

# Entropy-Functional-Based Online Adaptive Decision Fusion Framework With Application to Wildfire Detection in Video

Osman Gunay, Behçet Ugur Toreyin, Kivanc Kose, and A. Enis Cetin, *Fellow, IEEE*

**Abstract**—In this paper, an entropy-functional-based online adaptive decision fusion (EADF) framework is developed for image analysis and computer vision applications. In this framework, it is assumed that the compound algorithm consists of several subalgorithms, each of which yields its own decision as a real number centered around zero, representing the confidence level of that particular subalgorithm. Decision values are linearly combined with weights that are updated online according to an active fusion method based on performing entropic projections onto convex sets describing subalgorithms. 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 was developed to evaluate the performance of the decision fusion algorithm. In this case, image data arrive sequentially, and the oracle is the security guard of the forest lookout tower, verifying the decision of the combined algorithm. The simulation results are presented.

**Index Terms**—Active learning, decision fusion, entropy maximization, online learning, projections onto convex sets, wildfire detection using video.

## I. INTRODUCTION

IN THIS paper, an online learning framework, called entropy-functional-based online adaptive decision fusion (EADF), which can be used in various image analysis and computer vision applications, is proposed. In this framework, it is assumed that the compound algorithm consists of several subalgorithms, each of which yields its own decision. The final decision is reached based on a set of real numbers representing confidence levels of various subalgorithms. Decision values are linearly combined with weights that are updated online using an active fusion method based on performing entropic projections (e-projections) onto convex sets describing the subalgorithms.

Manuscript received January 23, 2011; revised May 06, 2011 and November 12, 2011; accepted December 14, 2011. Date of publication January 09, 2012; date of current version April 18, 2012. This work was supported in part by the Scientific and Technical Research Council of Turkey (TUBITAK) under Grant 111E057 and Grant 105E191, by the European Commission 7th Framework Program under Grant FP7-ENV-2009-1244088 FIRESENSE (Fire Detection and Management through a Multi-Sensor Network for the Protection of Cultural Heritage Areas from the Risk of Fire and Extreme Weather Conditions). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Arun A. Ross.

O. Gunay, K. Kose, and A. E. Cetin are with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: osman@ee.bilkent.edu.tr; kkivanc@ee.bilkent.edu.tr; cetin@bilkent.edu.tr).

B. U. Toreyin is with the Department of Electronic and Communication Engineering, Çankaya University, 06530 Ankara, Turkey (e-mail: toreyin@cankaya.edu.tr).

Adaptive learning methods based on orthogonal projections are successfully used in some computer vision and pattern recognition problems [1], [2]. A multiple-classifier system is useful for difficult pattern recognition problems, particularly when large class sets and noisy data are involved, by allowing the use of arbitrary feature descriptors and classification procedures at the same time [3]. Instead of determining the weights using orthogonal projections as in [1] and [2], we introduce the e-projection approach that is based on a generalized projection onto a convex set.

The studies in the field of collective recognition, which were started in the mid1950s, found wide application in practice during the last decade, leading to solutions to complex large-scale applied problems [4]. One of the first examples of the use of multiple classifiers was given by Dasarthy and Sheela in [5] in which they introduced the concept of composite classifier systems as a means of achieving improved recognition system performance compared with employing the classifier components individually. The method is illustrated by studying the case of the linear/nearest neighbor (NN) 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 achieved better performance than either one individually [6]. A multiple-classifier fusion algorithm is proposed for developing an effective video-based face recognition method [7]. 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 multisized windows [8]. This 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 sizes.

In this paper, the EADF framework is applied to a computer-vision-based wildfire detection problem. The system based on this method is currently being used in more than 60 forest-fire lookout towers in the Mediterranean region. The proposed automatic video-based wildfire detection algorithm is based on five subalgorithms: 1) slow moving video object detection; 2) smoke-colored region detection; 3) wavelet-transform-based region smoothness detection; 4) shadow detection and elimination; and 5) covariance-matrix-based classification. Each subalgorithm separately decides on the existence of smoke in the viewing range of the camera. Decisions from subalgorithms are combined with the adaptive decision fusion (ADF) method. Initial weights of the subalgorithms are determined from actual forest-fire videos and test fires. They are updated by using e-pro-

jections 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 a security guard. Whenever a fire is detected, 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 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; then, the system can be run without feedback.

This paper is organized as follows. The EADF framework is described in Section II. The first part of this section describes our previous weight update algorithm, which is obtained by orthogonal projections onto hyperplanes [1]; the second part proposes an entropy-based e-projection method for a weight update of the subalgorithms. Section III introduces the video-based wildfire detection problem. In Section IV, each one of the five subalgorithms, which make up the compound (main) wildfire detection algorithm, are described. In Section V, experimental results are presented, and the proposed online active fusion method is compared with the universal linear predictor (ULP) and the weighted majority algorithms. 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. The proposed EADF method is also evaluated on a data set from the University of California Irvine (UCI) machine learning repository [9]. Well-known classifiers (e.g., support vector machines (SVMs) and k-NN) are combined using EADF. During the training stage, individual decisions of classifiers are used to find the weight of each classifier in the composite EADF classifier. Finally, conclusions are drawn in Section VI.

## II. ADF FRAMEWORK

Let the compound algorithm be composed of  $M$ -many detection subalgorithms:  $D_1, \dots, D_M$ . Upon receiving a sample input  $x$  at time step  $n$ , each subalgorithm 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 subalgorithm. 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 subalgorithm of the computer vision problem. For example, in the wildfire detection problem presented in Section III, the number of subalgorithms 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  $\mathbf{D}(x, n) = [D_1(x, n), \dots, D_M(x, n)]^T$  be the vector of decision values of the subalgorithms for the pixel at location  $x$  of input image frame at time step  $n$ , and  $\mathbf{w}(x, n) = [w_1(x, n), \dots, w_M(x, n)]^T$  be the current weight vector. For simplicity, we will drop  $x$  in  $\mathbf{w}(x, n)$  for the rest of this paper.

We define

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

as an estimate of the correct classification result  $y(x, n)$  of the oracle for the pixel at location  $x$  of the input image frame at time step  $n$  and error  $e(x, n)$  as  $e(x, n) = y(x, n) - \hat{y}(x, n)$ . As shown in the next subsection, the main advantage of the proposed algorithm compared with other related methods in [10]–[12] is the controlled feedback mechanism based on the error term. Weights of the algorithms producing an incorrect (correct) decision is iteratively reduced (increased) at each time step. Another advantage of the proposed algorithm is that it does not assume any specific probability distribution about the data.

### A. Set Theoretic Weight Update Algorithm Based on Orthogonal Projections

In this subsection, we first review the orthogonal-projection-based weight update scheme [1]. Ideally, weighted decision values of subalgorithms should be equal to the decision value of  $y(x, n)$  and the oracle as follows:

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

which represents a hyperplane in the  $M$ -dimensional space  $\mathbb{R}^M$ . Hyperplanes are closed and convex in  $\mathbb{R}^M$ . At time instant  $n$ ,  $\mathbf{D}^T(x, n)\mathbf{w}(n)$  may not be equal to  $y(x, n)$ . In our approach, the next set of weights are determined by projecting the current weight vector  $\mathbf{w}(n)$  onto the hyperplane represented by (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  $\mathbf{w}(n)$ .

Let us formulate the problem as a minimization problem, i.e.,

$$\min_{\mathbf{w}^*} \|\mathbf{w}^* - \mathbf{w}(n)\|_2 \text{ subject to } \mathbf{D}^T(x, n)\mathbf{w}^* = y(x, n). \quad (3)$$

The solution can be obtained by using Lagrange multipliers. The solution is called the metric projection mapping solution. However, we use the term orthogonal projection because the line going through  $\mathbf{w}^*$  and  $\mathbf{w}(n)$  is orthogonal to the hyperplane. If we define the next set of weights as  $\mathbf{w}(n+1) = \mathbf{w}^*$ , it can be obtained by the following iteration:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(x, n)}{\|\mathbf{D}(x, n)\|_2^2} \mathbf{D}(x, n) \quad (4)$$

Hence, the projection vector is calculated according to (4). Note that (4) is identical to the normalized least mean-square (NLMS) algorithm with the update parameter  $\mu = 1$ . In the NLMS algorithm,  $0 < \mu < 2$  should be satisfied for convergence [13]. According to the projection onto convex sets (POCS) theory, when there are a finite number of convex sets, repeated cyclical projections onto these sets converge to a vector in the intersection set [14]–[18]. The case of an infinite number of convex sets is studied in [2], [19], and [20]. They propose to use the convex combination of the projections onto the most recent  $q$  sets for online adaptive algorithms [2]. In Section II-C, the block projection version of the algorithm that deals with the case when there are an infinite number of convex sets is presented.

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

$$y(x, n+1) = \mathbf{D}^T(x, n+1)\mathbf{w}^*. \quad (5)$$

This hyperplane will not be the same as the  $y(x, n) = \mathbf{D}^T(x, n)\mathbf{w}(n)$  hyperplane in general. The next set of weights, i.e.,  $\mathbf{w}(n+2)$ , is determined by projecting  $\mathbf{w}(n+1)$  onto the hyperplane in (5). When there are a finite number of hyperplanes, iterated weights that are obtained by cyclic projections onto these hyperplanes converge to the intersection of hyperplanes [14], [21], [22].

The pseudocode of the orthogonal projections onto the hyperplane-based algorithm is given in Algorithm 1, which summarizes the projection onto one hyperplane. The block diagram of the algorithm for wildfire detection problem is shown in Fig. 4. The weights are initialized before the first sample arrives. Then, for each incoming sample, the orthogonal projection algorithm is performed to find the new set of weights. The weights are adjusted so that their sum is 1. The estimated output  $\hat{y}(x, n)$  is passed through a nonlinear function to find the classification result for the current sample.

The relation between SVMs and orthogonal projections onto half-planes was established in [17], [23], and [24]. As pointed out in [23], the SVM is very successful in batch settings, but it cannot handle online problems with drifting concepts in which the data arrive sequentially.

---

**Algorithm 1** The pseudocode for the POCS-based algorithm

---

```

for  $i = 1$  to  $M$  do
     $w_i(0) = 1/M$ , Initialization
end for

For each sample at time step  $n$ .
 $e(x, n) = y(x, n) - \hat{y}(x, n)$ 
for  $i = 1$  to  $M$  do
     $w_i(n) \leftarrow w_i(n) + \mu(e(x, n)/\|\mathbf{D}(x, n)\|_2^2)D_i(x, n)$ 
end for
for  $i = 1$  to  $M$  do
     $w_i(n) \leftarrow w_i(n)/\sum_j w_j(n)$ 
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

```

---

**B. E-Projection-Based Weight Update Algorithm**

The  $l_1$ -norm-based minimization approaches provide successful signal reconstruction results in compressive sensing

problems [25]–[28]. However, the  $l_0$ - and  $l_1$ -norm-based cost functions used in compressive sensing problems are not differentiable everywhere. The entropy functional approximates the  $l_1$ -norm  $\sum_i |w_i(n)|$  for  $w_i(n) > 0$  [29]. Therefore, it can be used to find approximate solutions to the inverse problems defined in [25] and [26] and other applications requiring  $l_1$ -norm minimization. Bregman developed convex optimization algorithms in the 1960s, and his algorithms are widely used in many signal reconstruction and inverse problems [2], [15], [30]–[33]. Bregman's method provides globally convergent iterative algorithms for problems with convex, continuous, and differentiable cost functionals  $g(\cdot)$  as follows:

$$\min_{\mathbf{w} \in C} g(\mathbf{w}) \quad (6)$$

such that

$$\mathbf{D}^T(x, n)\mathbf{w}(n) = y \text{ for each time index } n. \quad (7)$$

In the EADF framework, the cost function is  $g(\mathbf{w}) = \sum_i^M w_i(n) \log(w_i(n))$ , and each equation in (7) represents a hyperplane  $H(x, n) \in \mathbb{R}^M$ , which is a closed and convex set. In Bregman's method, the iterative algorithm starts with an arbitrary initial estimate, and successive e-projections are performed onto the hyperplanes  $H(x, n)$ ,  $n = 1, 2, \dots, N$  in each step of the iterative algorithm in a cyclic manner. In this case, we may have infinitely many hyperplanes, but we will still use Bregman's e-projection approach.

The e-projection onto a closed and convex set is a generalized version of the metric projection mapping onto a convex set [29]. Let  $\mathbf{w}(n)$  denote the weight vector for the  $n_{\text{th}}$  sample. Its e-projection  $\mathbf{w}^*$  onto a closed convex set  $C$  with respect to a cost functional  $g(\mathbf{w})$  is defined as follows:

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in C} L(\mathbf{w}, \mathbf{w}(n)) \quad (8)$$

where

$$L(\mathbf{w}, \mathbf{w}(n)) = g(\mathbf{w}) - g(\mathbf{w}(n)) - \langle \nabla g(\mathbf{w}), \mathbf{w} - \mathbf{w}(n) \rangle \quad (9)$$

and  $\langle \cdot, \cdot \rangle$  represents the inner product.

In the adaptive learning problem, we have a hyperplane  $H(x, n) : \mathbf{D}^T(x, n)\mathbf{w}(n+1) = y(x, n)$  for each sample  $x$ . For each hyperplane  $H(x, n)$ , the e-projection (8) is equivalent to

$$\nabla g(\mathbf{w}(n+1)) = \nabla g(\mathbf{w}(n)) + \lambda \mathbf{D}(x, n) \quad (10)$$

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

where  $\lambda$  is the Lagrange multiplier. As pointed out earlier, the e-projection is a generalization of the metric projection mapping. When the cost functional is the Euclidean cost functional  $g(\mathbf{w}) = \sum_i w_i(n)^2$ , distance  $L(\mathbf{w}_1, \mathbf{w}_2)$  becomes the  $l_2$  norm square of the difference vector  $(\mathbf{w}_1 - \mathbf{w}_2)$ , and the e-projection simply becomes the well-known orthogonal projection onto a hyperplane.

When the cost functional is the entropy functional  $g(\mathbf{w}) = \sum_i w_i(n) \log(w_i(n))$ , the e-projection onto the hyperplane  $H(x, n)$  leads to the following update equations:

$$w_i(n+1) = w_i(n)e^{\lambda D_i(x, n)}, \quad i = 1, 2, \dots, M \quad (12)$$

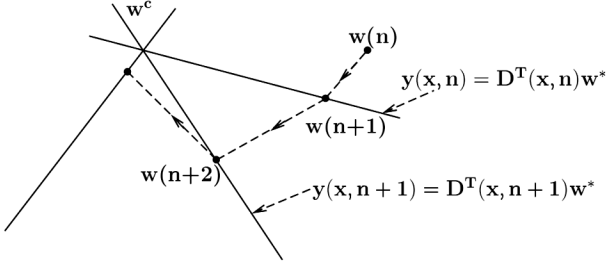


Fig. 1. Geometric interpretation of the e-projection method. Weight vectors corresponding to decision functions at each frame are updated to satisfy the hyperplane equations defined by the oracle's decision  $y(x, n)$  and the decision vector  $\mathbf{D}(x, n)$ . Lines in the figure represent hyperplanes in  $\mathbb{R}^M$ . Weight update vectors converge to the intersection of the hyperplanes. Notice that e-projections are not orthogonal projections.

where the Lagrange multiplier  $\lambda$  is obtained by inserting (12) into the hyperplane equation

$$\mathbf{D}^T(x, n)\mathbf{w}(n+1) = y(x, n) \quad (13)$$

because the e-projection  $\mathbf{w}(n+1)$  must be on the hyperplane  $H(x, n)$  in (11). When there are three hyperplanes, one cycle of the projection algorithm is depicted in Fig. 1. If the projections are continued in a cyclic manner, the weights will converge to the intersection of the hyperplanes, i.e.,  $w_c$ .

The earlier set of equations are used in signal reconstruction from Fourier transform samples and the tomographic reconstruction problem [16], [30]. The entropy functional is defined only for positive real numbers, which coincides with our positive weight assumption.

To find the value of  $\lambda$  at each iteration, a nonlinear equation has to be solved [see (12) and (13)]. In [34], globally convergent algorithms are developed without finding the exact value of the Lagrange multiplier  $\lambda$ . However, the tracking performance of the algorithm is very important. Weights have to be rapidly updated according to the oracle's decision.

In our application, we first use the second-order Taylor series approximation of  $e^{\lambda D_i(x, n)}$  from (12) and obtain

$$w_i(n+1) \approx w_i(n) \left( 1 + \hat{\lambda} D_i(x, n) + \frac{\hat{\lambda}^2 D_i^2(x, n)}{2} \right), \quad i = 1, 2, \dots, M \quad (14)$$

By multiplying both sides by  $D_i(x, n)$ , by summing over  $i$ , and by using (13), we get the following equation:

$$y(x, n) \approx \left( \sum_{i=1}^M D_i(x, n) w_i(n) + \hat{\lambda} \sum_{i=1}^M D_i^2(x, n) w_i(n) + \hat{\lambda}^2 \sum_{i=1}^M \frac{D_i^3(x, n) w_i(n)}{2} \right). \quad (15)$$

We can analytically solve for the initial value of  $\lambda$  from (15). We insert the two solutions of (15) into (12) and pick the  $\mathbf{w}(n+1)$  vector closest to the hyperplane in (13). This is determined by checking the error  $e(x, n)$ . We experimentally observed that this estimate provides convergence in forest-fire application. To determine a more accurate value of the Lagrange multiplier  $\lambda$ , we developed a heuristic search method based on the estimate

$\hat{\lambda}$ . If  $e(x, n) < 0$ , we choose  $\lambda_{\min} = \hat{\lambda} - 2|\hat{\lambda}|$ ,  $\lambda_{\max} = \hat{\lambda}$ , and if  $e(x, n) > 0$ , we choose  $\lambda_{\min} = \hat{\lambda}$ ,  $\lambda_{\max} = \hat{\lambda} + 2|\hat{\lambda}|$  as the upper and lower bounds of the search window. We only look at  $R$  values uniformly distributed between these limits to find the best  $\hat{\lambda}$  that produces the lowest error. In our wildfire detection application, we use  $R = 4$  as the length of the search window. We could have used a fourth-order Taylor series approximation in (14) and still obtained an analytical solution. After fourth-order approximations, a solution has to be numerically found. There are very efficient polynomial root-finding algorithms in the literature.

The pseudocode for the e-projection-based adaptive decision-fusion-based algorithm is given in Algorithm 2, which explains projection onto one hyperplane. In the algorithm,  $\lambda_{\min}$  and  $\lambda_{\max}$  are determined from the Taylor series approximation, as described earlier. The temporary variables  $\mathbf{v}$  and  $\mathbf{w}_T$  are used to find the  $\lambda$  value that produces the lowest error. A different  $\lambda$  value is determined for each sample at each time step. Obviously, a new value of  $\lambda$  has to be computed whenever a new observation  $x$  arrives.

---

#### Algorithm 2 The pseudocode for the EADF algorithm

---

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

$w_i(0) = 1/M$ , Initialization

**end for**

For each sample at time step  $n$ .

**for**  $\lambda = \lambda_{\min}$  to  $\lambda_{\max}$  **do**

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

$v_i(n) = w_i(n)$

$v_i(n) \leftarrow v_i(n) e^{\lambda D_i(x, n)}$

**end for**

**if**  $\|y(x, n) - \sum_i v_i(n) D_i(x, n)\|_2 < \|y(x, n) - \sum_i w_i(n) D_i(x, n)\|_2$  **then**

$\mathbf{w}_T(n) \leftarrow \mathbf{v}(n)$

**end if**

**end for**

$\mathbf{w}(n) \leftarrow \mathbf{w}_T(n)$

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

$w_i(n) \leftarrow w_i(n) / \sum_j w_j(n)$

**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**

---

Instead of the Shannon entropy  $x \log x$ , it is possible to use the regular entropy function  $\log x$  as the cost functional [34]. In this case

$$g(\mathbf{w}) = - \sum_i \log(w_i(n)) \quad (16)$$

which is convex for  $w_i(n) > 0$ . The e-projection onto the hyperplane  $H(x, n)$  can be obtained as follows:

$$w_i(n+1) = \frac{w_i(n)}{1 + \lambda w_i(n) D_i(x, n)}, \quad i = 1, 2, \dots, M \quad (17)$$

where the update parameter  $\lambda$  can again be obtained by inserting (17) into the hyperplane constraint in (13).

Penalizing the  $w_i(n) = 0$  case with an infinite cost may not be suitable for online adaptive fusion problems. However, the cost function

$$g(\mathbf{w}) = - \sum_i \log(w_i(n) + 1) \quad (18)$$

is always positive, convex, and differentiable for  $w_i(n) \geq 0$

In this case, weight update equation becomes

$$w_i(n+1) = \frac{w_i(n) - \lambda(w_i(n) + 1) D_i(x, n)}{1 + \lambda(w_i(n) + 1) D_i(x, n)}, \quad i = 1, 2, \dots, M \quad (19)$$

where the update parameter  $\lambda$  should be determined by substituting (19) into (13). Finding the exact value of  $\lambda$  when (13) is only a 4-D hyperplane using numerical methods is not difficult. In the forest-fire detection problem, we have only five subalgorithms. However, when the number of subalgorithms is high, new numerical methods should be determined for cost functions in (16) and (18).

For the wildfire detection problem, it is desirable that each subalgorithm should contribute to the compound algorithm because they characterize a feature of wildfire smoke. Therefore, weights of algorithms should be between 0 and 1. We want to penalize extreme weight values 0 and 1 more compared with the values in between. The entropy functional achieves this. On the other hand, the commonly used Euclidean norm penalizes high-weight values more compared with zero weight.

### C. Block Projection Method

Block projection-based methods are developed for inverse problems and active fusion methods [2], [19], [20], [30]. In this case, sets are assumed to arrive sequentially, and the  $q$  values of the most recently received observation sets are used to update the weights in the block projection approach. Adaptive projected subgradient method (APSM) works by taking a convex combination of the projections of the current weight vector onto those  $q$  sets. The weights calculated using this method are shown to converge to the intersection of hyperplanes [2], i.e., for each sample  $x$ , there exist  $\mathbf{w}^*$  such that

$$\mathbf{w}^* \in \bigcap_{n \geq n_0} H(x, n) \quad (20)$$

where  $n_0 \in \mathbb{N}$ .

The next values of weights  $\mathbf{w}(n+1)$  can be calculated from the  $q$  projections  $P_{H(x,j)}(\mathbf{w}(n))$  for  $j \in S_n = \{n-q+1, n-q+2, \dots, n\}$  using the APSM as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_n \left( \sum_{j \in S_n} \alpha_j(n) P_{H(x,j)}(\mathbf{w}(n)) - \mathbf{w}(n) \right) \quad (21)$$

where  $\alpha_j(n)$  is a weight used to control the contribution of the projection onto the  $j$ th hyperplane and  $\sum_{j \in S_n} \alpha_j(n) = 1$ ; any  $\mu_n$  can be chosen from  $(0, 2\mathcal{M}_n)$ , where

$$\mathcal{M}_n = \frac{\sum_{j \in S_n} \alpha_j(n) \|P_{H(x,j)}(\mathbf{w}(n)) - \mathbf{w}(n)\|_2}{\left\| \sum_{j \in S_n} \alpha_j(n) P_{H(x,j)}(\mathbf{w}(n)) - \mathbf{w}(n) \right\|_2}. \quad (22)$$

The weights of projections are usually chosen as  $\alpha_j(n) = 1/q$ , and  $\mu_n$  can be chosen as 1 since  $\mathcal{M}_n \geq 1$  is always true [2]. Both orthogonal and e-projections can be used as the projection operator  $P_{H(x,j)}$ . We experimentally observed the convergence of the entropic method. Proof of global convergence of the block e-projection method will be studied in the future.

## III. APPLICATION: COMPUTER-VISION-BASED WILDFIRE DETECTION

The EADF framework described in detail in the previous section with tracking capability is particularly useful when the online active learning problem is of a dynamic nature with drifting concepts [35]–[37]. In the video-based wildfire detection problem 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 (12) and (4) adjust the importance of individual subalgorithms by updating the weights according to the decisions of the oracle.

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

As an application of EADF, a computer-vision-based method for wildfire detection is presented in this paper. Security guards have to work 24 h in remote locations under difficult circumstances. They may simply 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 to reduce the average forest-fire detection time.



Fig. 2. Snapshot of typical wildfire smoke captured by a forest watch tower, which is 5 km away from the fire (rising smoke is marked with an arrow).

Cameras, once installed, operate at forest watch towers throughout the fire season for about six 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 feedback 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. The proposed active fusion algorithm can also be used in other supervised learning problems where classifiers combinations through feedback are required.

As described in the following section, the main wildfire detection algorithm is composed of five subalgorithms. 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 subalgorithms are linearly combined, and weights of subalgorithms are adaptively updated in our approach.

There are several approaches on automatic forest-fire detection in the literature. Some of the approaches are directed toward the detection of the flames using infrared and/or visible-range cameras, and some others aim at detecting the smoke due to wildfire [38]–[42]. There have been recent papers on sensor-based fire detection [43]–[45]. Infrared cameras and sensor-based systems have the ability to capture the rise in temperature; however, they are much more expensive compared with regular pan-tilt-zoom (PTZ) cameras. An intelligent space framework is described for indoor fire detection in [46]. However, in this paper, an outdoor (forest) wildfire detection method is proposed.

It is almost impossible to view flames of a wildfire from a camera mounted on a forest watch tower unless the fire is very near to the tower. However, smoke rising up in the forest due to a fire is usually visible from long distances. A snapshot of typical wildfire smoke captured by a lookout-tower camera from a distance of 5 km is shown in Fig. 2.

Guillemant and Vicente [42] based their method on the observation that the movements of various patterns, such as 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 the shape and smoothness of these histograms. It is possible to include most of the currently available methods as subalgorithms in the proposed framework and combine their decisions using the proposed EADF method.

Smoke at far distances ( $>100$  m to the camera) exhibits different spatio-temporal characteristics than nearby smoke and fire [47]–[49]. This demands specific methods explicitly developed for smoke detection at far distances rather than using nearby smoke detection methods described in [50]. The proposed approach is in accordance with the ‘weak’ artificial intelligence (AI) framework [51] 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 [52], [53].

#### IV. BUILDING BLOCKS OF A WILDFIRE DETECTION ALGORITHM

A 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 subalgorithms: 1) slow moving object detection in video; 2) smoke-colored region detection; 3) wavelet-transform-based region smoothness detection; 4) shadow detection and elimination; and 5) 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 subalgorithms are selected 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 Section II.

Decision functions  $D_i, i = 1, \dots, M$  of subalgorithms do not produce binary values 1 (correct) or  $-1$  (false), but they 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 forest fire in the viewing range of the camera. Output values of decision functions express the confidence level of each subalgorithm. The higher the value, the more confident the algorithm.

The first four subalgorithms are described in detail in [54], which is available online at the EURASIP webpage. We recently added the fifth subalgorithm to our system. It is briefly reviewed below.

##### A. Covariance-Matrix-Based Region Classification

The fifth subalgorithm deals with the classification of the smoke-colored moving regions. We first obtain a mask from the intersection of the first two subalgorithms and use the obtained smoke-colored moving regions as the input to the fifth algorithm. The regions are passed as bounding boxes of the connected regions of the mask. A region covariance matrix [55]



consisting of discriminative features is calculated for each region. For each pixel in the region, a 9-D feature vector  $z_k$  is calculated as follows:

$$z_k = \begin{bmatrix} x_1 & x_2 & Y(x_1, x_2) & U(x_1, x_2) & V(x_1, x_2) & \left| \frac{dY(x_1, x_2)}{dx_1} \right| & \left| \frac{dY(x_1, x_2)}{dx_2} \right| & \left| \frac{d^2Y(x_1, x_2)}{dx_1^2} \right| & \left| \frac{d^2Y(x_1, x_2)}{dx_2^2} \right| \end{bmatrix}^T \quad (23)$$

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

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

$$z_k = [z_k(i)]^T \quad (24)$$

where,  $z_k(i)$  is the  $i$ th entry of the feature vector. This feature vector is used to calculate the 9 by 9 covariance matrix of the regions using the fast covariance matrix computation formula [56], i.e.,

$$C_R = [c_R(i, j)] = \left( \frac{1}{n-1} \left[ \sum_{k=1}^n z_k(i)z_k(j) - Z_{kk} \right] \right) \quad (25)$$

where

$$Z_{kk} = \frac{1}{n} \sum_{k=1}^n z_k(i) \sum_{k=1}^n z_k(j)$$

where  $n$  is the total number of pixels in the region and  $c_R(i, j)$  is the  $(i, j)$ th 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 three 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 the feature vector  $f_R$  with  $9 \times 10/2 - 3 = 42$  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  $z_k$ .

An SVM with RBF kernel is trained with the region covariance feature vectors of smoke regions in the training database. We used 18 680 images used to train the SVM. The number of positive images that have actual smoke is 7011, and the rest are negative images that do not have smoke. Sample positive and negative images are shown in Fig. 3. The confusion matrix for the training set is given in Table I. The success rate is 99.3% for the positive images and 97.2% for the negative images.

The LIBSVM [57] software library is used to obtain the posterior class probabilities  $p_R = Pr(\text{label} = 1|f_R)$ , where  $\text{label} = 1$  corresponds to a smoke region. In this software

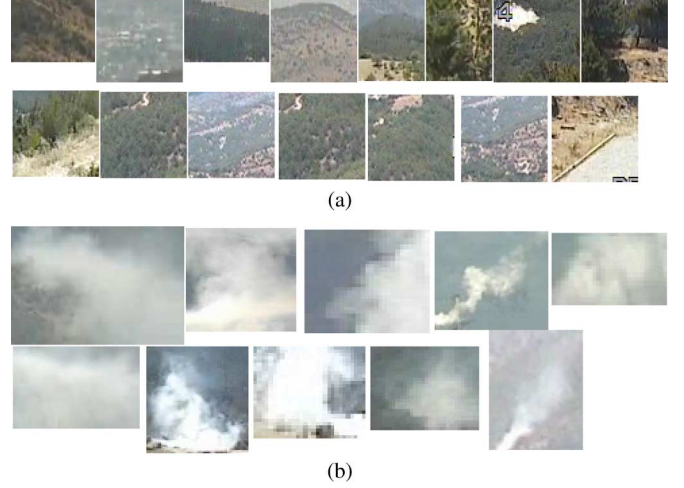


Fig. 3. Positive and negative images from the training set. (a) Negative training images. (b) Positive training images.

TABLE I  
CONFUSION MATRIX OF THE TRAINING SET

		Predicted Labels	
		Not Smoke	Smoke
Actual Labels	Not Smoke	11342/(97.2)%	327/(3.8%)
	Smoke	49/(0.7%)	6962/(99.3%)

library, posterior class probabilities are estimated by approximating the posteriors with a sigmoid function, as in [58]. If the posterior probability is larger than 0.5, the label is 1, and the region contains smoke according to the covariance descriptor. The decision function for this subalgorithm is defined as follows:

$$D_5(x, n) = 2p_R - 1 \quad (26)$$

where  $0 < p_R < 1$  is the estimated posterior probability that the region contains smoke. In [55], a distance measure based on eigenvalues is used to compare covariance matrices, but we found that individual covariance values also provide satisfactory results in this problem.

As pointed out earlier, the decision results of five subalgorithms  $D_1$ ,  $D_2$ ,  $D_3$ ,  $D_4$ , 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. 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. Initially, equal weights are assigned to each subalgorithm (see Fig. 4). There may be large variations between forestal areas, and substantial temporal changes may occur within the same forestal region. As a result, the weights of the individual subalgorithms will evolve in a dynamic manner over time.

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

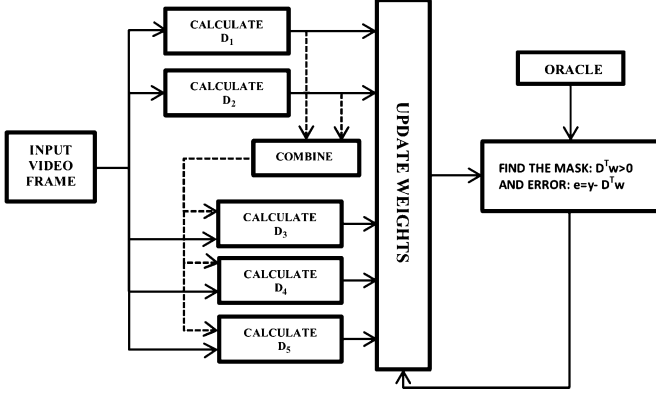


Fig. 4. Flowchart of the weight update algorithm for one image frame.

adjusting the weights for each preset once and then use 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; 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.

## V. EXPERIMENTAL RESULTS

### A. Experiments on Wildfire Detection

The proposed wildfire detection scheme with entropy-functional-based active learning method is implemented on a PC with an Intel Core Duo CPU 2.6-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 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 during the trials. A photograph from this test is presented in Fig. 5. The system also detected another forest fire in Cyprus in 2010. The software is currently being used in more than 60 forest watch towers in Turkey, Greece, and Cyprus.

The proposed EADF strategy is compared with the ULP scheme proposed in [59]. The ULP adaptive filtering method is modified to the wildfire detection problem in an online learning framework. In the ULP scheme, decisions of individual algorithms are linearly combined, similar to (1) as follows:

$$\hat{y}_u(x, n) = \sum_i v_i(n) D_i(x, n) \quad (27)$$

where the weights  $v_i(n)$  are updated according to the ULP algorithm, which assumes that the data (or decision values  $D_i(x, n)$ , in our case) are governed by some unknown probabilistic model  $P$  [59]. The objective of a universal predictor is to minimize the



Fig. 5. Photograph 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 rectangles.

expected cumulative loss. An explicit description of the weights  $v_i(n)$  of the ULP algorithm is given as follows:

$$v_i(n+1) = \frac{\exp\left(-\frac{1}{2c}\ell(y(x, n), D_i(x, n))\right)}{\sum_j \exp\left(-\frac{1}{2c}\ell(y(x, n), D_j(x, n))\right)} \quad (28)$$

where  $c$  is a normalization constant and the loss function for the  $i$ th decision function is

$$\ell(y(x, n), D_i(x, n)) = [y(x, n) - D_i(x, n)]^2. \quad (29)$$

The constant  $c$  is taken as 4, as indicated in [59]. The universal-predictor-based algorithm is summarized in Algorithm 3.

---

#### Algorithm 3 The pseudocode for the universal predictor

---

Universal Predictor( $x, n$ )

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

$$\ell(y(x, n), D_i(x, n)) = [y(x, n) - D_i(x, n)]^2$$

$$v_i(n+1) = \frac{\exp(-(1/2c)\ell(y(x, n), D_i(x, n)))}{\sum_j \exp(-(1/2c)\ell(y(x, n), D_j(x, n)))}$$

**end for**

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

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

    return 1

**else**

    return -1

**end if**

---

In the experiments, we compared eight different algorithms named FIXED, ULP, NLMS, NLMS-B, EADF, EADF-B, LOGX, and LOG(X + 1). NLMS-B and EADF-B are block projection versions of NLMS and EADF methods with block



TABLE II  
EIGHT DIFFERENT ALGORITHMS ARE COMPARED IN TERMS OF TRUE DETECTION RATES IN VIDEO CLIPS THAT CONTAIN WILDFIRE SMOKE

Video	Frames	True Detection Rates							
		FIXED	ULP	NLMS	NLMS-B	EADF	EADF-B	LOGX	LOG(X+1)
V1	768	87.63%	87.63%	87.63%	87.63%	87.63%	87.63%	87.89%	87.63%
V2	300	89.67%	<b>89.67%</b>	83.00%	89.66%	81.33%	86.00%	84.67%	89.66%
V3	550	<b>70.36%</b>	<b>70.36%</b>	68.18%	68.18%	67.09%	68.18%	67.09%	68.00%
V4	1000	<b>94.90%</b>	<b>94.90%</b>	90.80%	94.10%	90.50%	92.40%	93.30%	93.70%
V5	1000	<b>96.30%</b>	95.50%	91.10%	92.90%	91.90%	92.70%	92.40%	93.40%
V6	439	<b>80.87%</b>	<b>80.87%</b>	80.41%	80.41%	80.41%	80.41%	80.41%	80.41%
V7	770	85.71%	85.71%	85.71%	85.71%	85.84%	85.71%	85.71%	<b>85.97%</b>
V8	1060	98.68%	<b>99.15%</b>	98.86%	98.68%	98.77%	98.67%	98.96%	98.77%
V9	410	<b>80.24%</b>	<b>80.24%</b>	80.00%	80.00%	80.00%	80.00%	80.00%	80.00%
V10	1000	82.30%	82.30%	79.30%	82.40%	89.50%	90.70%	<b>91.10%</b>	81.30%
Average	-	<b>86.67%</b>	86.63%	84.50%	85.97%	85.30%	86.24%	86.15%	85.88

TABLE III  
EIGHT DIFFERENT ALGORITHMS ARE COMPARED IN TERMS OF FALSE-NEGATIVE (MISS) DETECTION RATES IN VIDEO CLIPS THAT CONTAIN WILDFIRE SMOKE

Video	Frames	Miss Detection Rates							
		FIXED	ULP	NLMS	NLMS-B	EADF	EADF-B	LOGX	LOG(X+1)
V1	768	12.37%	12.37%	12.37%	12.37%	12.37%	12.37%	12.11%	12.37%
V2	300	10.33%	<b>10.33%</b>	17.00%	10.34%	18.67%	14.00%	15.33%	10.34%
V3	550	<b>29.64%</b>	<b>29.64%</b>	31.82%	31.81%	32.91%	31.82%	32.91%	32.00%
V4	1000	<b>5.10%</b>	<b>5.10%</b>	9.20%	5.90%	9.50%	7.60%	6.70%	6.30%
V5	1000	<b>3.70%</b>	4.50%	8.90%	7.10%	8.10%	7.30%	7.60%	6.60%
V6	439	<b>9.13%</b>	<b>9.13%</b>	9.59%	9.59%	9.59%	9.59%	9.59%	9.59%
V7	770	4.29%	4.29%	4.29%	4.29%	4.16%	4.29%	4.29%	<b>4.03%</b>
V8	1060	1.32%	<b>0.85%</b>	1.14%	1.32%	1.23%	1.33%	1.04%	1.23%
V9	410	<b>19.76%</b>	<b>19.76%</b>	20.00%	20.00%	20.00%	20.00%	20.00%	20.00%
V10	1000	17.70%	17.70%	20.70%	17.60%	20.50%	19.30%	<b>8.90%</b>	18.70%
Average	-	<b>13.33%</b>	13.37%	15.50%	14.03%	14.70%	13.76%	13.85%	14.12

size  $q = 5$ . LOGX and LOG( $X + 1$ ) represent the algorithms that use  $-\log x$  and  $-\log(x + 1)$  as the distance functions. FIXED represents the unadaptive method that uses fixed weights, and ULP is the ULP-based approach. In Tables II IIIIV, and V, forest surveillance recordings containing actual forest fires and test fires, as well as video sequences where no fires are used.

In Table II, ten video sequences that contain wildfire smoke are tested in terms of true detection rates, which are defined as the number of correctly classified frames containing smoke divided by the total number of frames that contain smoke. V2, V4, V5, and V10 contain actual forest fires recorded by the cameras at forest watch towers, and the others contain artificial test fires. FIXED and ULP methods usually have higher detection rates, but there is not a significant difference from the adaptive methods. Our aim is to decrease false alarms without reducing the detection rates too much. Table IV is generated from the first alarm frames and times of the algorithms. The times are

comparable to each other, and all algorithms produced alarms in less than 13 s. Photographs from the test results in Table II are given in Fig. 6. For the wildfire detection problem, another important comparison criterion is false-negative (miss) detection rate, which is defined as the number of incorrectly classified frames containing smoke divided by the total number of frames that contain smoke. In Table III, the video sequences that contain wildfire smoke are tested in terms of false-negative (miss) detection rates.

A set of video clips containing clouds, moving cloud shadows, fog, and other moving regions that usually cause false alarms is used to generate Table V. The algorithms are compared in terms of false-alarm rates, which is defined as the number of misclassified frames that do not contain smoke divided by the total number of frames that do not contain smoke. Except for one video sequence, the EADF method produces the lowest false-alarm rate in the data set. The algorithms that use adaptive fusion strategy significantly reduce the false-alarm

TABLE IV  
EIGHT DIFFERENT ALGORITHMS ARE COMPARED IN TERMS OF FIRST ALARM FRAMES AND TIMES IN VIDEO CLIPS THAT CONTAIN WILDFIRE SMOKE

Video	First Alarm Frame / Time (secs.)							
	FIXED	ULP	NLMS	NLMS-B	EADF	EADF-B	LOGX	LOG(X+1)
V1	64/12.80	64/12.80	64/12.80	64/12.80	64/12.80	64/12.80	64/12.80	64/12.80
V2	42/8.40	42/8.40	67/13.40	42/8.40	68/13.60	53/10.60	58/11.60	42/8.40
V3	26/5.20	26/5.20	37/7.40	37/7.40	44/8.80	37/7.40	43/8.60	38/7.60
V4	25/5.00	25/5.00	58/11.60	25/5.00	59/11.80	33/6.60	25/5.00	43/8.60
V5	32/6.40	35/7.00	53/10.60	35/7.00	54/10.80	35/7.00	35/7.00	36/7.20
V6	21/4.20	21/4.20	21/4.20	21/4.20	21/4.20	21/4.20	21/4.20	21/4.20
V7	47/1.88	47/1.88	47/1.88	47/1.88	47/1.88	47/1.88	47/1.88	47/1.88
V8	12/1.33	12/1.33	12/1.33	12/1.33	12/1.33	12/1.33	12/1.33	12/1.33
V9	67/2.68	67/2.68	67/2.68	67/2.68	67/2.68	67/2.68	67/2.68	67/2.68
V10	33/6.60	33/6.60	50/10.00	33/6.60	51/10.20	33/6.60	33/6.60	44/8.80
Average	36.90/5.45	37.20/5.51	47.60/7.59	38.30/5.73	48.70/7.81	40.20/6.11	40.50/6.17	41.40/6.35

TABLE V  
EIGHT DIFFERENT ALGORITHMS ARE COMPARED IN TERMS OF FALSE-ALARM RATES IN VIDEO CLIPS THAT DO NOT HAVE WILDFIRE SMOKE

Video	Frames	False Alarm Rates							
		FIXED	ULP	NLMS	NLMS-B	EADF	EADF-B	LOGX	LOG(X+1)
V11	6300	0.03%	0.03%	0.03%	0.03%	<b>0.02%</b>	0.03%	0.03%	0.03%
V12	3370	7.00%	2.97%	1.01%	1.96%	<b>0.92%</b>	1.01%	1.66%	0.89%
V13	7500	3.13%	3.12%	2.77%	2.77%	2.77%	2.77%	<b>2.24%</b>	2.77%
V14	6294	17.25%	9.64%	2.27%	2.67%	<b>2.18%</b>	2.40%	3.23%	4.89%
V15	6100	4.33%	4.21%	2.72%	2.75%	1.80%	2.75%	<b>1.23%</b>	2.97%
V16	433	11.32%	11.32%	0.00%	0.00%	<b>0.00%</b>	0.00%	0.00%	0.00%
V17	7500	0.99%	0.00%	0.00%	0.00%	<b>0.00%</b>	0.00%	0.00%	0.00%
Average	-	6.29%	4.47%	1.26%	1.46%	<b>1.10%</b>	1.28%	1.20%	1.65%

rate of the system compared with the nonadaptive methods by integrating the feedback from the guard (oracle) into the decision mechanism within the active learning framework. One interesting result is that EADF-B and NLMS-B, which are the versions that use the block projection method developed for the case of infinite number of convex sets, usually produced more false alarms than the methods that do not use block projections.

In Fig. 7, typical false alarms issued to videos by an untrained algorithm with decision weights equal to 1/5 are shown.

In Fig. 8, the squared pixels errors of NLMS- and EADF-based schemes are compared for the video clip V12. The average pixel error for a video sequence  $v$  is calculated as follows:

$$\bar{E}(v) = \frac{1}{F_I} \sum_{n=1}^{F_I} \left( \frac{e_n}{N_I} \right) \quad (30)$$

where  $N_I$  is the total number of pixels in the image frame,  $F_I$  is the number of frames in the video sequence, and  $e_n$  is the sum of the squared errors for each classified pixel in image frame  $n$ . The figure shows the average errors for the frames between 500 and 900 of V12. At around the frames 510 and 800, the camera moves to a new position, and weights are reset to their initial values. The EADF algorithm achieves convergence faster than

the NLMS algorithm. The tracking performance of the EADF algorithm, which is better than the NLMS-based algorithm, can be observed after the frame number 600; at which point, some of the subalgorithms issue false alarms.

In Fig. 9, the weights of two different pixels from V12 are displayed for 140 frames. For the first pixel,  $D_1(x, n)$ ,  $D_3(x, n)$ , and  $D_4(x, n)$  get closer to 1 after the 60th frame; therefore, their weights are reduced. For the second pixel,  $D_2(x, n)$  issues false alarms after the fourth frame;  $D_2(x, n)$  and  $D_4(x, n)$  issue false alarms after the 60th frame.

#### B. Experiments on a UCI Data Set

The proposed method is also tested with a data set from the UCI machine learning repository to evaluate the performance of the algorithm in combining different classifiers. In the wildfire detection case, the image data arrive sequentially, and the decision weights are updated in real time. On the other hand, the UCI data sets are fixed. Therefore, the data set is divided into two parts: training and testing.

During the training phase, weights of different classifiers are determined using the EADF update method. In the testing phase, the fixed weights obtained from the training phase are used to



Fig. 6. Photographs from the test videos in Table II. The first two and the last two images are from the same video sequences.

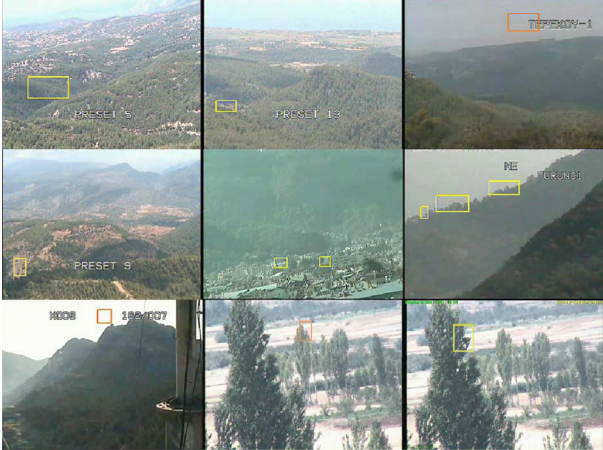


Fig. 7. False alarms issued to videos in Table V. The first two and the last two images are from the same video sequences. Cloud shadows, clouds, fog, moving tree leaves, and sunlight reflecting from buildings cause false alarms in an untrained algorithm with decision weights equal to 1/5.

combine the classifier decisions, which process the data in a sequential manner because both the NLMS and the EADF frameworks assume that the new data arrive in a sequential manner.

The test is performed on the ionosphere data from the UCI machine learning repository that consists of radar measurements to detect the existence of free electrons that form a structure in the atmosphere. The electrons that show some kind of structure in the ionosphere return “Good” responses; the others return “Bad” responses. There are 351 samples with 34-element feature vectors that are obtained by passing the radar signals through an autocorrelation function. In [60], the first 200 samples are used as the training set to classify the remaining 151 test samples. They obtained 90.7% accuracy with a linear perceptron, 92% accuracy with a nonlinear perceptron, and 96% accuracy with a back propagation neural network.

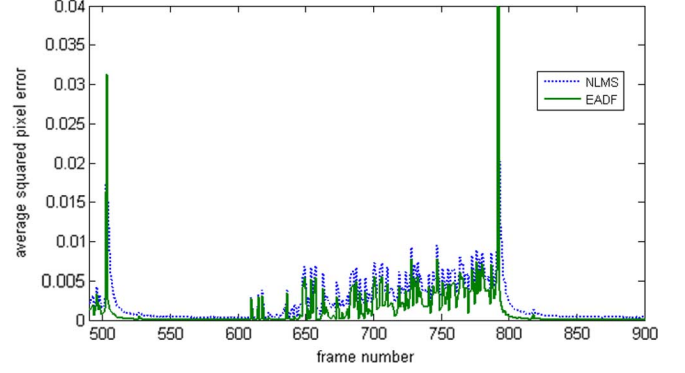


Fig. 8. Average squared pixel errors for the NLMS-based and the EADF-based algorithms for the video sequence V12.

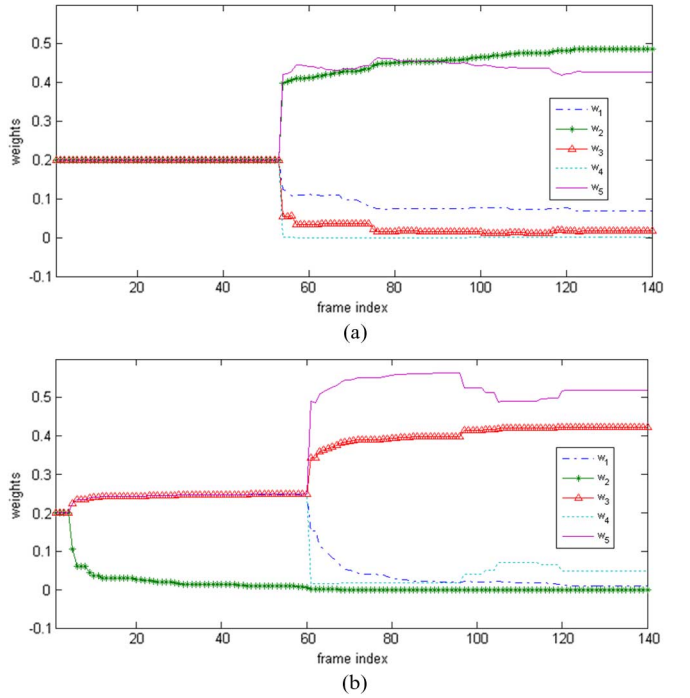


Fig. 9. Adaptation of weights in a video that do not contain smoke. (a) Adaptation of weights for a pixel at  $x = (55, 86)$  in V12. (b) Adaptation of weights for a pixel at  $x = (56, 85)$  in V12.

For this test, SVM, k-NN, and normalized cross-correlation classifiers are used. In addition, in this classification, the decision functions of these classifiers produce binary values with 1 corresponding to “Good” classification and -1 corresponding to “Bad” classification rather than scaled posterior probabilities in the range  $[-1, 1]$ .

The accuracy of the subalgorithms and EADF is shown in Table VI. The success rates of the proposed EADF and NLMS methods are both 98.01% which is higher than all the subalgorithms. Both the e-projection-based and orthogonal-projection-based algorithms converge to a solution in the intersection of the convex sets. It turns out that they both converge to the same solution in this particular case. This is possible when the intersection set of convex sets is small. The proposed EADF method is developed for real-time application in which data arrive sequentially. This example is included to show that the

TABLE VI  
ACCURACY OF SUBALGORITHMS AND EADF ON THE IONOSPHERE DATA SET

Data	Success Rates (%)				
	SVM	k-nn (k=4)	NCC	NLMS	EADF
Train	100.0	91.50	100.0	100.0	100.0
Test	94.03	97.35	91.39	98.01	98.01

EADF scheme can also be used in other data sets. It may be possible to get better classification results with other classifiers in this fixed UCI data set.

## VI. CONCLUSION

An EADF is proposed 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 subalgorithms, each of which yields 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 nonorthogonal e-projections onto convex sets describing subalgorithms. This general framework is applied to a real computer vision problem of wildfire 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. It is also observed that error rate of the proposed method is the lowest in our data set, compared with the ULP and the NLMS-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 a nonorthogonal e-projection onto a hyperplane describing the decision of the oracle.

## REFERENCES

- [1] O. Günay, K. Tademir, B. U. Töreyn, and A. E. Çetin, "Video based wildfire detection at night," *Fire Safety J.*, vol. 44, no. 6, pp. 860–868, Aug. 2009.
- [2] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Jan. 2011.
- [3] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, Jan. 1994.
- [4] V. I. Gorodetskiy and S. V. Serebryakov, "Methods and algorithms of collective recognition," *Autom. Remote Control*, vol. 69, no. 11, pp. 1821–1851, Nov. 2008.
- [5] B. V. Dasarthy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," *Proc. IEEE*, vol. 67, no. 5, pp. 708–713, May 1979.
- [6] A. Kumar and D. Zhang, "Personal authentication using multiple palm-print representation," *Pattern Recognit.*, vol. 38, no. 10, pp. 1695–1704, Oct. 2005.
- [7] X. Tang and Z. Li, "Video based face recognition using multiple classifiers," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2004, pp. 345–349.
- [8] M. A. Garciaacute;a and D. Puig, "Supervised texture classification by integration of multiple texture methods and evaluation windows," *Image Vis. Comput.*, vol. 25, no. 7, pp. 1091–1106, Jul. 2007.
- [9] A. Frank and A. Asuncion, UCI Machine Learning Repository Univ. California, Irvine, School of Information and Computer Sciences, 2010 [Online]. Available: <http://archive.ics.uci.edu/ml>
- [10] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 3, pp. 418–435, May/Jun. 1992.
- [11] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: an experiment," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 2, pp. 146–156, Apr. 2002.
- [12] D. Parikh and R. Polikar, "An ensemble-based incremental learning approach to data fusion," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 437–450, Apr. 2007.
- [13] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.
- [14] L. G. Gubin, B. T. Polyak, and E. V. Raik, "The method of projections for finding the common point of convex sets," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 6, pp. 1–24, 1967.
- [15] D. C. Youla and H. Webb, "Image restoration by the method of convex projections, Part I-Theory," *IEEE Trans. Med. Imag.*, vol. MI-1, no. 2, pp. 81–94, Oct. 1982.
- [16] A. E. Çetin, "Reconstruction of signals from Fourier transform samples," *Signal Process.*, vol. 16, no. 2, pp. 129–148, Feb. 1989.
- [17] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, Jul. 2008.
- [18] U. Niesen, D. Shah, and G. Wornell, "Adaptive alternating minimization algorithms," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1423–1429, Mar. 2009.
- [19] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Funct. Anal. Optim.*, vol. 25, no. 7/8, pp. 593–617, 2005.
- [20] K. Slavakis, I. Yamada, and N. Ogura, "The adaptive projected subgradient method over the fixed point set of strongly attracting nonexpansive mappings," *Numer. Funct. Anal. Optim.*, vol. 27, no. 7/8, pp. 905–930, 2006.
- [21] A. E. Çetin and R. Ansari, "Signal recovery from wavelet transform maxima," *IEEE Trans. Signal Process.*, vol. 42, no. 1, pp. 194–196, Jan. 1994.
- [22] P. L. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, no. 2, pp. 182–208, Feb. 1993.
- [23] S. Theodoridis and M. Mavroforakis, "Reduced convex hulls: A geometric approach to support vector machines," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 119–122, May 2007.
- [24] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. New York: Academic, 2006.
- [25] G. Baraniuk, "Compressed sensing [Lecture Notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
- [26] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [27] J.-F. Cai, S. Osher, and Z. Shen, Fast linearized Bregman iteration for compressed sensing UCLA, Los Angeles, CA, UCLA CAM Rep. (08-37), 2008.
- [28] J.-F. Cai, S. Osher, and Z. Shen, "Linearized Bregman iterations for compressed sensing," *Math. Comput.*, vol. 78, no. 267, pp. 1515–1536, Sep. 2009.
- [29] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, 1967.
- [30] G. T. Herman, "Image reconstruction from projections," *Real-Time Imag.*, vol. 1, no. 1, pp. 3–18, Apr. 1995.
- [31] Y. Censor and A. Lent, "An iterative row-action method for interval convex programming," *J. Optim. Theory Appl.*, vol. 34, no. 3, pp. 321–353, Jul. 1981.
- [32] H. J. Trussell and M. R. Civanlar, "The landweber iteration and projection onto convex set," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-33, no. 6, pp. 1632–1634, Dec. 1985.
- [33] M. I. Sezan and H. Stark, "Image restoration by the method of convex projections: Part 2—Applications and numerical results," *IEEE Trans. Med. Imag.*, vol. 1, no. 2, pp. 95–101, Oct. 1982.
- [34] Y. Censor and A. Lent, "Optimization of 'log x' entropy over linear equality constraints," *SIAM J. Control Optim.*, vol. 25, no. 4, pp. 921–933, Jul. 1987.
- [35] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data," *Mach. Learn.*, vol. 1, no. 3, pp. 317–354, Sep. 1986.

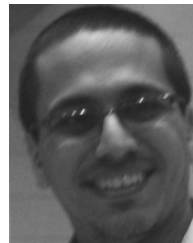
- [36] M. Karnick, M. Ahiskali, M. D. Muhlbaier, and R. Polikar, "Learning concept drift in nonstationary environments using an ensemble of classifiers based approach," in *Proc. IEEE IJCNN*, 2008, pp. 3455–3462.
- [37] K. Nishida, S. Shimada, S. Ishikawa, and K. Yamauchi, "Detecting sudden concept drift with knowledge of human behavior," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2008, pp. 3261–3267.
- [38] J. R. Martinez-de Dios, B. C. Arrue, A. Ollero, L. Merino, and F. Gómez-Rodríguez, "Computer vision techniques for forest fire perception," *Image Vis. Comput.*, vol. 26, no. 4, pp. 550–562, 2008.
- [39] J. Li, Q. Qi, X. Zou, H. Peng, L. Jiang, and Y. Liang, "Technique for automatic forest fire surveillance using visible light image," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2005, vol. 5, pp. 3135–3138.
- [40] I. Bosch, S. Gomez, L. Vergara, and J. Moragues, "Infrared image processing and its application to forest fire surveillance," in *Proc. IEEE Conf. AVSS*, 2007, pp. 283–288.
- [41] T. Celik, H. Ozkaramanli, and H. Demirel, "Fire and smoke detection without sensors: Image processing based approach," in *Proc. EUSIPCO*, 2007, pp. 1794–1798.
- [42] 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.*, vol. 40, no. 4, pp. 554–563, Apr. 2001.
- [43] M. Hefeeda and M. Bagheri, "Forest fire modeling and early detection using wireless sensor networks," in *Proc. IEEE Int. Conf. MASS*, 2007, pp. 1–6.
- [44] Y. G. Sahin, "Animals as mobile biological sensors for forest fire detection," *Sensors*, vol. 7, no. 12, pp. 3084–3099, Dec. 2007.
- [45] S. Chen, H. Bao, X. Zeng, and Y. Yang, "A fire detecting method based on multi-sensor data fusion," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2003, vol. 4, pp. 3775–3780.
- [46] P. Podrzaj and H. Hashimoto, "Intelligent space as a fire detection system," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2006, pp. 2240–2244.
- [47] B. U. Töreyn, Y. Dedeolu, and A. E. Çetin, "Flame detection in video using hidden Markov models," in *Proc. ICIP*, 2005, pp. II-1230–II-1233.
- [48] Y. Dedeolu, B. U. Töreyn, U. Güdükbay, and A. E. Çetin, "Real-time fire and flame detection in video," in *Proc. ICASSP*, 2005, pp. 669–672.
- [49] B. U. Töreyn, Y. Dedeolu, U. Güdükbay, and A. E. Çetin, "Computer vision based system for real-time fire and flame detection," *Pattern Recognit. Lett.*, vol. 27, pp. 49–58, 2006.
- [50] B. U. Töreyn, Y. Dedeolu, and A. E. Çetin, "Wavelet based real-time smoke detection in video," in *Proc. EUSIPCO*, 2005, pp. 2–5.
- [51] T. Pavlidis, *Computers vs Humans 2002* [Online]. Available: <http://www.theopavlidis.com/comphumans/comphuman.htm>
- [52] H. L. Dreyfus, *What Computers Can't Do*. Cambridge, MA: MIT Press, 1972.
- [53] H. L. Dreyfus, *What Computers Still Can't Do*. Cambridge, MA: MIT Press, 1992.
- [54] B. U. Töreyn, "Fire detection algorithms using multimodal signal and image analysis" Ph.D. dissertation, Bilkent Univ., Ankara, Turkey, 2009.
- [55] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. ECCV*, 2006, pp. 589–600.
- [56] F. Porikli and O. Tuzel, "Fast construction of covariance matrices for arbitrary size image windows," in *Proc. ICIP*, 2006, pp. 1581–1584.
- [57] C.-C. Chang and C.-J. Lin, LIBSVM: A Library for Support Vector Machines 2001 [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [58] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 1999, pp. 61–74.
- [59] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2685–2699, Oct. 1999.

- [60] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," in *Proc. Johns Hopkins APL Tech. Dig.*, 1989, pp. 262–266.



**Osman Gunay** received the B.Sc. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey. He is currently working toward the Ph.D. degree with the Department of Electrical and Electronics Engineering, Bilkent University.

His research interests include computer vision, video segmentation, and dynamic texture recognition.



University, Ankara.

**Behcet Ugur Toreyin** received the Ph.D. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, and the B.S. degree in electrical and electronics engineering from the Middle East Technical University, Ankara.

In 2009 and 2011, he was a Postdoctoral Research Associate with the Robotic Sensor Networks Laboratory, University of Minnesota, Minneapolis, and with the Wireless Research Laboratory, Texas A&M University at Qatar, Doha, Qatar, respectively. He is currently an Assistant Professor with Cankaya



His research interests include adaptive wavelet transformation and its applications in image processing.

**Kivanç Kose** is currently working toward the Ph.D. degree with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey.

During his M.Sc. period, he studied the compression of the 3-D mesh models under the supervision of Professor Enis Cetin. He implemented a new orthographic projection method for 3-D modeling. Moreover, he implemented a new adaptive wavelet transformation called connectivity-guided adaptive wavelet transformation for this projected 2-D model.



**A. Enis Cetin** (F'10) received the Ph.D. degree from the University of Pennsylvania, Philadelphia, in 1987.

From 1987 to 1989, he was an Assistant Professor of electrical engineering with the University of Toronto, Toronto, ON, Canada. Since 1989, he has been with Bilkent University, Ankara, Turkey. His research interests include signal and image processing, human-computer interaction using vision and speech, and audio-visual multimedia databases.

Dr. Çetin was an Associate Editor of the IEEE TRANSACTION ON IMAGE PROCESSING, between 1999 to 2003. he is currently a member of the Editorial Boards of *Journals Signal Processing* and *Journal of Advances in Signal Processing* and *Journal of Machine Vision and Applications*.