Optical Engineering

SPIEDigitalLibrary.org/oe

Online adaptive decision fusion framework based on projections onto convex sets with application to wildfire detection in video

Osman Günay
Behcet Uğur Töreyin
Ahmet Enis Çetin

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

Optical Engineering 50(7), 077202 (July 2011)

Online adaptive decision fusion framework based on projections onto convex sets with application to wildfire detection in video

Osman Günay
Bilkent University
TR-06800 Bilkent
Ankara, Turkey 06800

Behcet Uğur Töreyin
Texas A&M University – Qatar 23874
P.O. Box 23874
Doha, Qatar 23874
E-mail: behcet.toreyin@qatar.tamu.edu

Ahmet Enis Çetin
Bilkent University
TR-06800 Bilkent
Ankara, Turkey 06800

Abstract. In this paper, an online adaptive decision fusion framework is developed for image analysis and computer vision applications. In this framework, it is assumed that the compound algorithm consists of several sub-algorithms, each of which yields its own decision as a real number centered around zero, representing the confidence level of that particular sub-algorithm. Decision values are linearly combined with weights that are updated online according to an active fusion method based on performing orthogonal projections onto convex sets describing sub-algorithms. It is assumed that there is an oracle, who is usually a human operator, providing feedback to the decision fusion method. A video-based wildfire detection system is developed to evaluate the performance of the algorithm in handling the problems where data arrives sequentially. In this case, the oracle is the security guard of the forest lookout tower verifying the decision of the combined algorithm. Simulation results are presented.
© *2011 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.3595426]

Subject terms: projection onto convex sets; active learning; decision fusion; online learning; wild-fire detection.

Paper 101032R received Dec. 6, 2010; revised manuscript received Apr. 12, 2011; accepted for publication May 9, 2011; published online Jul. 6, 2011.

1 Introduction

An online learning framework called adaptive decision fusion (ADF) is proposed which can be used for various image analysis and computer vision applications. In this framework, it is assumed that the final decision is taken based on a set of real numbers representing confidence levels of various sub-algorithms. Decision values are linearly combined with weights that are updated online using a novel active fusion method based on performing orthogonal projections onto convex sets describing sub-algorithms.

The active learning method used in this work is similar to classifier ensembles used in pattern recognition, in which decisions from different classifiers are combined using a linear combiner.[1] A multiple classifier system can prove useful for difficult pattern recognition problems especially when large class sets and noisy data are involved, because it allows the use of arbitrary feature descriptors and classification procedures at the same time.[2]

The studies in the field of collective recognition, which were started in the middle of the 1950s, found wide application in practice during the last decade, leading to solutions for complex large-scale applied problems.[3] One of the first examples of the use of multiple classifiers was given by Dasarathy and Sheela in Ref. 1, in which they introduced the concept of composite classifier systems as a means of achieving improved recognition system performance compared to employing the classifier components individually. The method is illustrated by studying the case of the linear/nearest neighbor classifier composite system. Kumar and Zhang used multiple classifiers for palmprint recognition by characterizing the user's identity through the simultaneous use of three major palmprint representations and achieving better performance than either one individually.[4] A multiple classifier fusion algorithm is proposed for developing an effective video-based face recognition method.[5] Garcia and Puig present results showing that pixel-based texture classification can be significantly improved by integrating texture methods from multiple families, each evaluated over multi-sized windows.[6] The proposed technique consists of an initial training stage that evaluates the behavior of each considered texture method when applied to the given texture patterns of interest over various evaluation windows of different size.

In this paper, the ADF scheme is applied to a computer vision-based wildfire detection problem. The system based on this method is currently being used in more than 50 forest fire lookout towers. The proposed automatic video-based wildfire detection algorithm is based on five sub-algorithms: (i) slow moving video object detection, (ii) smoke-colored region detection, (iii) wavelet transform based region smoothness detection, (iv) shadow detection and elimination, (v) covariance matrix based classification. Each sub-algorithm decides on the existence of smoke in the viewing range of the camera separately. Decisions from sub-algorithms are combined together by the adaptive decision fusion method. Initial weights of the sub-algorithms are determined from actual forest fire videos and test fires. They are updated by using orthogonal projections onto hyperplanes defined by the fusion weights. It is assumed that there is an oracle monitoring the decisions of the combined algorithm. In the wildfire detection case, the oracle is the security guard. Whenever a fire is detected by the system, the decision should be acknowledged by the security guard. The decision algorithm will also produce false alarms in practice. Whenever an alarm occurs the system asks the security guard to verify its decision. If it is incorrect, the weights are updated according

0091-3286/2011/$25.00 © 2011 SPIE

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

to the decision of the security guard. The goal of the system is not to replace the security guard but to provide a supporting tool to help him or her. The attention span of a typical security guard is only 20 min in monitoring stations. It is also possible to use feedback at specified intervals and run the algorithm autonomously at other times. For example, the weights can be updated when there is no fire in the viewing range of the camera, and then the system can be run without feedback.

The paper is organized as follows: ADF framework is described in Sec. 2. Section 3 introduces the video-based wildfire detection problem. The proposed framework is not restricted to the wildfire detection problem. It can also be used in other real-time intelligent video analysis applications in which a security guard is available. In Sec. 4, each one of the five sub-algorithms which make up the compound (main) wildfire detection algorithm is described. In Sec. 5, experimental results are presented, and the proposed online active fusion method is compared with the universal linear predictor and the weighted majority algorithms. Finally, conclusions are drawn in Sec. 6.

2 ADF Framework

Let the compound algorithm be composed of M-many detection sub-algorithms: $D_1, \ldots, D_M$. Upon receiving a sample input x at time step n, each sub-algorithm yields a decision value $D_i(x,n) \in \mathbb{R}$ centered around zero. If $D_i(x,n) > 0$, it means that the event is detected by the i'th sub-algorithm. Otherwise, it is assumed that the event did not happen. The type of the sample input x may vary depending on the algorithm. It may be an individual pixel, or an image region, or the entire image depending on the sub-algorithm of the computer vision problem. For example, in the wildfire detection problem presented in Sec. 3, the number of sub-algorithms is M=5 and each pixel at the location x of the incoming image frame is considered as a sample input for every detection algorithm.

Let $\mathbf{D}(x,n) = [D_1(x,n) \ldots D_M(x,n)]^T$, be the vector of decision values of the sub-algorithms for the pixel at location x of input image frame at time step n, and $\mathbf{w}(x,n) = [w_1(x,n) \ldots w_M(x,n)]^T$ be the current weight vector. For simplicity, we will drop x in $\mathbf{w}(x,n)$ for the rest of the paper.

We define

$$\hat{y}(x,n) = \mathbf{D}^{\mathbf{T}}(x,n)\mathbf{w}(n) = \sum_i w_i(n) D_i(x,n) \tag{1}$$

as an estimate of the correct classification result $y(x,n)$ of the oracle for the pixel at location x of input image frame at time step n, and the error $e(x,n)$ as $e(x,n) = y(x,n) - \hat{y}(x,n)$. As it can be seen in Sec. 2.1, the main advantage of the proposed algorithm compared to other related methods in Refs. 7–10, is the controlled feedback mechanism based on the error term. Weights of the algorithms producing an incorrect (correct) decision is reduced (increased) iteratively at each time step. In a weighted majority algorithm,[7,11] conflicting weights with the oracle are simply reduced by a factor of 2, which is an *ad-hoc* approach. Another advantage of the proposed algorithm is that it does not assume any specific probability distribution about the data.

2.1 *Set Theoretic Weight Update Algorithm*

Ideally, weighted decision values of sub-algorithms should be equal to the decision value of $y(x,n)$ the oracle:

$$y(x,n) = \mathbf{D}^T(x,n)\mathbf{w}^*, \tag{2}$$

which represents a hyperplane in the M-dimensional space, $\mathbb{R}^M$. Hyperplanes are convex in $\mathbb{R}^M$. At time instant n, $\mathbf{D}^T(x,n)\mathbf{w}(n)$ may not be equal to $y(x,n)$. The next set of weights are determined by projecting the current weight vector $\mathbf{w}(n)$ onto the hyperplane represented by Eq. (2). This process is geometrically depicted in Fig. 2. The orthogonal projection $\mathbf{w}(n+1)$ of the vector of weights $\mathbf{w}(n) \in \mathbb{R}^M$ onto the hyperplane $y(x,n) = \mathbf{D}^T(x,n)\mathbf{w}^*$ is the closest vector on the hyperplane to the vector $\mathbf{w}(n)$ (cf. Fig. 1).

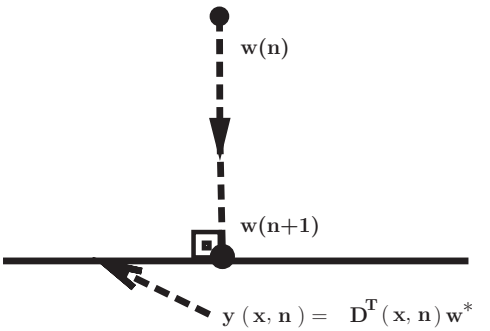


Fig. 1 Orthogonal projection: Find the vector $\mathbf{w}(n+1)$ on the hyperplane $y(x,n) = \mathbf{D}^T(x,n)\mathbf{w}$ minimizing the distance between $\mathbf{w}(n)$ and the hyperplane.

Let us formulate the problem as a minimization problem:

$$\min_{\mathbf{w}^*} ||\mathbf{w}^* - \mathbf{w}(n)||$$

$$\text{subject to } \mathbf{D}^T(x,n)\mathbf{w}^* = y(x,n). \tag{3}$$

The solution can be obtained by using Lagrange multipliers:

$$\mathcal{L} = \sum_i [w_i(n) - w_i^*]^2 + \lambda[\mathbf{D}^T(x,n)\mathbf{w}^* - y(x,n)]. \tag{4}$$

Taking partial derivatives with respect to w_i^*:

$$\frac{\partial \mathcal{L}}{\partial w_i^*} = 2(w_i(n) - w_i^*) + \lambda D_i(x,n), \quad i = 1, \ldots, M, \tag{5}$$

setting the result to zero:

$$2(w_i(n) - w_i^*) + \lambda D_i(x,n) = 0, \quad i = 1, \ldots, M, \tag{6}$$

and defining the next set of weights as $\mathbf{w}(\mathbf{n}+\mathbf{1}) = \mathbf{w}^*$, a set of M equations is obtained:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\lambda}{2}\mathbf{D}(x,n). \tag{7}$$

The Lagrange multiplier, λ, can be obtained from the condition equation:

$$\mathbf{D}^T(x,n)\mathbf{w}^* - y(x,n) = 0 \tag{8}$$

as follows:

$$\lambda = 2\frac{y(x,n) - \hat{y}(x,n)}{||\mathbf{D}(x,n)||^2} = 2\frac{e(x,n)}{||\mathbf{D}(x,n)||^2} \tag{9}$$

where the error, $e(x,n)$, is defined as $e(x,n) = y(x,n) - \hat{y}(x,n)$ and $\hat{y}(x,n) = \mathbf{D}^T(x,n)\mathbf{w}(n)$. Plugging this into Eq. (7)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(x,n)}{||\mathbf{D}(x,n)||^2}\mathbf{D}(x,n) \tag{10}$$

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

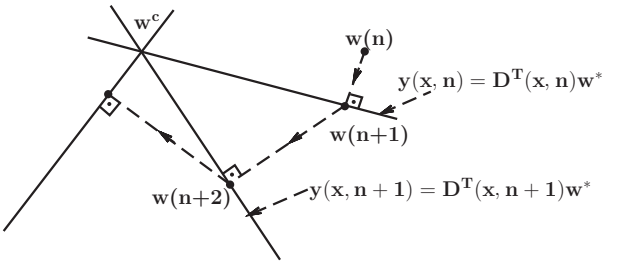


Fig. 2 Geometric interpretation: Weight vectors corresponding to decision functions at each frame are updated as to satisfy the hyperplane equations defined by the oracle's decision $y(x, n)$ and the decision vector $\mathbf{D}(x, n)$. Lines represent hyperplanes in $\mathbb{R}^M$. w^c is the weight vector at the intersection of the hyperplanes.

is obtained. Hence, the projection vector is calculated according to Eq. (10).

Whenever a new input arrives, another hyperplane based on the new decision values $\mathbf{D}(x, n)$ of sub-algorithms, is defined in $\mathbb{R}^M$

$$y(x, n+1) = \mathbf{D}^T(x, n+1)\mathbf{w}^*. \tag{11}$$

This hyperplane will probably not be the same as $y(x, n) = \mathbf{D}^T(x, n)\mathbf{w}(n)$ hyperplane as shown in Fig. 2. The next set of weights, $\mathbf{w}(n+2)$, are determined by projecting $\mathbf{w}(n+1)$ onto the hyperplane in Eq. (11). Iterated weights converge to the intersection of hyperplanes.[12,13] The rate of convergence can be adjusted by introducing a relaxation parameter μ to Eq. (10) as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(x, n)}{||\mathbf{D}(x, n)||^2}\mathbf{D}(x, n), \tag{12}$$

where $0 < \mu < 2$ should be satisfied to guarantee the convergence according to the projections onto convex sets theory.[14–17]

If the intersection of hyperplanes is an empty set, then the updated weight vector simply satisfies the last hyperplane equation. In other words, it tracks decisions of the oracle by assigning proper weights to the individual sub-algorithms.[15,16]

The relation between support vector machines and orthogonal projections onto halfplanes was established in Refs. 16,18 and19. As pointed out in Ref. 18, a support vector machine (SVM) is very successful in batch settings, but it cannot handle online problems with drifting concepts in which the data arrive sequentially.

3 Application: Computer Vision–Based Wildfire Detection

The set theoretic adaptive decision fusion framework described in detail in Sec. 2 with tracking capability is especially useful when the online active learning problem is of a dynamic nature with drifting concepts.[20–22] In the video-based wildfire detection problems introduced in this section, the nature of forestal recordings vary over time due to weather conditions and changes in illumination, which makes it necessary to deploy an adaptive wildfire detection system. It is not feasible to develop one strong fusion model with fixed weights in this setting with drifting nature. An ideal online active learning mechanism should keep track of drifts in video and adapt itself accordingly. The projections in Eq. (10) adjust the importance of individual sub-algorithms by updating the weights according to the decisions of the oracle.

Algorithm 1 The pseudo-code for the ADF algorithm.

Adaptive Decision Fusion (x,n)

for $i = 1$ to M **do**

$w_i(0) = \frac{1}{M}$, *Initialization*

end for

$e(x, n) = y(x, n) - \hat{y}(x, n)$

for $i = 1$ to M **do**

$w_i(n+1) \leftarrow w_i(n) + \mu \frac{e(x,n)}{||\mathbf{D}(x,n)||^2} D_i(x, n)$

end for

$\hat{y}(x, n) = \sum_i w_i(n) D_i(x, n)$

if $\hat{y}(x, n) \geq 0$ **then**

return 1

else

return -1

end if

Manned lookout posts are widely available in forests all around the world to detect wild fires. Surveillance cameras can be placed in these surveillance towers to monitor the surrounding forestal area for possible wild fires. Furthermore, they can be used to monitor the progress of the fire from remote centers.

As an application of ADF, a computer vision-based method for wildfire detection is presented in this article. Currently, the reported average wildfire detection time is 5 min in manned lookout towers in Turkey. Security guards have to work 24 h in remote locations under difficult circumstances. They may get tired or leave the lookout tower for various reasons. Therefore, computer vision-based video analysis systems capable of producing automatic fire alarms are necessary to help the security guards reduce the average forest fire detection time.

Cameras, once installed, operate at forest watch towers throughout the fire season for about 6 months, which is mostly dry and sunny in the Mediterranean region. There is usually a guard in charge of the cameras, as well. The guard can supply feed-back to the detection algorithm after the installation of the system. Whenever an alarm is issued, she/he can verify it or reject it. In this way, she/he can participate in the learning process of the adaptive algorithm.

As described in Sec. 4, the main wildfire detection algorithm is composed of five sub-algorithms. Each algorithm has its own decision function yielding a zero-mean real number for slow moving regions at every image frame of a video sequence. Decision values from sub-algorithms are linearly combined and weights of sub-algorithms are adaptively updated in our approach.

Notice that individual decision algorithms do not produce binary values 1 (correct) or −1 (false), but they do produce

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

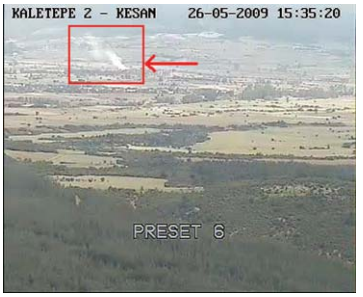


Fig. 3 Snapshot of a typical wildfire smoke captured by a forest watch tower which is 5 km away from the fire (rising smoke is marked with an arrow).

a real number in $[-1,1]$. If the number is positive (negative), then the individual algorithm decides that there is (not) smoke due to forest fire in the viewing range of the camera. The higher the absolute value, the more confident the sub-algorithm. Individual decision algorithms are based on support vector machines or other classifiers, depending on the nature of the problem.

There are several approaches on automatic (forest) fire detection in the literature. Some of the approaches are directed toward detection of the flames using infra-red and/or visible-range cameras, and some others aim at detecting the smoke due to wildfire.[23–26] There are recent papers on sensor-based fire detection.[27–29] Infrared cameras and sensor-based systems have the ability to capture the rise in temperature, however, they are much more expensive compared to regular pan-tilt-zoom (PTZ) cameras. An intelligent space framework is described for indoor fire detection in Ref. 30. However, in this paper, an outdoor (forest) fire detection method is proposed.

The flames and smoke of a wildfire can be viewed with a regular visible-range camera. However, especially in the early stages, it is hard for the flames of a wildfire to fall into the viewing range of a camera mounted on a forest watch tower due to trees and foliage occluding the scene in forestal areas, unless the fire takes place very close to the tower. On the contrary, smoke rising up in the forest due to a fire is usually visible from long distances. Indeed, only plumes of smoke fell into the viewing ranges of cameras throughout our joint project with the Turkish General Directorate of Forestry, which spanned a duration of 3 years. A snapshot of typical wildfire smoke captured by a lookout tower camera from a distance of 5 km is shown in Fig. 3.

Guillemant and Vicente[26] based their method on the observation that the movements of various patterns, like smoke plumes, produce correlated temporal segments of gray-level pixels. They utilized fractal indexing using a space-filling Z-curve concept along with instantaneous and cumulative velocity histograms for possible smoke regions. They made smoke decisions about the existence of smoke according to the standard deviation, minimum average energy, and shape and smoothness of these histograms. It is possible to include most of the currently available methods as sub-algorithms in the proposed framework and combine their decisions using the proposed ADF method.

Smoke at far distances (>100 m to the camera) exhibits different spatio-temporal characteristics than nearby smoke and fire.[31–34] This demands specific methods explicitly developed for smoke detection at far distances, rather than using nearby smoke detection methods described in Refs. 33 and 35. The proposed approach is in accordance with the "weak" artificial intelligence (AI) framework[36] introduced by Hubert L. Dreyfus, as opposed to "generalized" AI. According to this framework, each specific problem in AI should be addressed as an individual engineering problem with its own characteristics.[37,38]

4 Building Blocks of Wildfire Detection Algorithm

Wildfire detection algorithm is developed to recognize the existence of wildfire smoke within the viewing range of the camera monitoring forestal areas. The proposed wildfire smoke detection algorithm consists of five main sub-algorithms: i. slow moving object detection in video, ii. smoke-colored region detection, iii. wavelet transform-based region smoothness detection, iv. shadow detection and elimination, v. covariance matrix-based classification, with decision functions, $D_1(x,n)$, $D_2(x,n)$, $D_3(x,n)$, $D_4(x,n)$, and $D_5(x,n)$, respectively, for each pixel at location x of every incoming image frame at time step n. Computationally efficient sub-algorithms are selected in order to realize a real-time wildfire detection system working in a standard PC. The decision functions are combined in a linear manner, and the weights are determined according to the weight update mechanism described in Sec. 2.

Decision functions $D_i, i=1,\ldots,M$ of sub-algorithms do not produce binary values 1 (correct) or -1 (false), but they do produce real numbers centered around zero for each incoming sample x. If the number is positive (negative), then the individual algorithm decides that there is (not) smoke due to a forest fire in the viewing range of the camera. Output values of decision functions express the confidence level of each sub-algorithm. The higher the value, the more confident the algorithm.

4.1 *Detection of Slow Moving Objects*

The goal of this sub-algorithm is to detect slow moving regions in video. Video objects at far distances to the camera seem to move slower (px/s) in comparison to the nearby objects moving at the same speed. Ordinary moving object detection schemes estimate a background image and detect moving regions by subtracting the current image frame of the video from the estimated moving object. In order to eliminate fast moving objects such as birds, two background images, $B^{\text{fast}}(x,n)$ and $B^{\text{slow}}(x,n)$ corresponding to the scene with different update rates are estimated, where x is the location of the pixel at frame number n. This approach is used in left (abandoned) or removed object detection algorithms. These objects represent the stationary regions in image frames that are not in the background but are present in the scene at a later time. When a new object is brought into the scene, it is called left (abandoned) object, and when an object is removed from the scene it is called removed object.[39,40]

In Ref. 41, a background image $B(x,n+1)$ at time instant $n+1$ is recursively estimated from the image frame $I(x,n)$ and the background image $B(x,n)$ of the video as follows:

$$B(x,n+1) = \begin{cases} aB(x,n)+(1-a)I(x,n) & \text{if } x \text{ is stationary} \\ B(x,n) & \text{if } x \text{ is a moving pixel} \end{cases}, \quad (13)$$

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

where $I(x,n)$ represents the intensity value of the pixel at location x in the n'th video frame I, and a is a parameter between 0 and 1. It is assumed that the camera is stationary. Initially, $B^{\text{fast}}(x,0)$ and $B^{\text{slow}}(x,0)$ can be taken as $I(x,0)$. Moving pixels are determined by thresholding the difference between the current and previous image frames.[41] Background images $B^{\text{fast}}(x,n)$ and $B^{\text{slow}}(x,n)$ are updated as in Eq. (13) with different update rates. In our implementation, $B^{\text{fast}}(x,n)$ is updated at every frame and $B^{\text{slow}}(x,n)$ is updated once in a second with $a=0.7$ and 0.9, respectively. As a result, it is possible to eliminate objects which can enter and leave the viewing range of the camera in less than 1 s. Other slow moving regions within the viewing range of the camera are detected by comparing background images, B^{fast} and B^{slow}.[39,40,42] If there exists a substantial difference between the two images for some period of time, then an alarm for the slow moving region is raised, and the region is marked.

The decision value indicating the confidence level of the first sub-algorithm is determined by the difference between background images. The decision function $D_1(x,n)$ is defined as:

$$D_1(x,n)=\begin{cases}-1 & \text{if } |B^{\text{fast}}(x,n)-B^{\text{slow}}(x,n)|\le T_{\text{low}}\\ 2\dfrac{|B^{\text{fast}}(x,n)-B^{\text{slow}}(x,n)|-T_{\text{low}}}{T_{\text{high}}-T_{\text{low}}}-1 & \text{if } T_{\text{low}}\le|B^{\text{fast}}(x,n)-B^{\text{slow}}(x,n)|\le T_{\text{high}}\\ 1 & \text{if } T_{\text{high}}\le|B^{\text{fast}}(x,n)-B^{\text{slow}}(x,n)|\end{cases}, \tag{14}$$

where $0<T_{\text{low}}<T_{\text{high}}$ are experimentally determined threshold values. In our implementation, T_{low} (T_{high}) is taken as 10 (30) on the luminance (Y) component of video. The luminance component Y takes real values in the range [0, 255] in an image.

The confidence value is 1 (-1), if the difference $|B^{\text{fast}}(x,n)-B^{\text{slow}}(x,n)|$ is higher (lower) than threshold T_{high} (T_{low}). The decision function $D_1(x,n)$ takes real values in the range $[-1,1]$ if the difference is in between the two threshold values. The overall algorithm is not very sensitive to the threshold values because the above equation is just a soft-decision function. Let us assume that $|B^{\text{fast}}(x,n)-B^{\text{slow}}(x,n)|=9$ and then T_{low} is 5 instead of 10, then the decision function would still take a negative value indicating that there is no motion. Instead of getting -1 it would take -0.68, which is not as strong as -1, but it is still a negative decision.

Smoke due to forest fires at further distances (>5 km) to the camera seems to move even slower. Therefore, smoke regions at these distances appear neither in B^{fast} nor B^{slow} images. This results in lower difference values between background images B^{slow} and B^{fast}. In order to have substantial difference values and detect smoke at distances further than 5 km to the camera, B^{fast} terms in Eq. (14) are replaced by the current image I.

Background images in Eq. (14) can also be estimated using more complex schemes such as Ref. 43. The method in Ref. 41 is selected in this work because of its computational efficiency.

4.2 *Detection of Smoke-Colored Regions*

Whenever a slow moving region is detected, its color content is analyzed. Smoke due to forest fires is mainly composed of carbon dioxide, water vapor, carbon monoxide, particulate matter, hydrocarbons, and other organic chemicals, nitrogen oxides, trace minerals, and some other compounds.[44] Apparently, the whitish-gray color of the rising plume is primarily due to water vapor and carbon particles in the output fire composition. Other output chemicals, like carbon dioxide and carbon monoxide, are not visible. We used a Gaussian mixture model-based color modeling approach for flame color detection in Ref. 34. However, we experimentally observe that it is sufficient to use the YUV (luminance-bandwidth-chrominance) color space without any need for sophisticated color modeling for smoke detection because gray color values can be easily represented in the YUV color space. Ideally, chrominance values (U and V) should be close to zero in gray-colored smoke regions. Also, luminance value of smoke regions should be high, especially at the initial phases of a wildfire, as shown in Fig. 3. The confidence value corresponding to this sub-algorithm should account for these characteristics. The decision function $D_2(x,n)$ takes values between 1 and -1, depending on the values of the $Y(x,n)$, $U(x,n)$, and $V(x,n)$ channel values. The decision function $D_2(x,n)$ is defined as:

$$D_2(x,n)=\begin{cases}1-\dfrac{|U(x,n)-128|+|V(x,n)-128|}{12}, & \text{if } Y(x,n)>T_I\\ -1, & \text{otherwise}\end{cases}, \tag{15}$$

where $Y(x,n)$, $U(x,n)$, and $V(x,n)$ are the luminance and chrominance values of the pixel at location x of the input image frame at time step n, respectively. The luminance component Y takes real values in the range [0, 255] in an image, and the mean values of chrominance channels, U and V, are increased to 128 so that they also take values between 0 and 255. The threshold T_I is an experimentally determined value and taken as 128 on the luminance (Y) component in this work. The confidence level of $D_2(x,n)$ is -1 if $Y(x,n)$ is below T_I.

4.3 *Wavelet Transform-Based Region Smoothness Detection*

Wildfire smoke plumes soften the edges in image frames. Smoke-colored, slow moving regions are further analyzed using wavelet transform for this decision function. High frequency components in images produce large coefficients in wavelet domain.[45–48] Therefore, we can compare the

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

Fig. 4 Single-level wavelet decomposition of the snapshot image in Fig. 3 obtained using $h_{lp}[n] = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ and $h_{hp}[n] = \{-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\}$ as low and high-pass filters, respectively.

high-frequency wavelet energies of the current image and the background to confirm the existence of smoke. The main assumption is that the background has higher frequency components than smoke. This is a reasonable assumption for the wildfire detection system since the forest background usually has edgy features due to the fact that forestal areas are covered with tree foliage, leaves, branches, rocks, and other non-smooth surfaces. For illustrative purposes, a single-level wavelet decomposition of the snapshot image in Fig. 3 obtained using $h_{lp}[n] = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ and $h_{hp}[n] = \{-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\}$ as low- and high-pass filters, respectively, is given in Fig. 4.

The energy function that represents the high frequency content of the n'th image frame $I(n)$ is calculated as:

$$E_h[I(n)] = \sum_x |J_{LH}(n,x)| + \sum_x |J_{HL}(n,x)| + \sum_x |J_{HH}(n,x)|, \tag{16}$$

where $J_{LH}(n)$, $J_{HL}(n)$, and $J_{HH}(n)$ represent the horizontal, vertical, and detail sub-bands of a single stage wavelet transform of $I(n)$, respectively.

For the background image $B(n)$, the energy function is calculated as follows:

$$E_h[B(n)] = \sum_x D_{LH}(n,x) + \sum_x D_{HL}(n,x) + \sum_x D_{HH}(x), \tag{17}$$

where $D_{LH}(n)$, $D_{HL}(n)$, and $D_{HH}(n)$ represent the horizontal, vertical, and detail sub-bands of a single stage wavelet transform of $B(n)$, respectively.

The ratio between the energy functions of the background and current frames can be used to determine the likelihood of the region containing smoke:

$$\Delta_1(n) = \frac{E_h[B(n)]}{E_h[I(n)]}. \tag{18}$$

Since the smoke regions have low frequency characteristics, the low-low sub-band of the wavelet transform image should have the most energy. Therefore, the average energies of plume regions in the current frame and its corresponding LL sub-band image is expected to be close.

For a candidate smoke region Rs in the LL sub-band image, $J_{LL}(n)$, its average energy is given as follows:

$$E_{Rs}(n) = \frac{1}{N} \sum_{(x) \in Rs} |J_{LL}(n,x)|^2, \tag{19}$$

where N is the total number of pixels in Rs. Average energy of the corresponding region, Ro in the original image $I(n)$ is calculated as follows:

$$E_{Ro}(n) = \frac{1}{4N} \sum_{(x) \in Ro} |I(n,x)|^2, \tag{20}$$

where the scaling factor of 4 is used since the LL image is a quarter-size of the original image.

The candidate regions for which the difference between average energies is small are determined as smoke regions:

$$\Delta_2(n) = |E_{Rs,n} - E_{Ro,n}|. \tag{21}$$

The decision function $D_3(x,n)$ corresponding to this sub-algorithm is given as follows:

$$D_3(x,n) = \begin{cases} 2\Delta_1(n) - 1, & \text{if } \Delta_2(n) < T_{LL} \\ -1, & \text{else} \end{cases} \tag{22}$$

where T_{LL} is an experimentally determined threshold.

4.4 *Shadow Detection and Removal*

Shadows of slow moving clouds are a major source of false alarms for video-based wildfire smoke detection systems. Unfortunately, shadows of clouds have very low U and V values, similar to the smoke regions from wildfires.

The decision function for shadow regions are defined based on the shadow detection method described in Ref. 49. Average RGB values are calculated for slow moving regions both in the current and the background images. Let $S(n)$ represent a slow moving region in the image I at frame number n. The average color vector, $\vec{c_{I,S}}(n)$, of this region in the image I at frame number n is calculated as follows:

$$\vec{c_{I,S}}(n) = \frac{1}{A_{S(n)}} \left[\sum_{x \in S(n)} r_I(x,n), \sum_{x \in S(n)} g_I(x,n), \times \sum_{x \in S(n)} b_I(x,n) \right], \tag{23}$$

where $A_{S(n)}$ is the area of the slow moving region $S(n)$, and $r_I(x,n)$, $g_I(x,n)$, and $b_I(x,n)$ are the red, green, and blue channel values of the pixel at location x in the n'th image frame I. Similarly, average color vector, $\vec{c_{B,S}}$, of the same region in the background image, B, is calculated as follows:

$$\vec{c_{B,S}}(n) = \frac{1}{A_{S(n)}} \left[\sum_{x \in S(n)} r_B(x,n), \sum_{x \in S(n)} g_B(x,n), \times \sum_{x \in S(n)} b_B(x,n) \right], \tag{24}$$

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

where $r_B(x,n)$, $g_B(x,n)$, and $b_B(x,n)$ are the red, green, and blue channel values of the pixel at location x in the background image frame B at frame number n. We used the background image B^{slow} as the background image in our implementation.

In shadow regions, the angle, $\theta(x)$, between the average color vectors, $\vec{c_{I,S}}$ and $\vec{c_{B,S}}$, should be small and the magnitude of the vector in the current image should be smaller than that of the vector in the background image, i.e., $|\vec{c_{I,S}}(n)| < |\vec{c_{B,S}}(n)|$.[49] This is because shadow regions retain the color and the underlying texture to some extent.

The confidence value of this sub-algorithm is defined according to the angle and magnitudes of average color vectors, $\vec{c_{I,S}}(n)$ and $\vec{c_{B,S}}(n)$. The decision function $D_4(x,n)$ corresponding to this sub-algorithm for a pixel in the n'th image and background frames is given by:

$$D_4(x,n) = \begin{cases} \frac{4|\theta(x)|}{\pi} - 1, & \text{if } |\vec{c_{I,S}}(n)| > |\vec{c_{B,S}}(n)| \\ -1, & \text{if } |\vec{c_{I,S}}(n)| < |\vec{c_{B,S}}(n)| \end{cases}, \tag{25}$$

where $\theta(x)$ is the angle between the two color vectors. When the angle between the two color vectors are close to each other, the function $D_4(x,n)$ is close to -1 which corresponds to shadow regions. Similar decision functions for shadow detection can be defined according to other color spaces including the YUV space.

There are other shadow detection algorithms in the literature.[50] However, we selected the algorithm described in this section, because of its low computational complexity. Our aim is to realize a wildfire detection system working in real-time.

4.5 *Covariance Matrix-Based Classification*

The fifth sub-algorithm deals with the classification of the smoke-colored moving regions. A region covariance matrix consisting of discriminative features is calculated for each region.[51] For each pixel in the region, a nine-dimensional feature vector z_k is calculated as:

$$z_k = \left[x_1 \; x_2 \; Y(x_1,x_2) \; U(x_1,x_2) \; V(x_1,x_2) \; \left|\frac{dY(x_1,x_2)}{dx_1}\right| \; \left|\frac{dY(x_1,x_2)}{dx_2}\right| \; \left|\frac{d^2Y(x_1,x_2)}{dx_1^2}\right| \; \left|\frac{d^2Y(x_1,x_2)}{dx_2^2}\right| \right]^T, \tag{26}$$

where k is the label of a pixel, (x_1,x_2) is the location of the pixel, Y,U,V are the components of the representation of the pixel in YUV color space, $dY(x_1,x_2)/dx_1$ and $dY(x_1,x_2)/dx_2$ are the horizontal and vertical derivatives of the region, respectively, calculated using the filter $[-1\;0\;1]$, $d^2Y(x_1,x_2)/dx_1^2$ and $d^2Y(x_1,x_2)/dx_2^2$ are the horizontal and vertical second derivatives of the region calculated using the filter $[-1\;2\;-1]$.

The feature vector for each pixel can be defined as follows:

$$z_k = [z_k(i)]^T, \tag{27}$$

where i is the index of the feature vector. This feature vector is used to calculate the 9×9 covariance matrix of the regions using the fast covariance matrix computation formula:[52]

$$C_R = [c_R(i,j)] = \left\{ \frac{1}{n-1} \left[\sum_{k=1}^{n} z_k(i) z_k(j) - \frac{1}{n} \sum_{k=1}^{n} z_k(i) \sum_{k=1}^{n} z_k(j) \right] \right\}, \tag{28}$$

where n is the total number of pixels in the region and $c_R(i,j)$ is the (i,j) component of the covariance matrix.

The region covariance matrices are symmetric, therefore, we only need half of the elements of the matrix for classification. We also do not need the first 3 elements $c_R(1,1)$, $c_R(2,1)$, and $c_R(2,2)$ when using the lower diagonal elements of the matrix, because these are the same for all regions. Then, we need a feature vector f_R with $9\times10/2-3=42$ elements for each region. For a given region, the final feature vector does not depend on the number of pixels in the region, it only depends on the number of features in z_k.

A SVM with the RBF kernel is trained with the region covariance feature vectors of smoke regions in the training database. The LIBSVM (Ref. 53) software is used to obtain the posterior class probabilities, $p_R = Pr(\text{label}=1|f_R)$, where label $=1$ corresponds to a smoke region. In this software, posterior class probabilities are calculated by approximating the posteriors with a sigmoid function, as in Ref. 54. If the posterior probability is larger than 0.5, the label is 1 and the region contains smoke. The decision function for this sub-algorithm is defined as follows:

$$D_5(x,n) = 2p_R - 1, \tag{29}$$

where $0 < p_R < 1$ is the posterior probability that the region contains smoke.

The decision results of five sub-algorithms, D_1, D_2, D_3, D_4, and D_5 are linearly combined to reach a final decision on a given pixel whether it is a pixel of a smoke region or not. Morphological operations are applied to the detected pixels to mark the smoke regions. Specifically, we apply "opening" which is dilation followed by erosion to remove small noisy regions and enhance the larger regions.[55] The number of connected smoke pixels should be larger than a threshold to issue an alarm for the region. If a false alarm is issued during the training phase, the oracle gives feedback to the algorithm by declaring a no-smoke decision value ($y=-1$) for the false alarm region. The weights are updated using the correct classification results supplied by the oracle. Initially, equal weights are assigned to each sub-algorithm. There may be large variations between forestal areas, and substantial temporal changes may occur within the same forestal region. As a result, weights of individual sub-algorithms will evolve in a dynamic manner over time.

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

Fig. 5 A snapshot from an independent test of the system by the Regional Technology Clearing House of San Diego State University in California in April 2009. The system successfully detected the test fire and did not produce any false alarms. The detected smoke regions are marked with bounding rectangles.

In real-time operating mode, the PTZ cameras are in continuous scan mode visiting predefined preset locations. In this mode, constant monitoring from the oracle can be relaxed by adjusting the weights for each preset once, and then using the same weights for successive classifications. Since the main issue is to reduce false alarms, the weights can be updated when there is no smoke in the viewing range of each preset, and after that the system becomes autonomous. The cameras stop at each preset, and run the detection algorithm for some time before moving to the next preset. By calculating separate weights for each preset we are able to reduce false alarms.

5 Experimental Results

5.1 *Experiments on Wildfire Detection*

The proposed wildfire detection scheme with projection onto the convex set-based active learning method is implemented on a PC with an Intel Core 2 Duo CPU 2.66 GHz processor, and tested with forest surveillance recordings captured from cameras mounted on top of forest watch towers near Antalya and Mugla provinces in the Mediterranean region in Turkey. The weather is stable with sunny days throughout the entire summer in the Mediterranean. If it happens to rain, there is no possibility of forest fire. The installed system successfully detected three forest fires in the summer of 2008. The system was also independently tested by the Regional Technology Clearing House of San Diego State University in California in April 2009 and it detected the test fire and did not produce any false alarms. A snapshot from this test is presented in Fig. 5.

The proposed ADF strategy is compared with the weighted majority algorithm (WMA) Oza, of,[11,56] and one of our previous implementations.[57] The WMA is summarized in Algorithm 2.[11] In WMA, as opposed to our method, individual decision values from sub-algorithms are binary, i.e., $d_i(x,n) \in \{-1, 1\}$, which are simply the quantized version of real valued $D_i(x,n)$ defined in Sec. 4. In the WMA, the weights of sub-algorithms yielding contradictory decisions with that of the oracle are reduced by a factor of 2

Algorithm 2 The pseudo-code for the weighted majority algorithm.

Weighted Majority(x,n)

for $i = 1$ to M **do**

  $w_i(0) = \frac{1}{M}$, Initialization

end for

for $i = 1$ to M **do**

if $d_i(x,n) \neq y$ **then**

  $w_i(n+1) \leftarrow \frac{w_i(n)}{2}$

end if

end for

if $\sum_{i:d_i(x,n)=1} w_i(n) \geq \sum_{i:d_i(x,n)=-1} w_i(n)$ **then**

  return 1

else

  return -1

end if

in an uncontrolled manner, unlike the proposed ADF-based algorithm and the universal linear predictor (ULP) scheme. Initial weights for WMA are taken as $1/M$, as in the proposed ADF-based scheme.

The ADF-based scheme, the WMA-based scheme, the non-adaptive approach with fixed weights, and our previous method[57] are compared in the following experiments. In Tables 1 and 2, forest surveillance recordings containing actual forest fires and test fires, as well as video sequences with no fires, are used. In Tables 1 and 2, the true detection rate in a given video clip is defined as the number of correctly classified frames containing smoke divided by the total number of frames that contain smoke. Similarly, the false alarm rate in a given test video is defined as the number of misclassified

Fig. 6 Cameras used in wildfire detection system.

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

Table 1 The ADF method is compared with WMA, non-adaptive method and the method developed in Ref. 57 in terms of true detection rates in video clips that contain wildfire smoke.

		True detection rates				First alarm frame/time (S)			
Video	Frames	ADF	WMA	Fixed	OLD	ADF	WMA	Fixed	OLD
V1	788	85.40%	85.65%	85.40%	74.11%	85/17.00	85/17.00	85/17.00	144/28.80
V2	268	77.23%	79.10%	78.35%	23.88%	53/7.57	48/6.86	50/7.14	147/21.00
V3	439	80.41%	81.09%	80.86%	14.57%	23/4.60	22/4.40	22/4.40	58/11.60
V4	800	83.62%	82.50%	82.62%	27.37%	78/3.12	78/3.12	78/3.12	126/5.04
V5	600	56.00%	55.78%	55.57%	72.81%	45/5.00	45/5.00	45/5.00	187/20.78
V6	900	79.22%	84.88%	90.22%	36.55%	68/2.72	61/2.44	68/2.72	251/10.04
V7*	2783	93.89%	97.02%	97.05%	76.03%	51/10.20	34/6.80	34/6.80	77/15.40
V8*	1000	74.00%	79.40%	74.90%	94.90%	74/14.80	36/7.20	36/7.20	51/10.20
V9*	329	83.28%$	87.23%	85.41%	80.85%	54/10.80	41/8.20	43/8.60	58/11.60
V10	800	46.87%$	48.87%	48.37%	26.75%	18/3.60	3/0.60	7/1.40	290/58.00
V11	1450	73.51%	75.10%	72.41%	53.10%	139/27.80	139/27.80	139/27.80	15/3.00
V12*	1500	94.00%	93.93%	96.06%	79.73%	52/10.40	26/5.20	26/5.20	51/10.20
V13*	1000	94.60%	97.30%	96.50%	95.50%	54/10.80	28/5.60	33/6.60	52/10.40
Average	-	78.61%	80.60%	80.28%	59.70%	61.07/9.87	49.69/7.70	51.23/7.92	115.92/ 16.61

frames, which do not contain smoke divided by the total number of frames that do not contain smoke.

We have 5 actual forest fire videos and 8 test fire videos ranging from 2 to 6 km captured in Antalya and Mugla provinces in the Mediterranean region in Turkey, in the summers between 2007 and 2009. In Fig. 6, some of the cameras that are used to record the videos are shown. The clips marked with "*" are actual forest fire videos. All of the above-mentioned decision fusion methods detected forest fires within 10 s on the average, as shown in Table 1. The OLD method, previously developed by authors in Ref. 57, usually has a higher first detection time. The detection rates of the methods are comparable to each other. Compared to the previous method, the ADF method has a higher true detection rate in most of the video clips that contain actual smoke plumes. Although the true detection rate is low in some videos, we do not need to detect all smoke frames correctly to issue an alarm. It is enough to

Table 2 The ADF method is compared with WMA, non-adaptive method, and the method developed in Ref. 57 in terms of false alarm rates in video clips that do not contain wildfire smoke.

	False alarm rates			
Video name	ADF	WMA	Fixed	OLD
V14	$\frac{24}{6300} = 0.38\%$	$\frac{70}{6300} = 1.11\%$	$\frac{51}{6300} = 0.81\%$	$\frac{331}{6300} = 5.25\%$
V15	$\frac{147}{3370} = 4.36\%$	$\frac{199}{3370} = 5.91\%$	$\frac{398}{3370} = 11.81\%$	$\frac{256}{3370} = 7.60\%$
V16	$\frac{55}{1839} = 2.99\%$	$\frac{57}{1839} = 3.10\%$	$\frac{106}{1839} = 5.76\%$	$\frac{140}{1839} = 7.61\%$
V17	$\frac{322}{6294} = 5.12\%$	$\frac{881}{6294} = 14.00\%$	$\frac{2109}{6294} = 33.51\%$	$\frac{871}{6294} = 13.84\%$
V18	$\frac{8}{3005} = 0.27\%$	$\frac{1}{3005} = 0.03\%$	$\frac{8}{3005} = 0.27\%$	$\frac{1368}{3005} = 45.52\%$
V19	$\frac{0}{3478} = 0.00\%$	$\frac{0}{3478} = 0.00\%$	$\frac{0}{3478} = 0.00\%$	$\frac{1796}{3478} = 51.64\%$
V20	$\frac{1}{3462} = 0.03\%$	$\frac{1}{3462} = 0.03\%$	$\frac{2}{3462} = 0.06\%$	$\frac{1528}{3462} = 44.14\%$
Average	1.88%	3.45%	7.46%	25.08%

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

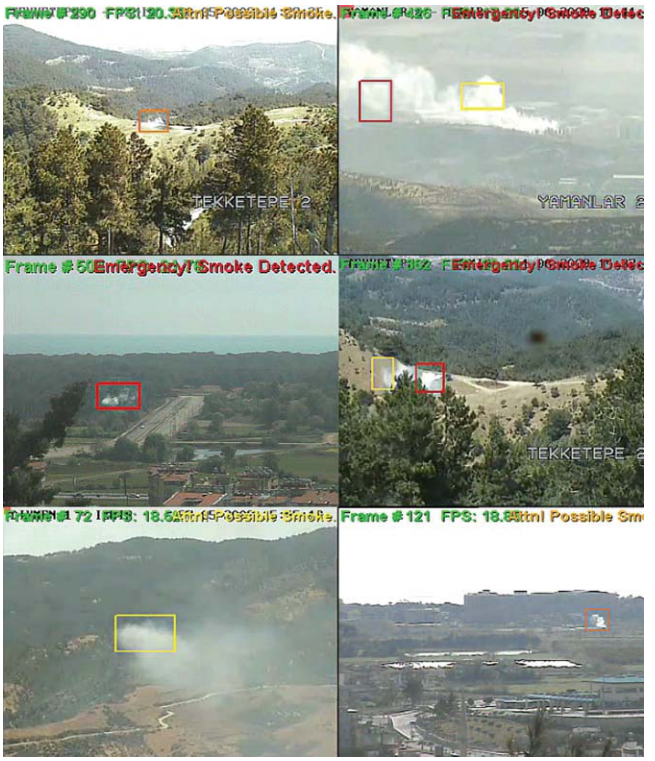


Fig. 7 Snapshots from the videos in Table 1.

detect smoke in a short time without too many false alarms. In Fig. 7 some snapshots from the videos in Table 1 are displayed.

On the other hand, the proposed adaptive fusion strategy significantly reduces the false alarm rate of the system by integrating the feedback from the guard (oracle) into the decision mechanism within the active learning framework described in Sec. 2.

The proposed method produces the lowest number of false alarms in our data set. A set of video clips containing clouds, cloud shadows, and other false alarm sources is used to generate Table 2. These video clips were selected on purpose to compare the performance of various methods. False alarm rates of different methods are presented in Table 2. The average percentage of false alarms for the methods (a) the ADF-based scheme, (b) the WMA-based scheme, (c) the nonadaptive approach with fixed weights, and (d) our OLD method, are 1.88%, 3.45%, 7.46%, and 25.08%, respectively.

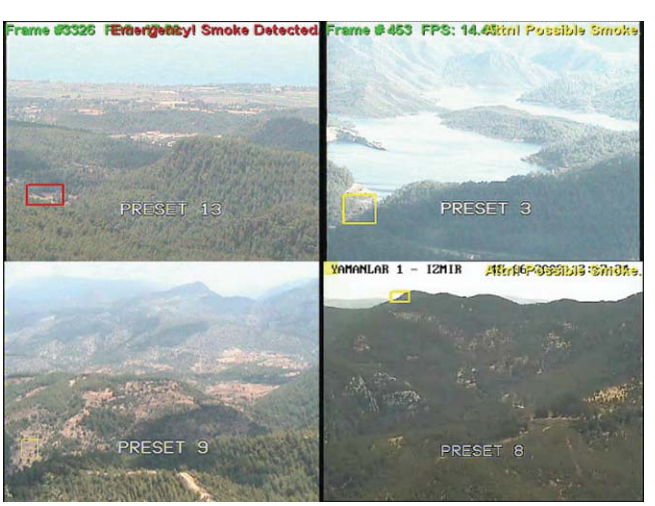


Fig. 8 Typical false alarms issued to clouds and cloud shadows.

As shown in Fig. 8, the main sources of false alarms that we observed are clouds, cloud shadows, and shaking cameras in the wind. By using support vector machines and the ADF algorithm, false alarms are significantly reduced. The system is currently being used in 57 forest watch towers in Turkey.

6 Conclusion

A general framework for online ADF is proposed to be used especially for image analysis and computer vision applications with drifting concepts. In this framework, it is assumed that the main algorithm for a specific application is composed of several sub-algorithms, each of which yielding its own decision as a real number centered around zero, representing its confidence level. Decision values are linearly combined with weights which are updated online, by performing orthogonal projections onto convex sets describing sub-algorithms. This general framework is applied to a real computer vision problem of wild-fire detection. The proposed adaptive decision fusion strategy takes into account the feedback from guards of forest watch towers. Experimental results show that the learning duration is decreased with the proposed online adaptive fusion scheme based on making orthogonal projections onto hyperplanes defined by update weights. It is also observed that the false alarm rate of the proposed method is the lowest in our data set, compared to the ULP and WMA-based schemes.

The proposed framework for decision fusion is suitable for problems with concept drift. At each stage of the algorithm, the method tracks the changes in the nature of the problem by performing an orthogonal projection onto a hyperplane describing the decision of the oracle.

Acknowledgments

This work was supported in part by the Scientific and Technical Research Council of Turkey, TUBITAK, with Grant Nos. 106G126 and 105E191, and in part by European Commission 7th Framework Program with Grant No. FP7-ENV-2009-1-244088-FIRESENSE.

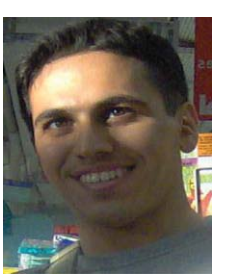
References

1. B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proc. IEEE* **67**(5), 708–713 (1979).
2. T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Tran. Pattern Anal. Machine Intell.* **16**(1), 66–75 (1994).
3. V. I. Gorodetskiy and S. V. Serebryakov, "Methods and algorithms of collective recognition," *Automation Remote Control* **69**(11), 1821–1851 (2008).
4. A. Kumar and D. Zhang, "Personal authentication using multiple palmprint representation," *Pattern Recog.* **38**(10), 1695–1704 (2005).
5. X. Tang and Z. Li, "Video based face recognition using multiple classifiers," in *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, 17–19 May 2004.
6. M. A. Garcia and D. Puig, "Supervised texture classification by integration of multiple texture methods and evaluation windows," *Image Visi. Comput.* **25**(7), 1091–1106 (2007).
7. N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inf. Comput.* **108**, 212261 (1994).
8. L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst. Man, Cybern.* **22**(3), 418–435 (1992).

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

9. L. K. Kuncheva, "Switching between selection and fusion in combining classifiers: an experiment," *IEEE Trans. Sys., Man, Cybern. Part B: Cybern.* **32**(2), 146–156 (2002).
10. D. Parikh and R. Polikar, "An ensemble-based incremental learning approach to data fusion," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.* **37**(2), 437–450 (2007).
11. N. Oza, "Online ensemble learning," PhD Dissertation, University of California, Berkeley (2002).
12. A. Enis Cetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Trans. Signal Process.* **42–1**, 194–196 (1994).
13. P. L. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE* **81**(2), 182–208 (1993).
14. D. C. Youla and H. Webb, "Image restoration by the method of convex projections, part i-theory," *IEEE Trans. Med. Imaging*, **MI-I-2**, 81–94 (1982).
15. E. Cetin, "Reconstruction of signals from fourier transform samples," *Signal Processing*, **16**, 129–148 (1989).
16. I. Yamada, K. Slavakis, and S. Theodoridis, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.* **56**, 2781–2796, (2008).
17. U. Niesen, D. Shah, and G. W. Wornell, "Adaptive alternating minimization algorithms," *IEEE Trans. Inform. Theory*, **55**(3), 1423–1429, (2009).
18. S. Theodoridis and M. Mavroforakis, "Reduced convex hulls: A geometric approach to support vector machines," *IEEE Signal Process. Mag.* **24**, 119–122 (2007).
19. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Orlando, FL (2006).
20. J. C. Schlimmer and R. H. Granger, Jr., "Incremental learning from noisy data," *Mach. Learn.* **1**, 317–354, (1986).
21. M. Karnick, M. Ahiskali, M. D. Muhlbaier, and R. Polikar, "Learning concept drift in nonstationary environments using an ensemble of classifiers based approach," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 3455–3462 (2008).
22. K. Nishida, S. Shimada, S. Ishikawa, and K. Yamauchi, "Detecting sudden concept drift with knowledge of human behavior," in *IEEE International Conference on Systems, Man and Cybernetics, 2008. SMC 2008*, 12–15 Oct. 2008, pp. 3261–3267 (2008).
23. M. de Dios et al., "Computer vision techniques for forest fire perception," *Image Vis. Comput.* **26**(4), 550–562, (2008).
24. J. Li, Q. Qi, X. Zou, H. Peng, L. Jiang, and Y. Liang, "Technique for automatic forest fire surveillance using visible light image," in *2005 IEEE International Geoscience and Remote Sensing Symposium, IGARSS '05*, 25–29 July 2005, Vol. 5.
25. I. Bosch, S. Gomez, L. Vergara, and J. Moragues, "Infrared image processing and its application to forest fire surveillance," in *IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS, 2007*, 5–7 Sept. 2007, pp. 283–288 (2007).
26. P. Guillemant and J. Vicente, "Real-time identification of smoke images by clustering motions on a fractal curve with a temporal embedding method," *Opt. Eng.* **40**(4), 554–563, (2001).
27. M. Hefeeda and M. Bagheri, "Wireless sensor networks for early detection of forest fires," in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, 2007, MASS 2007*, 8–11 Oct. 2007, pp. 1–6 (2007).
28. Y. G. Sahin, "Animals as mobile biological sensors for forest fire detection," *Sensors* **7**(12), 3084–3099, (2007).
29. S. Chen, H. Bao, X. Zeng, and Y. Yang, "A fire detecting method based on multi-sensor data fusion," in *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 14, 3775–3780 (2003).
30. P. Podrzaj and H. Hashimoto, "Intelligent space as a fire detection system," in *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*, 8–11 Oct. 2006, pp. 2240–2244 (2006).
31. B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Flame detection in video using hidden markov models," in *IEEE International Conference on Image Processing, 2005, ICIP 2005*, 11–14 Sept. 2005, Vol. 2, pp. 1230–1233 (2005).
32. Y. Dedeoglu, B. U. Toreyin, U. Gudukbay, and A. E. Cetin, "Real-time fire and flame detection in video," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, Proceedings, (ICASSP '05)*, March 18–23, 2005, Vol. 2, pp. 669–672 (2005).
33. W. Da-Jinn, C. Thou-Ho (Chao-Ho), Y. Yen-Hui, and C. Tsong-Yi, "Smoke detection for early fire-alarming system based on video processing," *J. Digital Info. Management*, **6**(2), 196–202, (2008).
34. B. U. Toreyin, Y. Dedeoglu, U. Gudukbay, and A. E. Cetin, "Computer vision based system for real-time fire and flame detection," *Pattern Recogn. Lett.* **27**, 49–58 (2006).
35. B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "Wavelet based real-time smoke detection in video," in *13th EUSIPCO*, (2005).
36. T. Pavlidis, "Computers vs humans," http://www.theopavlidis.com/comphumans/comphuman.htm, 31 May 2011.
37. H. L. Dreyfus, *What Computers Can't Do*, Harper & Row, New York, 1972.
38. H. L. Dreyfus, *What Computers Still Can't Do : a critique of artificial reason*, MIT Press, 1992.
39. A. E. Cetin, M. B. Akhan, B. U. Toreyin, and A. Aksay, "Characterization of motion of moving objects in video," U.S. Patent No. 20,040,223,652, (2004).
40. F. Porikli, Y. Ivanov, and T. Haga, "Robust abandoned object detection using dual foregrounds," *EURASIP J. Appl. Signal Process*, **2008**(1), 1–10, (2008).
41. R. T. Collins, A. J. Lipton, and T. Kanade, "A system for video surveillance and monitoring," in *8th Int. Topical Meeting on Robotics and Remote Systems*, American Nuclear Society (1999).
42. B. U. Toreyin, "Moving object detection and tracking in wavelet compressed video," MS Thesis, Bilkent University, Ankara, Turkey (2003).
43. C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6 August 1999, Vol. 2 (1999).
44. H. Ammann et al., "Wildfire smoke - a guide for public health officials," http://depts.washington.edu/wildfire/PubHlthGuidev.9.0.pdf, (2001).
45. A. E. Cetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Trans. Signal Process.* **42**, 194–196 (1994).
46. S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analy. Machine Intell.* **14**(7), 710–732 (1992).
47. A. Aksay, A. Temizel, and A. Enis Cetin, "Camera tamper detection using wavelet analysis for video surveillance," in *AVSS '07: Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, Washington, DC, pp. 558–562, IEEE Computer Society (2007).
48. B. U. Toreyin and A. E. Cetin, "Volatile organic compound plume detection using wavelet analysis of video," in *15th IEEE International Conference on Image Processing*, pp. 1836–1839 (2008).
49. T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *IEEE ICCV'99 Frame-Rate Workshop*, Corfu, Greece, Sep. 1999 (1999).
50. A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Trans. Pattern Anal. Machine Intell.* **25**, 918–923, 2003.
51. O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. 9th European Conf. on Computer Vision*, Graz, Austria, Vol. 2, pp. 589–600 (2006).
52. F. Porikli and O. Tuzel, "Fast construction of covariance matrices for arbitrary size image windows," in *2006 IEEE International Conference on Image Processing*, 8–11 Oct. 2006, pp. 1581–1584 (2006).
53. C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm (2001).
54. J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, pp. 61–74, MIT Press, Cambridge, MA (1999).
55. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Englewood Cliffs, NJ (2002).
56. N. C. Oza, "Online bagging and boosting," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, 10–12 Oct. 2005, Vol. 3, pp. 2340–2345 (2005).
57. B. U. Töreyin and A. E. Çetin, "Computer vision based forest fire detection," in *IEEE 16th Signal Processing, Communication and Applications Conference, 2008, SIU 2008*, 20–22 April 2008.

Osman Günay received his BS degree in 2007 and his MS degree in 2009, from the Electrical and Electronics Engineering Department of Bilkent University, Ankara, Turkey, where he is currently a PhD student and a research and teaching assistant. His research interests are in the area of signal processing, with an emphasis on image and video processing, pattern recognition, and computer vision.



Behcet Uğur Töreyin received his PhD and MS degrees from Bilkent University and BS degree from Middle East Technical University, Ankara, Turkey, all in electrical and electronics engineering. Between 2009 and 2010, he was a postdoctoral associate at the Robotic Sensor Networks Lab, University of Minnesota, Minnesota. Currently, he is a postdoctoral associate at the Wireless Research Lab, Texas A&M University at Qatar.

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms

Ahmet Enis Çetin received a BS degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, and MSE and PhD degrees in systems engineering from the Moore School of Electrical Engineering, University of Pennsylvania, Pennsylvania. From 1987 to 1989, he was an assistant professor of Electrical Engineering at the University of Toronto, Toronto, Ontario, Canada. Since then, he has been with Bilkent University, Ankara. Currently, he is a full professor. During the summers of 1988, 1991, and 1992, he was with Bell Communications Research (Bellcore) as a consultant. He spent the 1996 and 1997 academic years at the University of Minnesota, Minneapolis, as a visiting professor. He carried out contract research for both governmental agencies and industry, including Visioprime, U.K.; Honeywell Video Systems, Grandeye, U.K.; the National Science Foundation; NSERC, Canada; and ASELSAN. He is a senior member of EURASIP. He founded the Turkish Chapter of the IEEE Signal Processing Society in 1991. He was Signal Processing and AES Chapter Coordinator of IEEE Region-8 in 2003. He was the Co-Chair of the IEEE-EURASIP Nonlinear Signal and Image Processing Workshop, held in 1999 in Antalya, Turkey, and the technical Co-Chair of the European Signal Processing Conference (EUSIPCO) in 2005. He received the Young Scientist Award from the Turkish Scientific and Technical Research Council (TUBITAK) in 1993. He is a fellow of IEEE. He was an associate editor of the IEEE Transactions on Image Processing between 1999 and 2003, and a member of the SPTM technical committee of the IEEE Signal Processing Society. He is currently on the editorial boards of EURASIP Journal of Applied Signal Processing, Signal Processing and Journal of Advances in Signal Processing (JASP).

Downloaded From: http://opticalengineering.spiedigitallibrary.org/ on 11/26/2012 Terms of Use: http://spiedl.org/terms