Soft SVM and Its Application in Video-Object Extraction

Yi Liu and Yuan F. Zheng, *Fellow, IEEE*

***Abstract*—As a requisite of content-based multimedia technologies, video-object (VO) extraction is a very important yet challenging task. In recent years, classification-based approaches have been proposed to handle VO extraction as a classification problem, for which some promising results have been reported using adaptive neural networks and support vector machines (SVMs). We observe that some training samples in video sequences exhibit partial or ambiguous class memberships, which does not comply with standard membership setups. This problem is addressed in the context of SVM in this paper. By reformulating SVM for the noncrisp classification scenario, we propose a machine which is capable of dealing with binary (or hard) as well as real-valued (or soft) class memberships. The new machine, which is named Soft SVM, is integrated into a VO extraction method, and its effectiveness is demonstrated by the experimental results.**

***Index Terms*—Fuzzy support vector machine (SVM), Soft SVM (S_SVM), video-object (VO) extraction.**

I. Introduction

VIDEO-OBJECT (VO) extraction, which refers to the process of segmenting and tracking semantic entities with pixel-wise accuracy [1], is an important yet challenging task for content-based video processing. For this purpose, a great number of approaches have been proposed [1]–[11], and satisfactory results for extracting VOs of homogeneous motion characteristics have been obtained. However, being robust in complicated scenarios, such as VOs moving with abrupt motions or occlusions, still remains a challenge. In recent years, a new group of approaches has emerged to treat VO extraction directly as a classification problem [12]–[14]. In these classification-based approaches, each VO is considered as a class, and VO extraction is realized by classifying every pixel to one of the available classes. By doing so, temporal association of objects between frames is automatically maintained through correct classification. As a result, it is more robust when VOs are of complicated motion characteristics. Moreover, by using the state-of-the-art classification tools, such as adaptive neural networks [12], AdaBoost [13], SVM [14], and ψ-learning [14], high classification accuracy can be achieved which directly leads to better performance of VO extraction.

Manuscript received January 17, 2006; revised October 25, 2006. Part of this work was presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia, PA, March 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Elias S. Manolakos.

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: liuyi@ece.osu.edu; zheng@ece.osu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.894403

The classification-based approaches consist of two phases: the training phase and the tracking phase. In the training phase, a training set is constructed, which can be accomplished either automatically [12] or manually [14]. Either way, one may find it difficult to attach the class labels to the pixels around the object boundary which usually manifest a gradual rather than a clean-cut transition from the object to the background. Doulamis *et al.* [12] consider these boundary-adjacent pixels to be in a region of uncertainty and exclude them from the training set so as not to confuse the training of the classifier. Unfortunately, the boundary-adjacent pixels play a critical role in defining accurate boundaries of VOs. Without the latter, the performance of any content-based video-processing algorithms relying on VO extraction will be fundamentally affected. We argue that the boundary-adjacent pixels carry useful information regarding the transition from one class to the other which should be included in the training process to achieve an accurate boundary. To do so, we relax the constraint of the binary membership of either -1 or 1 to real-valued between -1 and 1, which, named soft membership, allows each sample to belong to different classes (i.e., object or background) by different degrees.

As a matter of fact, partial or ambiguous membership is a common phenomena for a large number of applications, such as climatic prediction [15], soil classification [16], remote sensing [17], and ecological modeling [18], where soft classification can capture the nature of the data better than hard classification. Unfortunately, many powerful classifiers lack this ability because they are formulated in the context of crisp classification where each sample falls into either one class or the other. SVM is one of them. To address this problem, Fuzzy SVM (FSVM) has been developed [19], which associates each training sample with a fuzzy membership $s_i > 0$ and employs s_i to weigh the corresponding penalty term in the objective function. FSVM extends the horizon of SVM, but the information embedded in the membership is missing when the corresponding sample is correctly classified because the penalty term is nonzero only when misclassification occurs.[1]

In this paper, we propose Soft SVM (S_SVM) which takes into account the real-valued memberships regardless of whether the samples are classified correctly or not. When the samples are classified to the wrong class, the errors are penalized in such a way that the more certain the class labels are, the heavier the penalty will be. If the samples are correctly classified, they are still allowed to influence the boundary by pulling it close or

[1]Here, the term "misclassification" refers to the error vectors, which include the samples that are wrongly classified and those that are correctly classified but lie inside the margin strip.

pushing it away depending on the relative magnitude of their memberships. In either case, the samples can make a different contribution to the learning of the decision boundary.

The fundamental feature of S_SVM is to treat samples discriminatingly by changing both the weight of the penalty term and the constraints in the objective function. Similar formulations motivated from different applications have been presented in [23] and [24]. The work of Pérez-Cruz [23] is focused on the problem of channel equalization. It associates each training sample with a so-called margin and penalty factor to make SVM adaptive to nonstationary environments. However, how to choose the two factors in practice is not discussed in detail in [23]. Chapelle's work [24], on the other hand, is motivated by a new induction principle which links the loss function to the density estimation. The author suggests including the quality of the estimation, measured by γ_i, in the formulation of SVM where a large γ_i means that the density of the corresponding point is poorly estimated or the "location is uncertain." Chapelle's formulation imposes light penalties on the uncertain points, which agrees with S_SVM, but pushes the hyperplane away from them, which is the opposite of S_SVM.

This paper is organized as follows. First, the training mechanism of S_SVM is formulated in Section II. Section III presents a short review of VO extraction, and then introduces a classification-based approach in which S_SVM is employed as the classifier. Experimental results are presented in Section IV which is followed by conclusions in Section V.

II. Soft SVM

A. Soft Memberships

Consider N training samples $\{x_1, y_1\}, \ldots, \{x_N, y_N\}$, where $x_i \in R^k$ is the input vector and y_i is the corresponding class label. For SVM $y_i \in \{-1, 1\}$ while for S_SVM $y_i \in [-1\ 1]$ is a real-valued variable called the soft membership.

In some applications, it is more typical and may be more convenient to employ the fuzzy membership

$$\left(x_1, m_1^+, m_1^-\right), \left(x_2, m_2^+, m_2^-\right), \ldots, \left(x_N, m_N^+, m_N^-\right) \quad (1)$$

where two quantities m_i^+ and m_i^- satisfying

$$0 \le m_i^+, m_i^- \le 1 \text{ and } m_i^+ + m_i^- = 1 \quad (2)$$

describe the partial memberships of class 1 and -1, respectively. It is easy to see that the quantity $(m_i^+ - 0.5)$ plays a similar role in the range $[-0.5\ 0.5]$ as y_i in the range $[-1\ 1]$. Based on this observation, a linear one to one mapping between the fuzzy and soft membership can be established as

$$y_i = 2\left(m_i^+ - 0.5\right) = 2m_i^+ - 1 = m_i^+ - m_i^- \quad (3)$$

and, therefore, they can be interchangeably used.

y_i can be considered as a slider moving between -1 and 1. The more it slides toward 1 (-1), the more degrees x_i exhibit to be a member of class 1 (-1), or the more certain we are about the fact that x_i belongs to class 1 (-1). When the membership stays in the middle ($y_i = 0$ or $m_i^+ = m_i^- = 0.5$), we have absolutely no idea which class the sample x_i comes from.

In order to fit into the notations of SVM, we further decompose y_i into two parameters as $y_i = \tilde{y}_i \lambda_i$, where $\tilde{y}_i = \text{sign}(y_i)$ is the binary class label as in SVM and $\lambda_i = |y_i|$, named certainty measure, contains the additional membership information. Now, the training set can be rewritten as $(x_1, \tilde{y}_1, \lambda_1), \ldots, (x_N, \tilde{y}_N, \lambda_N)$ with $\tilde{y}_i \in \{-1, 1\}$ and $\lambda_i \in [0\ 1]$. The above decomposition of y_i only leads to a minor variation of the original optimization process of SVM which will be shown.

B. Formulation of S_SVM

In this subsection, we derive the detailed formulation of S_SVM. We start with the simple case of linearly separable sets, and then generalize it to the linearly nonseparable case which is more general in reality. To facilitate the discussion, the formulation of S_SVM will be presented side by side with that of SVM. The review of SVM is brief and we refer interested readers to [20]–[22] for details.

1) Linearly Separable Case: When the training samples are linearly separable, SVM yields the optimal hyperplane that separates two classes without training error and maximizes the minimum distance from the samples to the hyperplane. Mathematically, the optimal hyperplane, described by the equation $w^T x + b = 0$ with $w \in R^k$ and $b \in R^1$, is obtained by solving the following optimization problem:

$$\begin{aligned} \text{minimize}: \quad & L(w) = \frac{1}{2}\|w\|^2 \\ \text{subject to}: \quad & \tilde{y}_i(w^T x_i + b) \ge 1. \end{aligned} \quad (4)$$

In S_SVM, we take into account the strength of the membership, or the certainty measures. Let us consider the simplest case where the training set consists of only two samples. Under the assumption that the uncertainties of the class labels are caused by the similarity of the data near the separating boundary, the optimal hyperplane is expected to move toward the point with smaller λ to reflect the unbalanced soft memberships rather than stay in the middle of the two samples as yielded by SVM. To do so, we relax the constraints in (4) as $\tilde{y}_i(w^T x_i + b) \ge \lambda_i$ and, as a result, the objective function of S_SVM becomes

$$\begin{aligned} &\text{minimize}: L(w) = \frac{1}{2}\|w\|^2 \\ &\text{subject to}: \quad \tilde{y}_i(w^T x_i + b) \ge \lambda_i. \end{aligned} \quad (5)$$

To illustrate how this modification is going to affect the position of the optimal hyperplane, we provide an example with only two samples in Fig. 1. Evidentally, the orientation of the hyperplane remains the same as that of SVM, which keeps the same maximal margin between the two classes. In the meantime, the hyperplane moves closer to the less certain sample which has smaller λ. Moreover, the analytical solution reveals that the hyperplane is shifted to the exact location where its distances to the support vectors (the two samples themselves in this case) are proportional to the certainty measures λ_i [Fig. 1(c)].

2) Linearly Nonseparable Case: For linearly nonseparable cases where zero training error is not attainable, SVM is generalized by introducing the non-negative slack variables ξ_i to

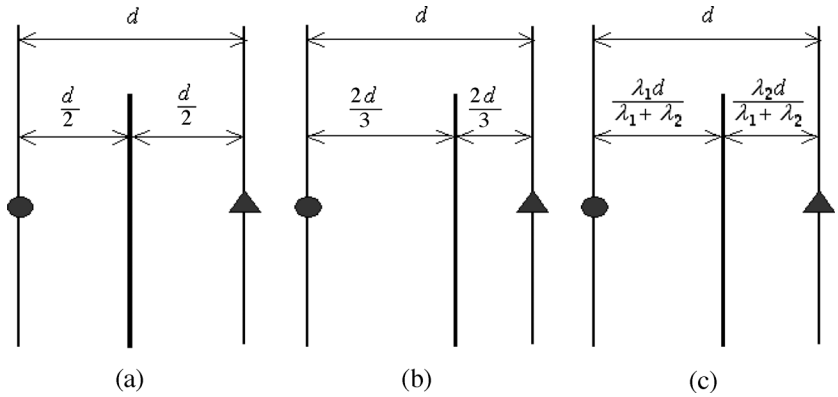


Fig. 1. Illustration of how SVM and S_SVM handle two samples with different λ's. Suppose we are given two training samples x_1 (the solid circle) and x_2 (the triangle). The class labels are $\tilde{y}_1 = 1$ and $\tilde{y}_2 = -1$ with $\lambda_1 = 1$ and $\lambda_2 = 0.5$, respectively. (a) SVM, which only considers the binary memberships, yields the hyperplane right in the middle between x_1 and x_2. (b) S_SVM, on the other hand, shifts the hyperplane to the sample with smaller certainty λ. (c) For the two-sample scenario, the location of the hyperplane, in general, can be solved analytically and given as shown.

penalize the misclassification, from which the optimization problem becomes

$$\begin{aligned} \text{minimize}: \quad & L(w,\xi_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to}: \quad & \tilde{y}_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \end{aligned} \tag{6}$$

Here, C is the parameter balancing the importance between the maximization of the margin and the minimization of the training error.

In analogy to SVM, we also introduce the non-negative variables ξ_i, which satisfy $\tilde{y}_i(w^T x_i + b) \geq \lambda_i - \xi_i$, to penalize the objective function of S_SVM when misclassification occurs. However, as mentioned before, x_i belongs to class $\tilde{y}_i$ at different certainty levels measured by λ_i and we should worry more about the misclassification of the samples with higher λ_i. Thus, in S_SVM, the error term ξ_i is further modified to $\lambda_i\xi_i$ to differentiate the penalty imposed on the error, which yields the following formulation:

$$\begin{aligned} \text{minimize}: \quad & L(w,\xi_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\lambda_i\xi_i \\ \text{subject to}: \quad & \tilde{y}_i(w^T x_i + b) \geq \lambda_i - \xi_i. \end{aligned} \tag{7}$$

As one can see from (6) and (7), all of the training samples are treated equally when $\lambda_i = 1$ and S_SVM is reduced to SVM. Another extreme is $\lambda_i = 0$. When that occurs, the penalty term $\lambda_i\xi_i$ becomes zero regardless of the result of the classification. As a result, the sample x_i is equivalently disregarded and would have no contribution to the learning of the decision function, which makes perfect sense since the total uncertainty ($\lambda_i = 0$) makes x_i no different than any other uncollected samples at all.

Similar to SVM, the optimization problem of S_SVM can be transformed into the dual problem

$$\begin{aligned} \text{maximize}: \quad & \sum_{i=1}^{N}\lambda_i\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j x_i^T x_j \\ \text{subject to}: \quad & \sum_{i=1}^{N}\tilde{y}_i\alpha_i = 0, \qquad 0 \leq \alpha_i \leq \lambda_i C. \end{aligned} \tag{8}$$

The optimal $\bar{w}$ is a linear combination of x_i as $\bar{w} = \sum_{i=1}^{N}\bar{\alpha}_i\tilde{y}_i x_i$, where $\bar{\alpha}_i$ denotes the optimal point of (8). As for the optimal b, it can be determined from the following Kuhn–Tucker conditions:

$$\bar{\alpha}_i\left(\tilde{y}_i(\bar{w}^T x_i + \bar{b}) - \lambda_i + \bar{\xi}_i\right) = 0, \text{ and } (\lambda_i C - \bar{\alpha}_i)\bar{\xi}_i = 0. \tag{9}$$

From the derivation above, one can see that the dual problem of S_SVM is a quadratic programming problem similar to that of SVM. The computational load for training the new machine thus stays the same.

For the applications where linear S_SVM is not suitable, nonlinear S_SVM is suggested. Similar to nonlinear SVM, it maps the input vector x nonlinearly to a much higher dimensional space in which the optimal hyperplane is derived by applying the linear S_SVM described before.

C. Discussions About S_SVM

1) Generating Certainty Measures: One important issue associated with S_SVM is to assign the certainty measures appropriately to training samples which, in general, is application dependent. Here, we offer a choice from the probability point of view.

Considered as a random variable, the class label Y_i of the training sample x_i is either 1 or -1 with certain conditional probability $P(Y_i = 1|x_i)$ and $P(Y_i = -1|x_i)$. We propose to utilize the conditional probability to describe the partial memberships in (1), which defines

$$m_i^+ = P(Y_i = 1|x_i) \quad \text{and } m_i^- = P(Y_i = -1|x_i). \tag{10}$$

Using (3), we obtain the soft membership y_i as

$$y_i = m_i^+ - m_i^- = P(Y_i = 1|x_i) - P(Y_i = -1|x_i). \tag{11}$$

The maximal certainty $\lambda_i = |y_i| = 1$ is produced when $P(Y_i = 1|x_i) = 1$ or $P(Y_i = -1|x_i) = 1$. On the contrary, when $P(Y_i = 1|x_i) = P(Y_i = -1|x_i) = 0.5$, we have $\lambda_i = |y_i| = 0$, indicating the maximal uncertainty about the class label.

It should be mentioned here that the conditional probability may be obtained only for training samples based on the knowledge specified by the application and, therefore, are not gen-

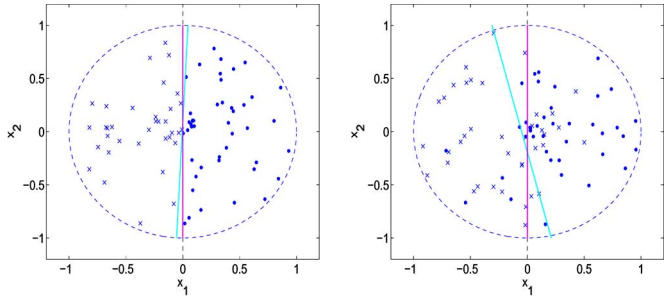


Fig. 2. Example of 2-D linear classification. 80 training samples are randomly generated from the uniform distribution over the unit disk $\{x_i = (x_{i1}, x_{i2}) | x_{i1}^2 + x_{i2}^2 \leq 1\}$ with the conditional probability $P(Y_i = 1|x_i) = (1 + x_{i1})/2$ and $P(Y_i = -1|x_i) = (1 - x_{i1})/2$. The samples in class 1 and -1 are plotted as dots and crosses, respectively. By using (11), the soft membership is obtained as $y_i = x_{i1}$. The hyperplanes yielded by S_SVM and SVM are represented by the magenta and cyan lines, respectively. Obviously, in both cases, S_SVM produces the perfect hyperplane $x_1 = 0$ while SVM has a much larger deviation. (a) Linearly separable case. (b) Linearly nonseparable case.

erally available for all x to be classified. One example will be shown later in Section III-C.

2) Invariance Analysis: The superior performance of S_SVM on synthetic data using the soft membership described above is shown in Fig. 2, but in reality, it is difficult to precisely know the absolute value of the memberships. The conditional probabilities in (11), for instance, are usually unknown and have to be estimated. Fortunately, S_SVM is invariant to the scale of λ_i, which makes it less sensitive to the imprecision of the certainty assignment.

Suppose each λ_i is scaled by a positive number M and let $\lambda_i' = M\lambda_i$. It can be shown that the pair $(\bar{w}', \bar{b}')$, which is the solution to the following optimization problem:

$$\begin{aligned} \text{minimize}: \quad & L(w', \xi_i) = \frac{1}{2}\|w'\|^2 + C\sum_{i=1}^{N} \lambda_i' \xi_i \\ \text{subject to}: \quad & \tilde{y}_i(w'^T x_i + b') \geq \lambda_i' - \xi_i \end{aligned} \tag{12}$$

satisfies $\bar{w}' = M\bar{w}$ and $\bar{b}' = M\bar{b}$ where $(\bar{w}, \bar{b})$ is the solution to (7). Since we have

$$\bar{w}'^T x + \bar{b}' = 0 \Leftrightarrow M\bar{w}^T x + M\bar{b} = 0 \Leftrightarrow \bar{w}^T x + \bar{b} = 0 \tag{13}$$

it is concluded that the two hyperplanes which are described by $\bar{w}^T x + \bar{b} = 0$ and $\bar{w}'^T x + \bar{b}' = 0$ represent the same decision boundary. This property of scale-invariant makes S_SVM more applicable for solving real-world problems since, in reality, the relative value of certainty measures can be obtained more easily than the absolute values.

3) Underlying Assumptions of S_SVM: As mentioned before, the nonbinary membership is not an unusual phenomena for real-world problems. The motivation of FSVM and S_SVM is the same: to treat different samples differently according to their different memberships. S_SVM is not always more applicable whenever we are given a set of training samples $(x_i, \tilde{y}_i, \lambda_i)$ with $\tilde{y}_i \in \{-1, 1\}$ and $0 \leq \lambda_i \leq 1$. Whether S_SVM should be employed depends on how the memberships are generated. One fundamental feature of S_SVM is that the information of λ_i is utilized in the way that the optimal hyperplane moves toward the samples with smaller λ_i. The underlying assumption for doing so is that the soft memberships result from the ambiguous nature of the class labeling caused by the overlapping of the data near the true boundary. Another good example is the application of face recognition. Given a number of pictures of two faces, one could have two different cases:

Case 1) the two faces look alike;
Case 2) some pictures were taken more recently and others are older.

Between the above two cases, case 1 is a better target application of S_SVM because of the ambiguity between the two faces. In case 2, ambiguity is not an issue, but the idea instead is that more recent pictures are more useful and, therefore, should be given more weight. In the latter case, λ_i can be used as the regularization parameters to weight the error terms differently and accordingly which leads to the following:

$$\begin{aligned} \text{minimize}: \quad & L(w, \xi_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \lambda_i \xi_i \\ \text{subject to}: \quad & \tilde{y}_i(w^T x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \end{aligned} \tag{14}$$

Equation (14) as a matter of fact is the FSVM as proposed in [19] where λ_i replaces the notation s_i. Unlike S_SVM, the solution to FSVM is not scale invariant. The scaling of λ_i by M is equivalent to substituting the regularization parameter C to MC, which changes the originally desired balance between the margin and the classification error, and potentially leads to a new hyperplane.

However, for applications where the labeling of the collected data is ambiguous because of the similarity feature they exhibit, we expect S_SVM to deliver better performance than FSVM, and such an application is presented in the following section to demonstrate the advantage of S_SVM.

III. VO Extraction: An Application of S_SVM

A. Related Work in VO Extraction

To facilitate the content-based video processing, MPEG-4 introduces the concept of video object (VO) and brings up the problem of VO extraction. Also referred to as the problem of object "segmentation and tracking" in many papers, VO extraction is faced with the following challenges.

1) VOs are defined as semantic entities and are therefore heterogenous in spatial features.
2) VOs cannot be represented only by the centroid because they are often nonrigid and deformable.
3) VO extraction needs to do more than localization. The pixel-wise resolution is required for segmenting or extracting objects from video frames.

The classic approaches for object tracking or object detection aim at localizing the centroid of objects and only provide bounding boxes as the results. For this reason, they cannot be applied directly to VO extraction and, hence, new techniques have been developed.

In the literature, the problem of VO extraction is approached either in an automatic or semiautomatic fashion. Segmentation

and tracking are the two major components of the automatic approaches, and according to the criterion of segmentation, they can be further categorized into two classes: spatial based and temporal based. The spatial-based method [2]–[4] partitions each frame into homogeneous regions with respect to color or intensities, and then every region is tracked using the motion information. Typical partitioning algorithms include morphological watershed [2], K-means clustering [3], region growing [4], and the recursive shortest spanning tree [5]. The temporal-based approach [6]–[8] on the other hand, utilizes the motion rather than spatial information to obtain the initial position of VOs. One popular scheme is to detect the changes between the frames to track the objects. Due to the image noise, the temporally segmented objects have an irregular boundary which needs to be further refined utilizing the spatial information of the image.

Automatic or unsupervised VO extraction is extremely difficult, and approaches aforementioned are hardly robust to any modality complicated videos. VO extraction can also be achieved in a semiautomatic fashion [1], [9]–[11], which is characterized as modeling and searching. VOs are initially extracted with the user's assistance and a model representing the object is created. A variety of models has been proposed including: active contour [10], 2-D mesh [25], [26], binary model [27], deformable templates [28], corners, and lines [29], etc. Then, the model is placed on the possible positions in subsequent frames and the object is located where the best match is found.

Accuracy and complexity are two critical issues for VO extraction, which have to be traded off in practice. The spatial-based approach can yield relatively accurate object boundary, but the computational load is quite high since the segmentation has to be performed on the whole image for every frame. In comparison, the temporal-based approaches often produce irregular object boundaries which need to be refined using the spatial information. As the boundary fine-tuning procedure involves only the segmented moving regions instead of the whole frame, higher efficiency can be achieved. For the modeling-and-searching type of methods, the key to efficiency is to effectively predict where the object might be to reduce the searching area. Mostly realized by motion estimation based on the assumption of smooth motion, the prediction is not reliable when occlusion or abrupt motion occurs.

Unlike the traditional approaches, the classification-based methods treat VO extraction explicitly as an object/background classification problem, and they have the potential to achieve both accuracy and low complexity. The methods are accurate because powerful classifiers can be trained for specific object/background separation. Low complexity, on the other hand, is achieved through evaluating the classification function at each pixel which involves only simple calculations (e.g., $w^T x + b$ for linear SVM). To further improve the efficiency, we suggested earlier that it is not necessary to classify pixel by pixel [14]. Instead, by designing a pyramid boundary refining schemes, the number of the classification operation is significantly reduced and the processing time to produce smooth boundary with pixel-wise accuracy reaches around 0.4 s per frame.

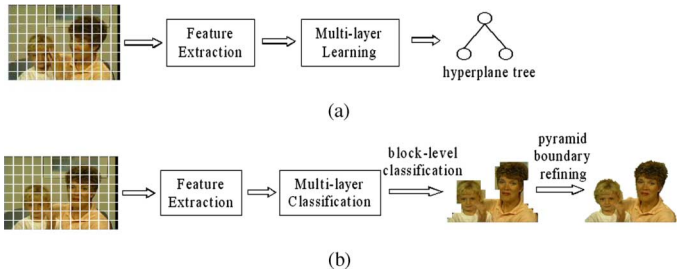


Fig. 3. Overview of the proposed approach. (a) Training phase. (b) Tracking phase.

For the remainder of this paper, we will integrate S_SVM into the classification-based VO extraction method described in [14] and present the experimental results which demonstrate the effectiveness of S_SVM.

B. Feature Extraction and Block Modeling for VO and Background Classification

The method proposed in [14] is a semiautomatic approach which needs the user to identify the VO of interest in the first frame. Then, the algorithm will locate and extract the object with pixel-wise accuracy from every subsequent frame automatically. An overview of the method is presented in Fig. 3 and described below.

The first step of the training phase concerns the construction of the training set (x_i, y_i). The first frame of the sequence is always chosen as the training frame, in which the VO of interest is defined and segmented from the background with the user's assistance. As a result, we know for every pixel in the first frame whether it belongs to the object. Instead of treating pixels individually, the approach suggests block modeling which represents pixels and their local neighbors collectively. More specifically, the first frame is decomposed into blocks, which are defined as object blocks $(y_i = 1)$ if the pixel at the center of the block belongs to the object or background blocks $(y_i = -1)$ otherwise. Discrete cosine transform (DCT) is then applied to each block and based on the DCT coefficients' local features $\vec{f}_{\text{local}}$ and neighboring features $\vec{f}_{\text{neighbor}}$ are constructed as follows:

$$\vec{f}_{\text{local}} = (f_0, f_1, f_2, f_3)^T = \begin{pmatrix} c(0,0) \\ \sqrt{\sum_{j=1}^{8} c(0,j)^2} \\ \sqrt{\sum_{i=1}^{8} c(i,0)^2} \\ \sqrt{\sum_{i=1}^{8}\sum_{j=1}^{N-1} c(i,j)^2} \end{pmatrix}. \quad (15)$$

Here, f_0 is the average intensity, and f_1 and f_2 represent the horizontal and vertical edges, respectively. All of the other high-frequency information is contained in the last component f_3.

The neighboring features $\vec{f}_{\text{neighbor}}$ are extracted from eight neighboring blocks which are adjacent to the block under the study in the vertical, horizontal, and diagonal directions as

B_1	B_2	B_3
B_8	B_0	B_4
B_7	B_6	B_5

Fig. 4. Eight-connected neighboring blocks of block B_0.

shown in Fig. 4. With $\mathrm{avg}(\mathrm{B}_i)$ denoted as the average intensity of block B_i, we compute the neighboring features as

$$\vec{f}_{\text{neighbor}} = \begin{pmatrix} \mathrm{avg}(\mathrm{B}_1 + \mathrm{B}_2 + \mathrm{B}_3) \\ \mathrm{avg}(\mathrm{B}_3 + \mathrm{B}_4 + \mathrm{B}_5) \\ \mathrm{avg}(\mathrm{B}_5 + \mathrm{B}_6 + \mathrm{B}_7) \\ \mathrm{avg}(\mathrm{B}_7 + \mathrm{B}_8 + \mathrm{B}_1) \end{pmatrix}. \tag{16}$$

The calculations given above only consider the gray-scale information. When the video sequence is chromatic, we compute (15) and (16) for each color component and then concatenate the vectors, respectively, to form the training vector x_i. Now with the training pairs (x_i, y_i) ready, a classifier, such as SVM, can be trained to separate the object from the background.

In the tracking phase, each subsequent frame is also divided into blocks, and for each block, the trained classifier is applied to determine the class label of the centering pixel as well as the block. Then, the tracking mask is formed by all of the identified object blocks, at which point the resolution of boundary of the object is as large as the size of the block. Finally, the pixel-wise accuracy is obtained by applying a pyramid boundary refining algorithm [14] which refines the object boundary in an efficient and scalable manner. The details of the latter algorithms are not important in presenting the current approach. The interested readers are referred to [14] for more information.

C. Apply S_SVM to VO Extraction

Recall that in the training phase, the training frame is divided into blocks and each of them is labeled depending on which class the centering pixels belong to. As mentioned before, we observe during the experiments that there are uncertainties or ambiguities about the class labels of the blocks that lay around the object boundary (i.e., they cannot be fully assigned to either one of the two classes). For this reason, we propose to employ S_SVM to train the classifier, which is a major component of the classification-based VO extraction approach.

The question immediately follows is how to generate the probabilities in (11) to obtain the real-valued membership $y_i \in [-1\ 1]$ for a given block B_i. An abundant volume of methods can be found in the literature for probability computation, and recently, an algorithm based on SVM has been proposed [30]. Here, for this specific application, we propose a very simple method as the following. First, we use the normalized number of object and background pixels contained in block B_i to define the conditional probability. More specifically, for the block size of $L \times M$, we have

$$\begin{cases} P(Y_i = 1 | X_i = \mathrm{B}_i) = \dfrac{\#\text{ of object pixels}}{L \times M} \\ P(Y_i = -1 | X_i = \mathrm{B}_i) = \dfrac{\#\text{ of background pixels}}{L \times M}. \end{cases} \tag{17}$$

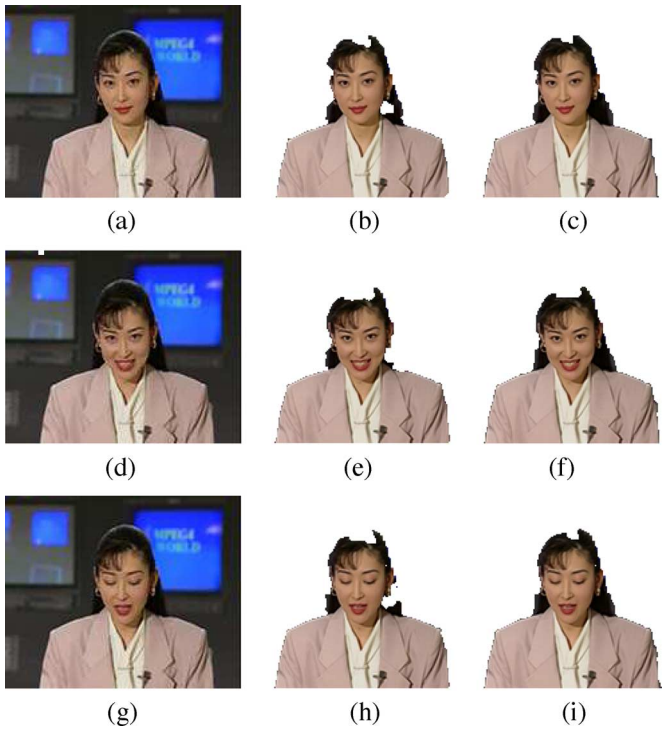


Fig. 5. Applying SVM and S_SVM to Akiyo on the first layer. (a) Frame 4. (b) SVM. (c) S_SVM. (d) Frame 255. (e) SVM. (f) S_SVM. (g) Frame 280. (h) SVM. (i) S_SVM.

Then, according to (11), the soft membership y_i is obtained as

$$y_i = \frac{\#\text{ of object pixels} - \#\text{ of background pixels}}{L \times M}. \tag{18}$$

Obviously, y_i is the normalized difference between the number of object and background pixels contained in the block. When the block locates completely inside (or outside) the object, we have $y_i = 1$ (or $y_i = -1$), showing no labeling ambiguity at all. When the blocks reside on the object boundary, y_i varies between -1 and 1. More specifically, if the block contains more object pixels than background pixels, y_i is positive and, therefore, the corresponding block is an object block. Similarly, $y_i \in (-1\ 0]$ if the background is dominant. The extreme case $y_i = 0$ occurs when the number of object and background pixels are equal, which indicates the maximal uncertainty about the class label.

IV. Experimental Results

Experiments are conducted to the standard MPEG-4 test video sequences including Akiyo, Mom and Daughter, and Sun Flower Garden, and the performance is compared among SVM, S_SVM, and FSVM.

The membership y_i of every training block is generated using (18). Then, the optimization problems in (6) and (7) are solved to produce SVM and S_SVM, respectively, where the variables $\tilde{y}_i$ and λ_i are obtained from y_i by $\tilde{y}_i = \mathrm{sign}(y_i)$ and $\lambda_i = |y_i|$ as stated in Section II-A. Another learning machine employed for performance comparison is FSVM. As shown in [19], the performance of FSVM is affected by the choice of memberships. In our experiment setup, we assign FSVM as the same memberships as in S_SVM, which yields the optimization of (14)

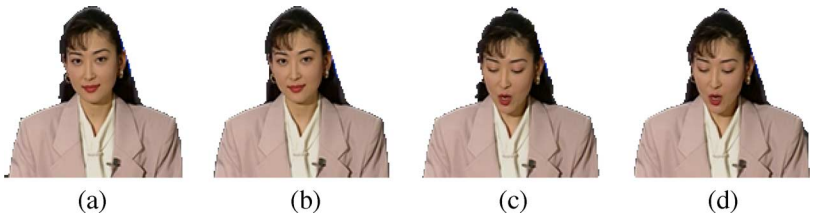


Fig. 6. Extracted VO of Akiyo after the second layer classification. (a) and (b) from frame 4 and (c) and (d) are from frame 126. (a) SVM. (b) S_SVM. (c) SVM. (d) S_SVM.

for the training of FSVM. By doing so, the membership issue is put aside and the fundamental difference between these learning machines becomes the key factor that affects their performance, which makes the comparison more focused and meaningful.

The performance of S_SVM is also affected by the parameter C, which is selected during the training process as follows. First, S_SVM is trained on the first frame of a sequence by setting $C = 2^{-2}$. Then, the trained S_SVM is applied to each subsequent frame of the sequence for classification, and the average classification error $r(C)$ over the whole sequence is calculated. This process is repeated for every C in $[2^{-2}, 2^{-1}, \ldots, 2^{12}]$, among which the one that yields the lowest $r(C)$ is considered as the optimal one of S_SVM for the particular sequence. At last, the performance of S_SVM that is trained with the optimal C is reported in the performance comparison. The procedure is the same to choose the best C for SVM and FSVM as well. When another video sequence is tested, the whole training process stated above will restart.

The first sequence to test is Akiyo. It is a typical head-and-shoulder type of sequence, and the simplest kernel, the linear kernel, is employed. Some original frames and the extracted VO are given in Fig. 5, which show that the objects extracted by S_SVM are more complete than that by SVM. The errors concentrate on the area of the lady's dark hair where the transition from the object to the dark background is blurred and, consequently, the blocks exhibit similarities. Evidentally, S_SVM is more accurate in classifying the "confusing" hair regions. To obtain a more complete object, we apply second-layer learning which intends to reclassify the object blocks that have been wrongly classified as background by the first classifier [14]. As one can see, the performance of SVM improves after the second-layer classification is applied as shown in Fig. 6, but again it is outperformed by S_SVM which yields a smoother and more accurate contour of the object.

Another test sequence is Mom and Daughter, in which mother and daughter are intended to be extracted together. In contrast to Akiyo, this sequence shows heterogenous spatial and temporal characteristics. The mother's head turns around slowly with her left hand even disappearing in the middle of the sequence while the daughter stays nearly still most of the time. We first apply the linear kernel to this sequence. However, a significant part of the object is missing, which shows that in this case, the linear kernel is too simple to model the nonlinear boundary between the object and background classes. Nonlinear kernel functions, such as polynomial kernels and radial basis function (RBF), are then tested. Through the experiments, RBF is found to be the one that yields the lowest classification errors and, hence, we

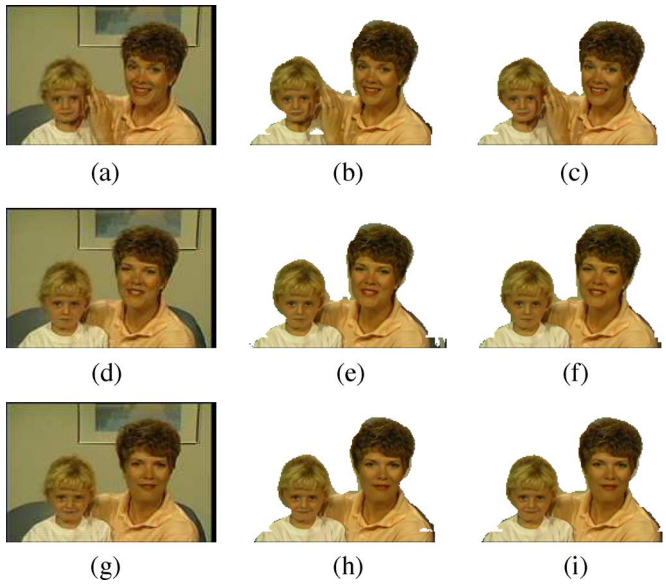


Fig. 7. Applying SVM and S_SVM to Mom and Daughter using the RBF kernel. (a) Frame 16. (b) SVM. (c) S_SVM. (d) Frame 60. (e) SVM. (f) S_SVM. (g) Frame 146. (h) SVM. (i) S_SVM.

choose RBF as the kernel function. The extracted objects by SVM and S_SVM are displayed in the second and third columns of Fig. 7, respectively. Obviously, the nonlinear S_SVM works well too.

Among the sequences tested in the experiments, Sun Flower Garden is the most challenging. Unlike the previous videoconference kind of sequences, it displays a natural scene that is rich in color and texture with a nonstationary camera. In addition, for a few frames, the houses, which are the selected VO, are only partially viewable. Once again, RBF is employed as the kernel function and S_SVM produces better results as one can see in Fig. 8 where S_SVM recognizes the houses that are cluttered by tree branches while SVM does not.

Some different results produced by S_SVM, SVM and FSVM are also displayed side by side in Fig. 9 which, together with the previous figures, shows the superiority of S_SVM over SVM and FSVM visually.

So far, the proposed method is evaluated on a subjective basis. In recent years, a number of measures have been proposed to objectively assess the performance of video segmentation and tracking [31], [32]. When the ground-truth segmentation maps are available, which is the case for the three sequences we are working with, the so-call relative quality evaluations are applied. As the name suggests, relative evaluations measure the

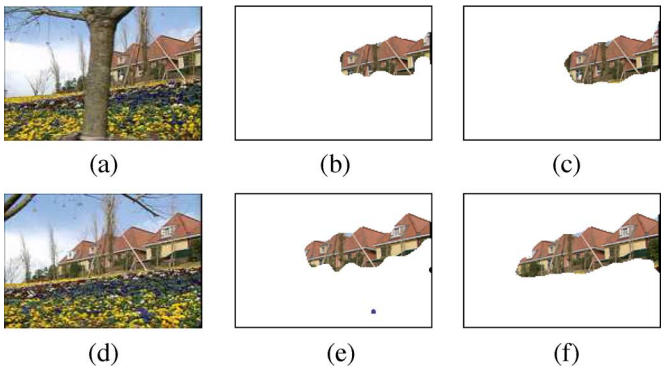


Fig. 8. Applying SVM and S_SVM to Sun Flower Garden. (a) Frame 12. (b) SVM. (c) S_SVM. (d) Frame 65. (e) SVM. (f) S_SVM.

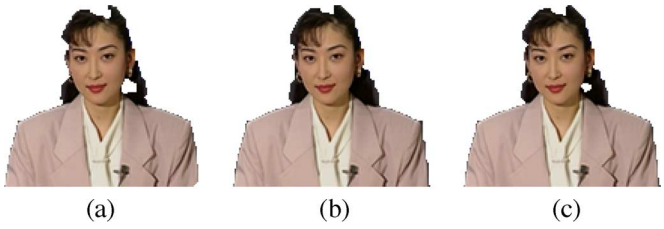


Fig. 9. Extracted VO of Akiyo by different machines on the first layer. (a) SVM. (b) S_SVM. (c) FSVM.

similarity between the reference segmentation and the estimate segmentation obtained by certain VO extraction methods. The typically employed metrics include shape fidelity [31], the similarity of spatial features, such as areas [33]; the consistence of the trajectories and velocities of the objects [34]; or a combination of both spatial and temporal distortions [35]. Here, we adopt the criterion proposed in [36] as

$$d\left(\mathrm{VO}_n^{\mathrm{est}}, \mathrm{VO}_n^{\mathrm{ref}}\right) = \frac{\sum_{(x,y)} \mathrm{VO}_n^{\mathrm{est}}(x,y) \bigoplus \mathrm{VO}_n^{\mathrm{ref}}(x,y)}{\sum_{(x,y)} \mathrm{VO}_n^{\mathrm{ref}}(x,y)} \tag{19}$$

where $\mathrm{VO}_n^{\mathrm{est}}$ and $\mathrm{VO}_n^{\mathrm{ref}}$ denote the estimated and the reference binary VO mask of frame n, and $\bigoplus$ is the binary XOR operation. This measure is chosen for the following three reasons. First of all, it is simple to compute. Second, it has been widely used in VO extraction [7], [12]. Third and most important, it basically measures the classification error of each frame which is the ultimate metric to assess classifiers, such as SVM and S_SVM. In Fig. 10, the errors yielded by SVM, S_SVM, and FSVM are plotted versus the number of frames as the dashed, solid, and dotted lines, respectively. Throughout the whole sequences, the solid line is below the other two lines showing that S_SVM achieves the highest classification accuracy and, consequently, yields the best extracted VO.

As one can see, the error curves in Fig. 10 have peaks and valleys along the sequence. We observe that when the content of the frames is similar to the training frame, the error is usually low. Take Akiyo as an example. The training frame, which is the first frame, is shown in Fig. 11(a). In the frames such as 22 or 216, the lady is talking while keeping her posture almost the same as in the first frame Fig. 11(b). As can be seen in Fig. 10(a), the errors are relatively low. Due to the similarity between the

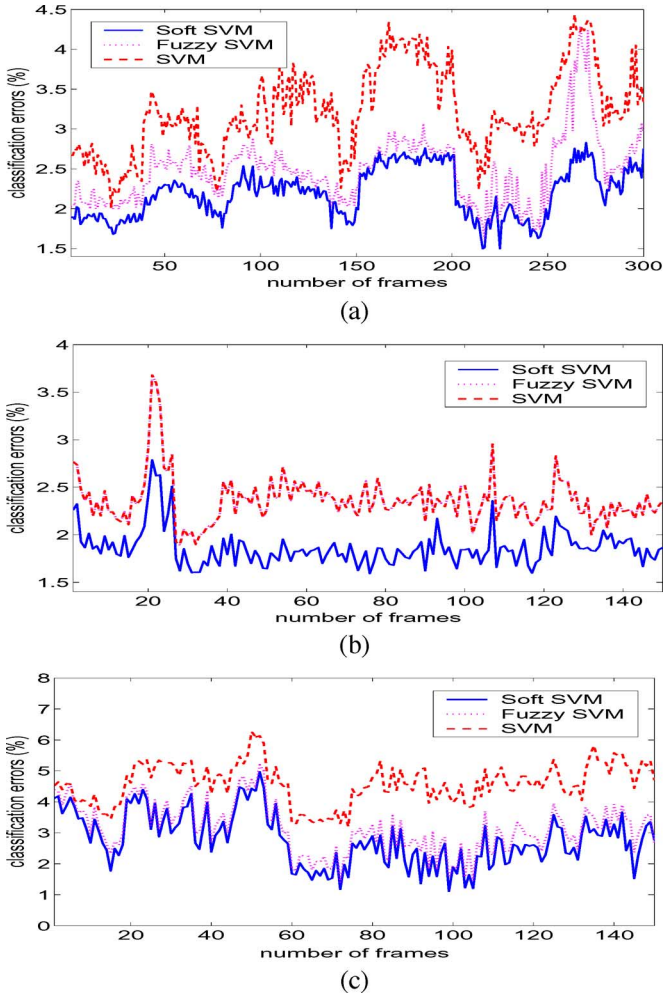


Fig. 10. Comparison of the classification accuracy among SVM, FSVM, and S_SVM. (a) Akiyo. (b) Mom and Daughter. (c) Sun Flower Garden.

Fig. 11. Sample frames of sequence Akiyo. (a) Frame 1. (b) Frame 22. (c) Frame 267.

lady's hair and the dark background around the head, the hair area is the most vulnerable part for misclassification. So when the anchor lady bows her head to a large degree in frames 265 $\sim$ 270 and more hair part is introduced, the classification error goes up. A typical frame is shown in Fig. 11(c). The performance of SVM, FSVM, and S_SVM all degrades in this case, but S_SVM still does the best.

The average errors of classification as well as the standard deviations for each sequence are shown in Table I. It should be pointed out that the absolute difference between the errors does not fully demonstrate how powerful S_SVM is considering the fact that SVM and FSVM have delivered very low classification errors already. To get an idea of how much S_SVM improves

TABLE I
AVERAGE CLASSIFICATION ERRORS PRODUCED BY SVM, S_SVM, AND FSVM, AND THE RELATIVE ERROR REDUCTION YIELDED BY S_SVM WITH RESPECT TO SVM (R_1) AND FSVM (R_2)

	Absolute Errors of Classification						Relative Error Reduction			
	Average			Std ($\times 10^{-3}$)			R_1		R_2	
	SVM	S_SVM	FSVM	SVM	S_SVM	FSVM	average	max	average	max
Akiyo	3.2%	2.2%	2.5%	5.4	3.1	4.2	31.3%	51.7%	11.0%	39.7%
Mom and Daughter	2.4%	1.8%	2.4%	2.6	2.4	2.6	20.9%	30.8%	20.9%	30.8%
Sun Flower Garden	4.6%	2.8%	3.1%	6.5	8.5	8.1	39.9%	73.6%	12.1%	43.3%

with respect to SVM and FSVM, we also list the relative difference in Table I, which is defined as

$$R_1 = \frac{E_{\text{SVM}} - E_{\text{S\_SVM}}}{E_{\text{SVM}}} \tag{20}$$

and

$$R_2 = \frac{E_{\text{FSVM}} - E_{\text{S\_SVM}}}{E_{\text{FSVM}}} \tag{21}$$

where $E_{(\cdot)}$ denotes the classification errors yielded by the corresponding machine. The relative improvement as shown is significant. For example, for the Akiyo sequence, S_SVM outperforms SVM and FSVM by 31.3% and 11.0% in average, and by 51.7% and 39.7% in maximum, respectively.

It should also be noted that for the Mom and Daughter sequence Fig. 10(b), the dashed line and the dotted line coincide because FSVM and SVM yield the same decision function. This is not surprising because by using the RBF kernel, zero training error is attainable, and all of the penalty terms ξ_i become zero which flattens the only effect of different λ_i in FSVM. Nevertheless, FSVM delivers the second best performance among the three machines, which supports our motivation that the fuzzy feature of data, if there is any, should be taken into consideration when the machine is trained.

S_SVM is also compared against SVM when trained with only samples whose certainty measures are higher than certain thresholds. Five thresholds are tested, and the average classification errors of both methods are reported in Table II. For Akiyo, S_SVM is marginally outperformed at certain thresholds, but for the other two sequences S_SVM is significantly better. This result shows that even when SVM is trained with the samples with high certainty measures, S_SVM still delivers better performance. Recall that when the certainty measures all become 1, S_SVM is identical to SVM. Therefore, as long as there is uncertainty, S_SVM is still a better choice.

We also compare S_SVM with classifiers outside of the family of SVM, namely K nearest neighbors (KNNs) and neural networks. We test different K from $K = 1$ to $K = 15$, and the lowest errors produced are reported in Table III. For the sequence of Akiyo, the KNN classifier with $K = 1$, produces less errors around the lady's hair part and, consequently, performs better than S_SVM. Sun Flower Garden is the sequence for which S_SVM performs significantly better. KNN struggles because it can only recognize the red brick roofs of the houses and incorrectly classifies most of the remaining part such as the yellow walls as background. The second classifier we test is neural networks. In [12], a feedforward network with one output layer and one hidden layer of 15 neurons is used for VO extraction. In our experiment setup, we adopt the same structure and the performance is reported in the fourth column of Table III. This network is outperformed by both S_SVM and KNN.

The major computational burden of the proposed method is the training of S_SVM. Once trained, the classification step of S_SVM involves only simple calculation which is cost efficient. In our approach, the training of S_SVM is performed only once for the first frame, and for every subsequent frame, it is the classification step that is applied. Therefore, although the method is dealing with a sequence of images, the computational burden is not high.

TABLE II
COMPARISON OF THE AVERAGE CLASSIFICATION ERRORS BETWEEN S_SVM AND SVM WITH THRESHOLDING

	SVM with Thresholding					S_SVM
	0.1	0.3	0.5	0.7	0.9	
Akiyo	2.4%	2.0%	1.8%	2.2%	2.9%	2.2%
Mom and Daughter	4.3%	3.4%	3.0%	3.7%	3.6%	1.8%
Sun Flower Garden	5.2%	5.3%	4.7%	4.4%	4.9%	2.8%

TABLE III
COMPARISON OF THE AVERAGE CLASSIFICATION ERRORS AMONG S_SVM, KNN, AND NEURAL NETWORKS

	S_SVM	KNN	Neural Networks
Akiyo	2.2%	1.7% ($K = 1$)	2.3%
Mom and Daughter	1.8%	2.7% ($K = 5$)	3.4%
Sun Flower Garden	2.8%	9.17% ($K = 2$)	11.2%

As pointed out in Section II, the computational complexity for training S_SVM is the same as SVM. In the tracking phase, the run time consumed by these two machines is almost identical as well.[2] For instance, for the aforementioned three sequences, the average run time per frame is 0.396, 0.388, and 0.439 s, respectively, when SVM is applied, and 0.401, 0.395, and 0.441 s when S_SVM is applied.

V. Conclusion

Despite its great success in a large number of applications, SVM is yet limited to crisp classification scenarios. In this paper, we present S_SVM, a reformulated version of SVM which allows training samples to belong to different classes by different degrees without increasing the computational cost. Classification-based VO extraction is presented as an application of the approach showing that S_SVM captures the fuzzy nature better between two classes than the traditional SVM. S_SVM also outperforms FSVM which uses fuzzy memberships to consider the uncertainty of the training samples.

As pointed out before, the assignment of the soft membership is application dependent and requires a deep understanding of the property of the data involved. In this work, a connection between the soft memberships and the conditional probability is made, which yields a simple yet effective membership function (18) for the particular application presented.

It is observed from the experimental results that even when the training of the classifier is only performed once, the tracking results for the whole sequence are of good quality. This is because there is no significant change of the video content in the presented sequences so that the information captured by the first frame is rich enough to generate a classifier that is robust for the rest of the sequence. Otherwise, retraining is necessary. To do so, a scene change module, such as the one proposed in [12], should be incorporated in the system to detect the change of the video content and to signal the necessity of the retraining when necessary.

How to extend the current approach from single VO to multiple VO extraction is a natural question. In this paper, single VO extraction is treated as a binary classification problem. Similarly, the task to extract N objects can be formulated as an $(N+1)$-category classification problem, that is, one class for the background and N classes for the VOs of interest. Most of the mechanisms, such as block representation and classification, are still applicable while an $(N+1)$-category rather than a binary classifier is needed. Unfortunately, SVM is a binary classifier, and so is the proposed S_SVM. Moreover, multiclass SVM is still an ongoing and immature topic in machine learning. For this reason, this paper only considers the single VO scenario. An attempt to tackle the case of multiple VO without using S_SVM is reported in another paper [37].

In this paper, discussions have been focused on the training stage of SVM. In the meantime, research work can also be found on "softening" the decision output of SVM at the classification stage [38], [39]. How to include both in the framework of SVM to render a more powerful classifier will be an interesting research topic.

[2]The frame size is 176×144 and the experiments are carried out on a Pentium IV 2.5-GHz PC.

References

[1] D. G. Prerz, C. Gu, and M. T. Sun, "Semantic video object extraction using four-band watershed and partition lattice operators," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 603–618, May 2001.

[2] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 539–546, Sep. 1998.

[3] I. Kompatsiaris and M. G. Strintzis, "Spatialtemporal segmentation and tracking of objects for visualization of videoconference image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1388–1403, Dec. 2000.

[4] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, Aug. 2001.

[5] E. Tuncel and L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 776–781, Aug. 2000.

[6] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving objects and background separation," *Signal Process.*, vol. 66, no. 2, pp. 219–232, 1998.

[7] C. Kim and J. N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 122–129, Feb. 2002.

[8] S. Y. Chien, S. Y. Ma, and L. G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 577–586, Jul. 2002.

[9] C. Gu and M. C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 572–584, Sep. 1998.

[10] S. Sun, D. R. Haynor, and Y. Kim, "Semiautomatic video object segmentation using Vsnakes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 75–82, Jan. 2003.

[11] C. He, J. Dong, Y. F. Zheng, and S. C. Ahalt, "Object tracking using the Gabor wavelet transform and the golden section algorithm," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 528–538, Dec. 2002.

[12] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, "An efficient fully unsupervised video object segmentation scheme using an adaptive neural-network classifier architecture," *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 616–630, May 2003.

[13] S. Avidan, "Ensemble tracking," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Jun. 2005, vol. 2, pp. 20–25.

[14] Y. Liu and Y. F. Zheng, "Video object segmentation and tracking using ψ-learning classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 885–899, Jul. 2005.

[15] A. B. McBratney and A. W. Moore, "Application of fuzzy sets to climatic classification," *Agricult. Forest Meteor.*, vol. 35, pp. 85–165, 1985.

[16] A. B. McBratney and J. J. de Gruijter, "A continuum approach to soil classification by modified fuzzy K-means with extragrades," *J. Soil Sci.*, vol. 43, pp. 79–159, 1992.

[17] A. D. Amo, J. Montero, A. Fernandez, M. Lopez, J. M. Tordesillas, and G. Biging, "Spectral fuzzy classification: An application," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 32, no. 1, pp. 42–48, Feb. 2002.

[18] A. Salski, O. Franzle, and P. Kandzia, "Fuzzy logic in ecological modeling," *Ecol. Modeling*, vol. 85, 1995.

[19] C. F. Lin and S. D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[20] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[21] C. Cortes and V. N. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[22] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[23] F. Pérez-Cruz and A. Artés-Rodríguez, "Adaptive SVC for nonlinear channel equalization," in *Proc. EUSIPCO*, 2002, vol. II, pp. 45–48.

[24] O. Chapelle, "Support vector machines: Induction Principles, adaptive tuninig and prior knowledge," Ph.D. dissertation, Laboratoire d'Informatique, Univ. Paris 6, Paris, France, 2002.

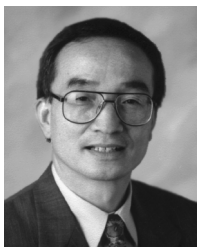
[25] Y. Altunbasak and A. M. Tekalp, "Occlusion-adaptive, content-based mesh design and forward tracking," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1270–1280, Sep. 1997.

[26] P. V. Beek, A. M. Tekalp, N. Zhuang, I. Celasun, and M. Xia, "Hierarchical 2D mesh representation, tracking and compression for object-based video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 353–369, Mar. 1999.

[27] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 525–538, Sep. 1998.
[28] Y. Zhong, A. K. Jain, and M. P. Dubuisson-Jolly, "Object tracking using deformable templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 544–549, May 2000.
[29] H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation," *Image Vis. Comput.*, vol. 13, pp. 695–703, Nov. 1995.
[30] K. Goh, E. Chang, and B. Li, "Using one-class and two-class SVMs for multiclass image annotation," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 10, pp. 1333–1346, Oct. 2005.
[31] P. L. Correia and F. Pereira, "Objective evaluation of video segmentation quality," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 186–200, Feb. 2003.
[32] C. Erdem, B. Sankur, and A. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Trans. Image Process.*, vol. 13, no. 7, pp. 937–951, Jul. 2004.
[33] M. Levine and A. Nazif, "Dynamic measurement of computer generated image segmentations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7, no. 2, pp. 155–164, Mar. 1985.
[34] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," presented at the 2nd IEEE Workshop Performance Evaluation of Tracking and Surveillance, 2001.
[35] C. E. Erdem and B. Sankur, "Performance evaluation metrics for object-based video segmentation," in *Proc. 10th Eur. Signal Processing Conf.*, Sep. 2000, vol. 2, pp. 917–920.
[36] *Refined Procedure for Objective Evaluation of Video Generation Algorithms*, ISO/IEC JTC1/SC29/WG11 M3448, Mar. 1998, M. Wollborn and R. Mech.
[37] Y. Liu and Y. F. Zheng, "Multiple video object extraction using multicategory ψ-learning," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, May 2006, vol. 5, pp. 909–912.
[38] T. Inoue and S. Abe, "Fuzzy support vector machines for pattern classification," in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2001, pp. 1449–1454.
[39] J. Platt, "Probabilistic outputs for SVMs and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 2000, pp. 61–74.

Yi Liu received the Ph.D degree in electrical and computer engineering from The Ohio State University, Columbus, in 2006 and the B.S. and M.S. degrees in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 1997 and 2000, respectively.

Her research interests include machine learning, pattern recognition, and their applications in the area of image/video processing.



Yuan F. Zheng (F'97) received the B.S. degree from Tsinghua University, Beijing, China, in 1970 and the M.S. and Ph.D. degrees in electrical engineering from The Ohio State University (OSU), Columbus, in 1980 and 1984, respectively.

From 1984 to 1989, he was with the Department of Electrical and Computer Engineering at Clemson University, Clemson, SC. Currently, he is Professor and was the Chairman of the Department of Electrical and Computer Engineering from 1993 to 2004 at The OSU, where he has been since 1989. From 2004 to 2005, he spent a sabbatical year at the Shanghai Jiao Tong University, Shanghai, China, where he continues to be involved as Dean of the School of Electronic, Information and Electrical Engineering for part-time administrative and research activities. His research interests include image and video processing for compression, object classification, object tracking, and robotics for which his current activities are in robotics and automation for high-throughput applications in biology studies. His research has been supported by the National Science Foundation, Air Force Research Laboratory, Office of Naval Research, Department of Energy, DAGSI, and ITEC-Ohio. He was and is on the Editorial Board of five international journals.

Dr. Zheng received the Presidential Young Investigator Award from Ronald Reagan in 1986, and the Research Awards from the College of Engineering of The OSU in 1993 and 1997, respectively. He and his students received the Best Student Paper or Best Conference Paper Awards several times, and received the Fred Diamond Award for Best Technical Paper from the Air Force Research Laboratory in 2006.