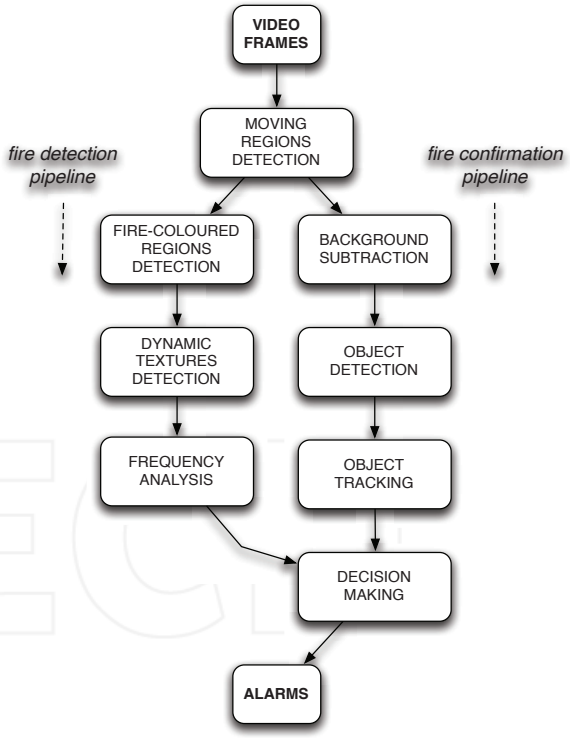


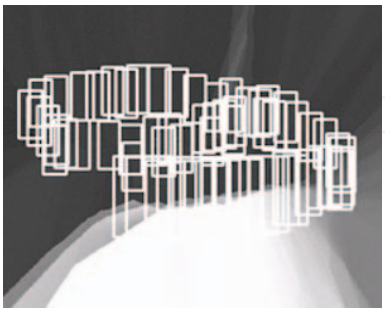
Figure 1. The proposed method's processing pipeline

Many well-known techniques for motion detection can be applied for this purpose [15–17]. Due to its simplicity - which is vital for fast computation - in this work motion detection is done by employing a dynamic threshold to the magnitude of each pixel's intensity variation across three consecutive frames [16]. The result of this process is a binary image $\mathbf{M}(n)$. Next, focusing on these regions, the processing is split into two processing pipelines.

The *fire detection pipeline* (see Section 3) is responsible for: (1) segmenting fire regions according to a colour model; (2) determining which of the segmented regions present a dynamic texture; and (3) filtering out the regions with dynamic texture that do not exhibit the spatio-temporal frequency signature of typical fires. Despite the pipeline's robustness, the presence of challenging fire-coloured moving objects may still induce false fire alarms. To reduce the fire false alarm rate in these situations, knowledge about the location and category of the moving objects in the scene is used. This processing is the responsibility of the *fire confirmation pipeline* (see Section 4), which: (1) detects foreground objects invariant to the presence of shadows; (2) tracks the objects across frames; (3) recognizes the objects' categories; and (4) fuses the objects' awareness with the putative fire alarms to decide whether these should actually be issued. A by-product of this pipeline is the possibility of generating alarms due to the presence of the specified categories of objects being detected and tracked.

Ideally, to foster situational awareness in humans, fire alarms should be geo-referenced. For this purpose, the events natively described in the camera-frame need to be described in the world-frame, that is, they

(a)



(b)

Figure 2. Typical calibration obtained in the environment depicted in Figure 9. (a) GPS positions, recorded during the learning phase, overlaid on satellite imagery - the positions are interpolated for improved readability. The red overlay corresponds to the camera's field of view. (b) Expected bounding box's height represented by brightness level, given the training set represented by the set of overlaid bounding boxes. The smear in the image is a result of the inability of the system to generalize beyond the boundaries imposed by the learning set.

need to be mapped from pixel coordinates to GPS coordinates. Conversely, the inverse mapping allows the object detection and tracking process to reject distractors based on the expected size of the objects' bounding boxes. One possibility to solve the mapping problem would be to know beforehand the GPS position of the camera, to make a few assumptions regarding the planarity of the environment, and to employ a camera calibration procedure. Nonetheless, here, calibration is done by learning from observing a moving person in the environment. This approach avoids the hard planar assumption and makes the calibration procedure intuitive and, thus, easily deployable.

To accumulate learning data, a human equipped with a GPS-enabled PDA moves in the scene while being tracked by the system. During the process, the person's GPS position is stored alongside with the bounding box reported by the tracker. The resulting set of these tuples defines the learning set, which is then processed by a weighted K-means algorithm whenever the world-frame position and the expected object's bounding box must be retrieved, given an image position. In our experiments, k = 3 provided the best results. To match the query and the elements in the learning set, the Euclidean distance is used. Figure 2 depicts a typical calibration. This process provides good enough results given a learning set that covers at least the boundaries of the scene.

3. Fire Detection

3.1 Colour-based Analysis

For the colour-based pixel classification process of the detected moving regions, the performance of three pixel colour classification methods for fire detection [4, 5, 7] are analysed here. The methods described in [7], [5], and [4] rely on the HSI, YCbCr, and RGB colour spaces and are hereafter referred to as the 'H-method', 'Y-method', and 'R-method', respectively. Two novel combinations of these three original methods are also studied here. In the first combination - hereafter the 'HYR-method' - pixels are classified as fire iff consensually classified likewise by the three original methods. In the second combination - hereafter the 'HY-method' - a consensus between the H-method and the Y-method suffices to classify a given pixel as fire. The goal of all these methods is to produce, for a given frame n, a fire/non-fire binary image, $\mathbf{C}(n)$ (see Figure 3).

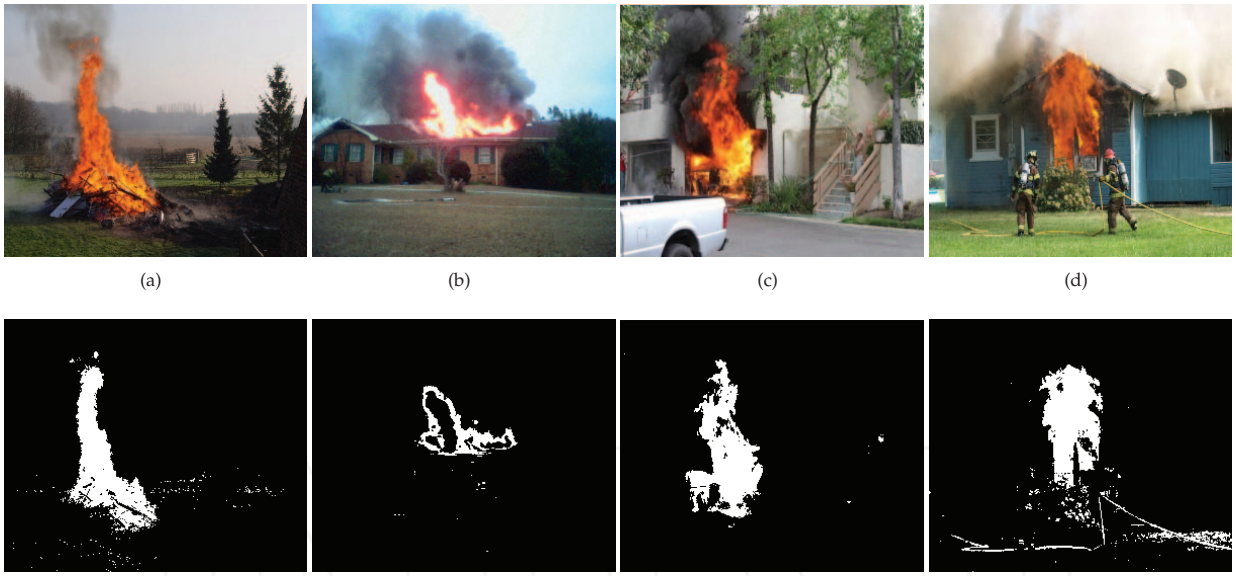
To assess each of the classification methods, a dataset of 217 images containing flames arising from everyday situations was used. For an analysis in context, these images were divided into four categories: indoor, night, rural, and urban. Ground-truth data was generated by hand-labelling all the images' pixels as either fire or non-fire (see Figure 4). The classification methods were applied to all the images and their output binary masks compared to the hand-labelled ground-truth data. The resulting pixel-wise true and false positives and negatives were used to build a confusion matrix for each method-category pair. The two-class Matthews correlation coefficient (MCC) was then calculated for each confusion matrix (see Table 1). The MCC metric is well known for its ability to handle unbalanced datasets. The closer that the MCC is to 1, the better the hypothesis matches the ground-truth. The results show that the HY configuration is the most consistent across the dataset and, thus, it is selected for the proposed method. These results highlight the weakness of the RGB colour space and the complementary role of both the HSI and YCbCr colour spaces in fire detection. The results also show that a colour-based recognition process alone is insufficient for the robust segmentation of fire regions. Table 2 summarizes the average processing time of each of the tested classification methods (see Section 5.1 for details on the experimental setup).

3.2 Dynamic Textures Detection

Fire regions in video streams exhibit a dynamic texture. To perform the rapid detection of dynamic textures, a motion-history image (inspired by [15]) is computed with a parametric recursive temporal filter applied to the fire/non-fire binary image $\mathbf{C}(n)$ (see Section 3.1):

$$\mathbf{D}(n) \leftarrow \mathbf{D}(n-1) + \lambda_1 \cdot \mathbf{C}(n) - \lambda_2 \cdot \big(1 - \mathbf{C}(n)\big), \qquad (1)$$

where λ_1 and λ_2 are empirically defined scalars, and $\mathbf{D}(0) = 0$. With $\lambda_1 = 3$ and $\lambda_2 = 1$, the filter is approximately tuned to the typical frequency exhibited by dynamic fire textures. With these scalars, we ensure that



(a) (b) (c) (d)

(e) (f) (g) (h)

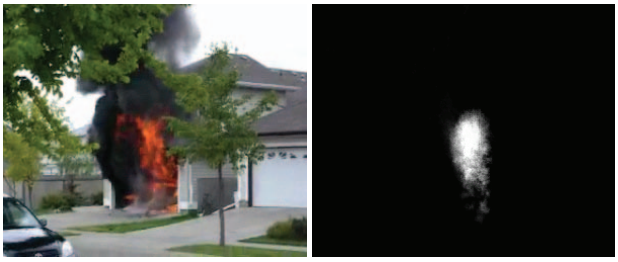
Figure 3. Representative fire-containing images (top-row) and corresponding classification with the proposed HY-method for colour-based fire regions' segmentation (bottom-row), i.e., binary images $\mathbf{C}(n)$. Fire and non-fire labels are represented by white and black pixels, respectively.

Category	Nr. Images	H-meth.	Y-meth.	R-meth.	HYR-meth.	HY-meth.
Urban	124	0.78	0.51	0.19	0.81	0.83
Rural	73	0.73	0.57	0.17	0.77	0.79
Indoors	20	0.36	0.47	0.12	0.43	0.46
Night	21	0.60	0.61	0.06	0.64	0.64
Total	217	0.74	0.54	0.15	0.78	**0.79**

Table 1. Colour-based fire detection comparative results (MCC)

(a) night (b) interior (c) urban (d) rural

Figure 4. Representative fire-containing images (top-row) and corresponding ground-truth fire/no-fire labels (bottom-row) for each of the four analysed categories. Fire and non-fire labels represented by white and black pixels, respectively.



(a) (b)

Figure 5. Dynamic texture detection. (a) Input image of a video stream containing fire. (b) Temporal filter output, $\mathbf{D}(n)$. Brightness is representative of the confidence level stored in $\mathbf{D}(n)$ with respect to the presence of a dynamic texture.

$\mathbf{D}(n)$ grows at the rate λ_1 when there are fire pixels in motion and decreases at the rate λ_2 otherwise. The result of the filtering process, $\mathbf{D}(n)$, is truncated to $[0, 255]$ and is expected to represent the confidence on the presence of a dynamic texture (see Figure 5).

H-meth.	Y-meth.	R-meth.	HY-meth.	HYR-meth.
28 ms	3 ms	14 ms	34 ms	48 ms

Table 2. Colour-based fire detection processing times

To actually determine the presence of a dynamic texture, a threshold κ is applied to $\mathbf{D}(n)$, resulting in a binary image $\mathbf{B}(n)$. The set of connected components present in $\mathbf{B}(n)$ is identified and their bounding boxes compared to those of the connected components detected in previous frames. If a connected component i does not intersect any bounding box of a previous frame, then it is considered new and, as a result, the set of subsequent m input images cropped according to the stored bounding box, V_i, is added to a set of dynamic fire-coloured regions, $T(n+m) \leftarrow T(n+m) \cup \{V_i\}$.

3.3 Spatio-temporal Frequency Analysis

The elements present in $T(n)$ represent videos of dynamic fire-coloured regions, cropped from the input video stream, whose definitive classification as fire regions still needs to be checked. This section describes a spatio-temporal frequency analysis applied to each video in $T(n)$ for this purpose. The high computational load of such an analysis is here compensated for by the fact that it is only applied to a subset of the input video stream (i.e., the elements in $T(n)$).

One of the main characteristics of fire is its flickering rate at a frequency of around 10 Hz, no matter what materials and fuels are involved in the process [18]. This *a priori* knowledge can be exploited to assess the presence of a fire signature on a video stream. For instance, Toreyin et al. [3] proposed analysing the intensity variation in the red channel at each pixel of a given video stream using a one-dimensional (1-D) discrete wavelet transform (DWT), implemented with a two-stage filter bank. The output of the filter bank is taken as the concatenation of the signals obtained from the two half-band high-pass filters. Due to its promising results, we followed closely this DWT-based pipeline for the spatio-temporal frequency analysis stage.

As in [3], the actual decision as to whether a pixel corresponds to a fire region is reached if its corresponding DWT filter bank's output has a minimum of a few peaks (three in the current implementation) above a reasonably high amplitude (100 in the current implementation). For the entire area under analysis to be labelled as fire, the following two conditions must be met. First, the ratio of the analysed pixels that were labelled as fire must be above a given threshold (0.15 in the current implementation). Second, the accumulated number of zero-crossings in the filters' outputs must be above a another given threshold (three-times the area of the image in the current implementation). Peaks, which are not considered in the original, DWT-based model [3], are analysed in a pixel-wise fashion in order to avoid saturating the metric with spuriously high, peaked locations.

To further reduce the chance of generating false alarms, the textured nature of a fire's flames (i.e., its spatial frequency) is also verified. This is attained by means of applying a 2-D DWT to the first image of the video stream under analysis [3]. Distinct from the original application of this method to fire detection [3] - which used a single-stage bank filter - here, a three-stage filter bank is considered for additional accuracy. Furthermore, rather than applying a threshold to the energy of the pixels belonging to a single frame, the threshold is applied to the average, minimum, and sum values computed across the entire frame set. In the current implementation, these thresholds are set to 1.0, 0.01, and 50.0, respectively (see Figure 6). To avoid polluting the frequency analysis with the oscillation caused by the intermittent visualization of the foreground and background in the flame's boundaries, these are first removed.

4. Fire Confirmation

This section describes the pipeline responsible for detecting, tracking, and recognizing objects in the scene, as well as for determining whether these were confused as fire regions by the fire detection pipeline (see Section 3). If not, then a fire alarm is generated by the system.

4.1 Object Detection

In line with Nummiaro et al. [19], the object detection and recognition process uses an object detection technique to initialize a set of particle filters capable of tracking objects according to a colour-based appearance model.

To determine which regions of the visual field are potentially populated by an object, we provide an adaptation to the well-known background subtraction technique proposed by Kim et al. [20]. Background subtraction is used to detect foreground objects because it is faster than solutions based on optical flow (e.g., [21]) and more robust than simple temporal differencing (e.g.,

(a) (b) (c) (d)

Figure 6. Spacial-temporal frequency analysis. (a) An image from the input video stream. (b) An image of the cropped video stream around a dynamic fire-coloured region. Classifications of the cropped video stream with 1-D DWT and 2-D DWT in (c) and (d), respectively.

[16]) (note that these considerations operate under the assumption that the camera is static in the environment).

The original background subtraction model upon which the proposed method builds on [20] uses a vector of codebooks to build a model of the scene's background, which is iteratively updated. To avoid mistakenly learning foreground objects, the presence of motion can be used to cancel the update process, which makes the solution not ideal for dynamic environments. To overcome this limitation, a 3×3 regular grid superposed on the input image is applied here, with each cell being associated with an independently updated background model. This means that a moving object (e.g., a waving tree) in a given region of the scene no longer cancels the background update in other portions of the scene, which suffices for uncrowded scenes. In addition, the update process in each cell only occurs if there is no object already being tracked therein and no considerable motion is observed over a few seconds. Motion information comes from the method proposed by Collins et al. [16], $\mathbf{M}(n)$, and helps to cope with moving objects that are not yet being tracked. To reduce computational load, cells are only updated on a second-wise basis or sooner if a large variation in brightness is observed. The output of the background subtraction process is a binary image representing the foreground pixels, $\mathbf{F}(n)$.

To remove spurious noise and holes in $\mathbf{F}(n)$, a sequence of opening and closing morphological operations is applied. The resulting binary image, $\mathbf{F}^*(n)$, is subsequently processed in order to find a set of connected components present therein and determine which of them are potential objects of interest. To be an object of interest, the connected component must be associated with a bounding box that is similar to the expected bounding box, according to the calibration data (see Section 2), it must not intersect an object that is already being tracked (see Section 4.2), and it must last for a few frames (typically 20). To determine the age of a connected component, correspondences between connected components across frames are employed based on the intersection of their bounding boxes.

Once objects are detected in $\mathbf{F}^*(n)$, shadows are removed according to the model proposed by [22], resulting in a binary image $\mathbf{S}(n)$. To reduce the computational cost, the shadow-removal process is applied only to the bottom region of the object. Figure 7 illustrates the various steps of the object detection process.

4.2 Object Tracking

Based on the binary mask $\mathbf{M}(n)$, the presence of motion in the bounding box of a detected object triggers the deployment of an object tracker. To allow for the proper tracking of multiple objects in the presence of noise, a sequential importance sampling particle filter is used for each tracked object. Each particle corresponds to a possible bounding box of the tracked object. The goal of a particle filter is then to estimate, for each new frame n, the most likely bounding box of a given object o, which is obtained as the weighted average of the bounding boxes associated with all particles:

$$\mathbf{b}^o[n] = \sum_{i=\{1,2,\ldots,k\}} w_i^o[n]\mathbf{b}_i^o[n] \tag{2}$$

where k is the number of particles for each filter, $\mathbf{b}_i^o[n]$ is the bounding box of particle i, which is tracking object o, and $w_i^o[n]$ is the importance weight of the particle.

The importance weight of a given particle i tracking an object o is determined taking into account two factors. The first factor penalizes particles that exhibit a high degree of discrepancy between an appearance model of the region encompassed by the particle, $\mathbf{h}_i^o[n] = \mathcal{H}(\mathbf{b}_i^o[n])$, and a reference appearance model of the object, $\mathbf{h}_r^o[n]$. $\mathcal{H}(\cdot)$ is a function that returns a $25 \times 25 \times 25$ histogram in the RGB colour space of the input image's region, encompassed by the given bounding box, as the object's appearance model [19]. The histogram is computed taking into account only those pixels that have been found to belong to the foreground and have not been marked as shadow, which is enforced by using the composite binary mask $\mathbf{F}(n)\mathbf{S}(n)$ (see Figure 7(d)). This approach removes any spurious background pixels that have been misclassified as shadowless foreground in $\mathbf{S}(n)$. Distant objects and challenging lighting conditions may impact negatively upon the background subtraction and shadow-removal processes. To tackle this problem, a set of

(a) (b) (c) (d)

Figure 7. Background subtraction process. (a) Input image. (b) Foreground image mask prior to shadow removal, $\mathbf{F}(n)$. (c) Foreground image mask after shadow removal, $\mathbf{S}(n)$. (d) Input image masked with $\mathbf{F}(n)\mathbf{S}(n)$.

(a) (b) (c)

Figure 8. Typical occlusion situation between two moving objects. (a) Moments before occlusion. (b) Occlusion occurs. (c) Moments after occlusion. The objects are successfully associated with the same trackers before and after the occlusion.

(a) (b) (c) (d)

Figure 9. Typical occlusion situation between a moving object and a static structure in the environment. (a) Moments before occlusion. (b) Occlusion occurs. (c)-(d) Moments after occlusion. The object is successfully associated with the same trackers before and after the occlusion.

exception-handling situations are considered. If the object being tracked is fully marked as shadow, i.e., completely absent in $\mathbf{S}(n)$, then only $\mathbf{F}(n)$ is used as a mask. If $\mathbf{F}(n)$ also fails to signal the object, then the object is reported as occluded (see below).

The second factor affecting a particle's importance weight penalizes particles with a small ratio of moving pixels in its associated bounding box, given by $\vartheta_i^o[n]$. Formally, the importance weight of a given particle i tracking an object o is:

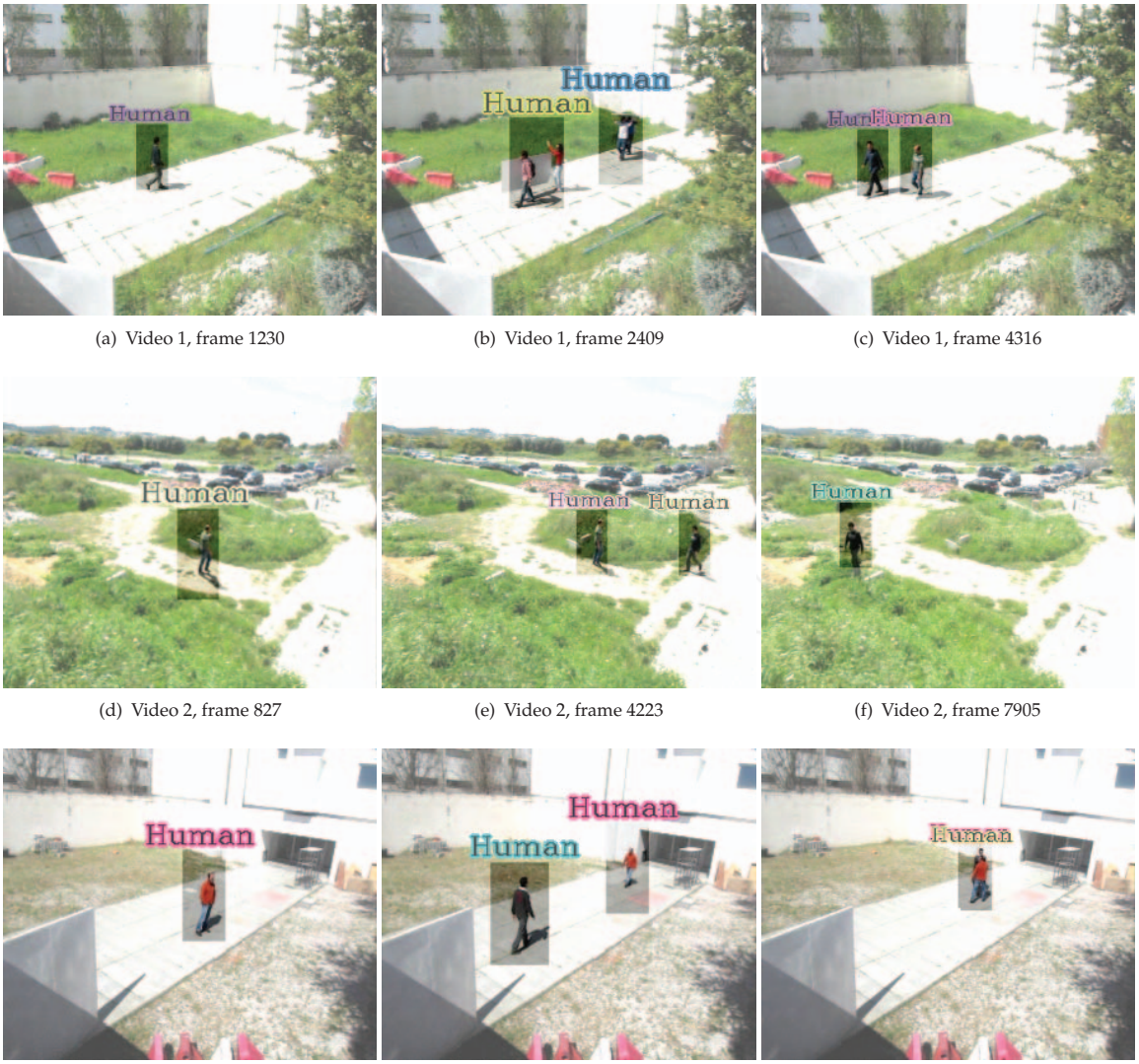
$$w_i^o[n] = \tau \cdot \mathcal{D}(\mathbf{h}_i^o[n], \mathbf{h}_r^o[n]) + (1-\tau)(1-\vartheta_i^o[n]), \qquad (3)$$

where $\mathcal{D}(\cdot,\cdot)$ is the Bhattacharyya distance and τ weights the contribution of each component relative to the overall importance weight. In our experiments, $\tau = 0.98$ provided the best results. Experiments also revealed that 100 particles gave the best speed-accuracy trade-off in the tested dataset. The reference appearance model of the object is updated as follows [19]:

$$\mathbf{h}_r^o[n] = \varphi \cdot \mathbf{h}_r^o[n-1] + (1-\varphi) \cdot \mathcal{H}(\mathbf{b}^o[n]), \qquad (4)$$

where φ is an empirically defined scalar. To allow for rapid adaptation to the appearance dynamics of the object in motion, φ was set to 0.3 in the current implementation. Finally, the proposal distributions are set based on a stochastic first-order motion model [19].

A key topic in object tracking is handling occlusions. One way to handle occlusions is to use multiple cameras with overlapping fields of view [23–25]. To cope with occlusions when using a single camera, particle filter forecasting can be used [19, 26–28]. In this manner, an object o_1 and an object o_2 are tagged as occluded if their estimated bounding boxes, $\mathbf{b}^{o_1}[n]$ and $\mathbf{b}^{o_2}[n]$, respectively, are intersecting. Occluded objects see the stochastic behaviour



(a) Video 1, frame 1230 (b) Video 1, frame 2409 (c) Video 1, frame 4316

(d) Video 2, frame 827 (e) Video 2, frame 4223 (f) Video 2, frame 7905

(g) Video 3, frame 700 (h) Video 3, frame 2133 (i) Video 3, frame 2645

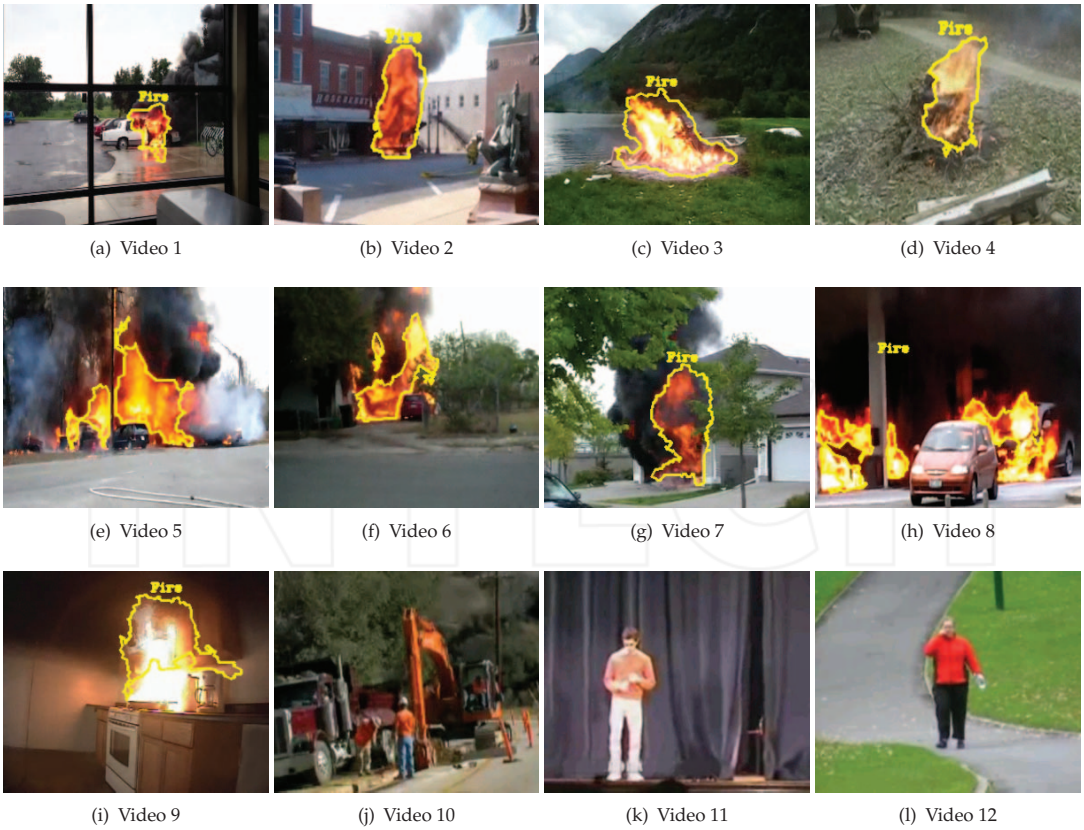
Figure 10. Object recognition and tracking results in three typical situations (one per row). The most likely bounding box of each box is represented by the shaded rectangles. The human labels represent the fact that the proposed method was able to recognize the objects as humans.

of the linear model ruling the proposal distribution grow in order to actively search for the reappearance of the object. In addition, $\tau = 1$ (see Eq. 3), such that the importance weights do not depend on the existence of moving pixels. Finally, the reference appearance histogram is not updated, i.e., $\varphi = 1$ (see Eq. 4). If the objects remain tagged as occluded for more than a few seconds, then the younger tracker is destroyed. Figure 8 depicts a typical occlusion situation between two moving objects. The appearance model shows itself to enable a proper association between the trackers and the moving objects, once the occlusion ends.

If the estimated bounding box of a given tracker finds itself in a region without moving or foreground pixels, then the object being tracked is tagged as occluded by a static structure of the environment. In this case, the system waits for the emergence of a moving object with a similar appearance in the vicinity of the occlusion. If such an object emerges, then it is associated with the tracker and the occlusion is considered to be no longer active. If the object does not emerge for a few seconds, then the tracker is killed. The same logic is applied if objects disappear in the borders of the visual field. Figure 9 depicts a typical occlusion situation between a moving object and a static structure in the environment.

4.3 Decision-making

To reduce false alarms induced by fire-coloured moving objects, one of the two following conditions must be met in order to issue the alarm: (1) the bounding boxes of the fire region in question and of any other object being tracked do not overlap; (2) there is overlap in the bounding boxes but the distance between the current position of the overlapped object and its position when it emerged in the scene does not cross a given threshold (typically 50 pixels). The second condition ensures that only stationary objects are considered as putative fire regions.

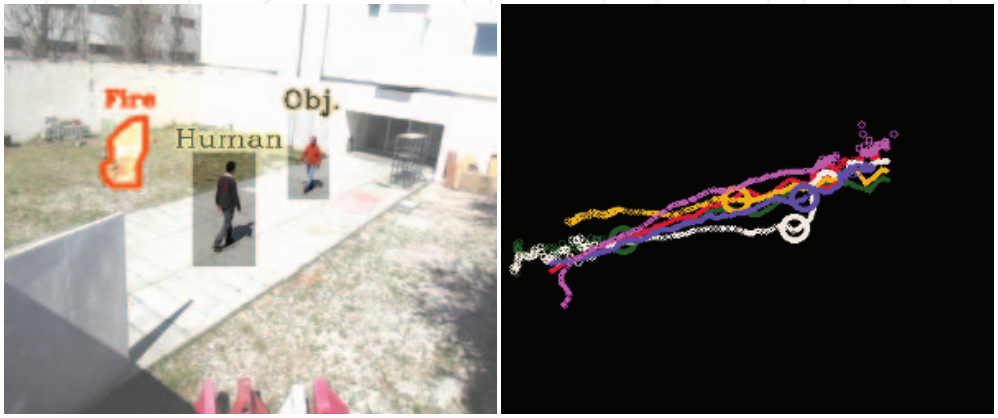


(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4

(e) Video 5 (f) Video 6 (g) Video 7 (h) Video 8

(i) Video 9 (j) Video 10 (k) Video 11 (l) Video 12

Figure 11. Representative frames of the tested dataset with the proposed fire detection algorithm output overlaid. Results are represented by the yellow contours. Note that the presence of distractors (i.e., fire-coloured moving objects) does not influence the algorithm's accuracy.



(a) (b)

Figure 12. Typical output with the proposed method. (a) Output with fire, human and object alarms. Note that in the imaged frame, the person with reddish clothing has not yet been detected as human - this occurs in a later frame. (b) Paths taken by the several objects that crossed the scene. The circles correspond to the position in which the human category classifier reported a positive. Remember that the classifier is used only until it reports a human. Individuals are represented by different colours.

Videos	Nr. of analysed frames	Detection rate (%)
1	2984	94.0
2	3904	98.1
3	812	88.4
4	1423	94.7
5	2422	93.8
6	848	82.3
7	1073	85.9
8	3410	94.2
9	2950	86.0
10	265	100.0
11	1438	100.0
12	324	100.0
Total	21853	**93.1**

Table 3. Fire regions recognition success rate

A quasi-static object exhibiting a fire-like dynamic texture (e.g., a full-bodied person who is shaking but slowly moving and wearing fire-textured clothing) complies with these conditions and, thus, may generate an undesired fire alarm. If the object is of a known non-fire category, the alarm can be discarded immediately. The object's category is estimated with an offline learned classifier based on the histogram of oriented gradient (HOG) descriptor [29]. In the current implementation, only the people category is considered. Thus, with this information, the tracked moving objects are classified as human or else as a generic object.

Although the main purpose in this paper in using an object detection and tracking pipeline is to reduce false fire alarms, it can be used by itself to generate additional useful alarms. One of these alarms regards the presence of moving and static objects. The paths taken by the objects is also reported in order to help the operator to detect suspicious behaviour. To enrich the alarm, the object's category is also reported.

5. Experimental Results

5.1 Experimental Setup

The proposed method was fully implemented in C++ and all tests were run on a Ubuntu Linux machine equipped with an Intel Core 2 Duo 2.53 GHz processor. OpenCV library [30] was used for the implementation of low-level image processing routines. With this setup, the method exhibits a processing rate of 10 Hz.

To validate the fire detection algorithm, a set of 12 videos obtained from the Internet was used. These videos encompass a total of 21992 frames of 300×250 resolution. To validate the object detection and tracking pipeline, a set of three videos with a total of 17247 frames of 600×480 resolution was used. In both cases, different environments and lighting conditions were covered by the dataset.

5.2 Fire Detection Results

Figure 11 depicts key frames from each video in which the fire region recognition pipeline (see Section 3) was tested. These results show the ability of the pipeline to accurately segment the fire regions from the background in a wide variety of situations. Moreover, the results also show that the presence of fire-coloured moving objects, such as people and cars, does not produce false positives. Table 3 summarizes the quantitative results obtained for the same dataset. Overall, the proposed method is able to attain a detection rate of 93.1 %. A detection is considered successful when there are no false positives in the evaluated frame and at least 90 % of the fire region is properly labelled by the proposed method.

5.3 Fire Confirmation Results

Figure 10 illustrates key frames from each video in which the object detection and tracking pipeline was tested. These results show that the proposed method is able to track the objects even when their appearance and that of their surroundings are similar.

Table 4 summarizes the quantitative results obtained for the same dataset. Overall, the proposed method is able to attain 95.9 % and 92.8 % detection and tracking rates, respectively. The detection/tracking rate refers to the number of frames in which the presence/tracking of the objects present in the scene are reported without false positives. The lag between the detection of the object and the initialization of the corresponding tracker is responsible for the lower value in the tracking rate when compared to the detection rate. This lag is caused by the minimum number of frames in which the object must be consecutively detected before creating a new tracker or associating it with an existing one. The proposed method also exhibited robustness to the presence of shadows. Moreover, the presence of waving trees and grass had little effect on the results. This is largely due to the extensive use of the expected object's size, given by the calibration data. A limitation exhibited by the proposed method regards its inability to robustly detect motion in the far field, which means that object detection is delayed until the object is sufficiently near to the camera.

A final test was run in order to assess the ability of the system to run as whole, that is, with the fire detection and confirmation pipelines processing simultaneously. To perform this test, a properly scaled, real fire video was overlaid on a video with multiple people entering and leaving a scene (see Figure 12). This experimental

Videos	Nr. of Analysed Frames	Detection rate (%)	Tracking rate (%)
1	5549	96.3	89.0
2	8245	91.7	89.7
3	3453	99.6	99.6
Total	17247	95.9	**92.8**

Table 4. Object detection and tracking results

setup aims to overcome the logistic difficulty of obtaining videos acquired from static cameras in situations that simultaneously exhibit fire regions and dynamic objects. Nevertheless, for an outdoor environment, this setup shows itself to be capable of producing videos with good enough fidelity for the purposes of the method's validation. In fact, the fire region is promptly detected by the fire detection pipeline. All the moving people are also appropriately detected and tracked by the fire confirmation pipeline. One of the people in the video was wearing fire-coloured clothing and was asked to move vigorously in order to increase the chance that the fire detector would report an alarm. The fire confirmation pipeline always inhibited the sporadic alarms induced by this person's behaviour.

6. Conclusions

A vision-based method for fire detection was presented. Experimental results showed that the method is able to segment fire regions in 93.1 % of the tested dataset. A novelty in the presented method is the use of an object detection and tracking pipeline in order to reduce false fire alarms caused by fire-coloured moving objects. It was shown that the object detection and tracking pipeline by itself is able to produce a success rate of roughly 92.8 % in the tested dataset. Overall, and without special code optimizations, the proposed method runs at 10 Hz.

Background subtraction was implemented in a windowed manner so as to increase robustness to the presence of artefacts. To focus the application of a computationally expensive frequency analysis component and - therefore - reduce the computational cost, an attentive mechanism based on a rough temporal analysis was proposed. An object detection and tracking algorithm was proposed and its output was integrated with the fire detection algorithm to further reduce the false alarm rate in fire detection. A new colour-based model of fire's appearance and a new wavelet-based model of fire's spatio-temporal frequency signature were proposed for improved accuracy. Finally, to avoid hard assumptions regarding the environment's configuration when determining the camera-world mapping, a GPS-based learning procedure was proposed.

In future work, we expect to improve the method in order to attain a full frame-rate on low-end computational units equipping affordable smart cameras. We also intend to include the ability to recognize multiple categories in the object detection and tracking pipeline and introduce the ability to detect smoke (which would reduce the response time in an emergency). Finally, we also intend to build a video dataset acquired from fixed surveillance cameras covering situations of co-existing dynamic objects and real fire regions. This will foster the further development of the object-based fire confirmation pipeline.

7. Acknowledgements

This article is a revised and expanded version of a conference paper [31]. We would like to acknowledge the fruitful comments provided by the anonymous reviewers. This work was partially supported by the QREN-funded project DVA and by CTS multi-annual funding, through the PIDDAC Programme funds.

8. References

[1] W. Phillips, M. Shah, and N. V. Lobo. Flame recognition in video. *Pattern Recognition Letters*, 23(1-3):319–327, 2002.

[2] T.-H. Chen, C.-L. Kao, and S.-M. Chang. An intelligent real-time fire-detection method based on video processing. In *Proc. of the IEEE 37th Annual International Carnahan Conference on Security Techonology*, pages 104–111, 2003.

[3] B. U. Töreyin, Y. Dedeoğlu, U. Güdükbay, and A. E. Çetin. Computer vision based method for real-time fire and flame detection. *Pattern Recognition Letters*, 27(1):49–58, 2006.

[4] J. Chen, Y. He, and J. Wang. Multi-feature fusion based fast video flame detection. *Building and Environment*, 45(5):1113–1122, 2010.

[5] T. Celik and H. Demirel. Fire detection in video sequences using a generic color model. *Fire Safety Journal*, 44(2):147–158, 2009.

[6] T. Celik. Fast and efficient method for fire detection using image processing. *ETRI Journal*, 32(6), 2010.

[7] W.-B. Horng, J.-W. Peng, and C.-Y. Chen. A new image-based real-time flame detection method using color analysis. In *Proc. of the IEEE Networking, Sensing and Control*, pages 100–105, 2005.

[8] T. Celik, H. Demirel, and H. Ozkaramanli. Automatic fire detection in video sequences. In *Proc. of the 14th European Signal Processing Conference (EUSIPCO)*, 2006.

[9] T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu. Fire detection using statistical color model in video sequences. *Journal of*

Visual Communication and Image Representation, 18(2):176–185, 2007.

[10] B. C. Ko, K.-H. Cheong, and J.-Y. Nam. Fire detection based on vision sensor and support vector machines. *Fire Safety Journal*, 44(3):322–329, 2009.

[11] C.-B. Liu and N. Ahuja. Vision based fire detection. In *Proc. of the International Conference on Pattern Recognition (ICPR)*, volume 4, pages 134–137, 2004.

[12] T.-H. Chen, P.-H. Wu, and Y.-C. Chiou. An early fire-detection method based on image processing. In *Proc. of the International conference on Image Processing (ICIP)*, volume 3, pages 1707–1710, 2004.

[13] B. U. Töreyin, Y. Dedeoğlu, and A. E. Cetin. Flame detection in video using hidden markov models. In *Proc. of the IEEE International Conference on Image Processing*, pages 1230–1233, 2005.

[14] A. Hammis, J. C. Yang, and T. Kashiwagi. An experimental investigation of the pulsation frequency of flames. In *Proc. of the 24th International Symposium on Combustion*, volume 24, pages 1695–1702, 1992.

[15] James W. Davis and Aaron F. Bobick. The representation and recognition of human movement using temporal templates. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 928–934. IEEE, 1997.

[16] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyosh, D. Duggins, Y. Tsi, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson. A system for video surveillance and monitoring. In *Proc. of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, 1999.

[17] A. Fernández-Caballero, J. Mira Mira, A. E. Delgado, and M. A. Fernández Graciani. Lateral interaction in accumulative computation: A model for motion detection. *Neurocomputing*, 50:341–364, 2003.

[18] D. Drysdale. *An introduction to fire dynamics*. John Wiley and Sons, third edition edition, 2011.

[19] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.

[20] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[21] J. Black, D. Makris, and T. Ellis. Validation of blind region learning and tracking. In *Proceedings of the second IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 9–16, 2005.

[22] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

[23] S. Khan, O. Javed, and M. Shah. Tracking in uncalibrated cameras with overlapping field of view. In *Proceedings of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, volume 50, 2001.

[24] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1355–1360, 2003.

[25] K. Kim and L. Davis. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filter. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, pages 98–109, 2006.

[26] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D.G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of the 8th European Conference on Computer Vision (ECCV)*, pages 28–39, 2004.

[27] C. Yang, L. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, volume 1, pages 212–219, 2005.

[28] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. volume 21, pages 99–110. Elsevier, 2003.

[29] N. Dadal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of the IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[30] G. Bradski and A. Kaehler. *Learning OpenCV: computer vision with the OpenCV library*. O'Reilly Media, 2008.

[31] P. Santana, P. Gomes, and J. Barata. A vision-based system for early fire detection. In *Proc. of the IEEE Intl. Conf on Systems, Man, and Cybernetics (SMC)*, pages 739–744. IEEE, 2012.