Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 2777–2788
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Applying the multi-category learning to multiple video object extraction

Yi Liu[a], Yuan F. Zheng[b,d,*], Xiaotong Shen[c]

[a]PIPS Technology, A Federal Signal Company, Knoxville, TN 37932, USA
[b]Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA
[c]School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA
[d]School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Received 3 May 2006; received in revised form 9 January 2008; accepted 19 February 2008

Abstract

Video object (VO) extraction is of great importance in multimedia processing. In recent years approaches have been proposed to deal with VO extraction as a classification problem. This type of methods calls for state-of-the-art classifiers because the performance is directly related to the accuracy of classification. Promising results have been reported for single object extraction using support vector machines (SVM) and its extensions. Multiple object extraction, on the other hand, still imposes great difficulty as multi-category classification is an ongoing research topic in machine learning. This paper introduces a new scheme of multi-category learning for multiple VO extraction, and demonstrates its effectiveness and advantages by experiments.
© 2008 Elsevier Ltd. All rights reserved.

Keywords: VO extraction; Multiple object tracking; ψ-Learning; Support vector machines (SVM); Multi-class classification

1. Introduction

Video object (VO) extraction, 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 deal of approaches have been proposed [2–10], which provide satisfactory results for extracting VOs of homogeneous motion characteristics. Unfortunately, dealing with VOs with abrupt motions or occlusions remains a challenge. In recent years classification-based approaches have been proposed to meet the challenge by handling object tracking as a classification problem [11–13]. Each VO is considered as a class, and VO extraction is achieved by classifying every pixel to one of the available classes. By doing so, temporal associations of objects between frames are automatically maintained through correct classifications which is therefore motion-assumption free. As a result, the approaches are more robust to complicated motion fluctuations.

What learning algorithm to use is key to the success of the classification-based approaches. By using powerful classifiers high classification accuracy can be achieved which leads to better performance for VO extraction. However, most of the results reported are limited to single object scenarios. In other words, only binary classification between the object and the background has been tackled. At the first glance, the extension from single object to multiple object extraction is straightforward since conceptually one only needs to replace the binary classifier with a multi-class classifier. Unfortunately, the implementation of such an extension is far more difficult than it appears because multi-category classification is still an ongoing and immature research topic itself in machine learning. Only recently have works emerged to offer new tools that can help tackle the multi-object problem. This work presents an attempt of such.

Over the last decade, margin-based classification technologies for which the best known example is support vector machines (SVM) [14] have drawn tremendous attention due to their theoretical merits and practical success. Instead of directly estimating the conditional probabilities, the margin-based classifiers focus on the decision boundary which; however, makes it difficult to generalize their applications from binary to multi-class scenarios.

"Single machine" and "error correcting" are two mainstreams for multi-class margin-based classification. As its name

* Corresponding author. Tel.: +1 614 292 8039; fax: +1 614 292 7596.
E-mail addresses: yi.liu@pipstechnology.com (Y. Liu), zheng@ece.osu.edu (Y.F. Zheng), xshen@stat.umn.edu (X. Shen).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.02.007

suggests, the "single machine" type of approaches attempts to construct a multi-class classifier by solving just a single optimization problem [15–19]. On the contrary, the "error correcting" type of approaches [20,21] works with a collection of binary classifiers, for which the primary goal is to determine what binary classifiers should be chosen to train and how to combine their classification results to make the final decision. Among all the methods published in the literature, "one-against-all", "one-against-one" and directed acyclic graph (DAG) [22] are most popular choices in solving real-world problems. A good overview of multi-class classification can be found in Refs. [23,24].

As a natural extension of binary large margin classification, the "single machine" type of approaches is intuitively appealing. It has drawn even more attention when certain formulations are reported to yield classifiers with consistency approaching the optimal Bayes error rate in the large sample limit [25]. Multi-class ψ-learning is such a learning algorithm [26]. Moreover, ψ-learning aims directly at minimizing the generalization error (GE), which is the reason why its binary version has shown significant advantage over SVM in terms of generalization both theoretically and experimentally [27]. The extended multi-class ψ-learning retains the desirable properties of its binary counterpart. In addition, a computational tool based on the recent advance in global optimization has been developed to reduce the time of training for the "single machine" [28].

The purpose of this paper is twofold. First, it introduces multi-category ψ-learning [26] to tackle the multiple VO extraction problem. Secondly, it reports the performance of the new learning algorithm on several MPEG-4 standard video sequences instead of synthetic data on which many multi-class learning algorithms are tested.

The rest of the paper is organized as follows. Section 2 gives an introduction of multi-class ψ-learning. Then a multiple VO extraction method using this new learning methodology is explained in Section 3. Section 4 provides the experimental results which are followed by conclusions in Section 5.

2. Multi-category ψ-learning

We first introduce the notations that will be used for the rest of the paper. In the framework of multi-category ψ-learning, the class label is coded as $y \in \{1, 2, \ldots, M\}$, and for a sample $x \in \mathbb{R}^d$ the decision rule is

$$y = \arg\max_{i=1,\ldots,M} f_i(x), \tag{1}$$

where M is the number of classes and f_i is the decision function of class i for $i = 1, \ldots, M$. For the linear classifier, we have $f_i(x) = w_i^{\mathrm{T}} x_i + b_i$ with $w_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$. Conventionally, the classifier is represented as $\mathbf{f} = (f_1, f_2, \ldots, f_M)$.

As a characteristic of multi-class problems, multiple comparisons between classes need to be performed. In order to simplify the notations an $(M-1)$-dimensional vector-valued function $g(x, y)$ and a multivariate sign function $\mathrm{sign}(u)$ where $u = (u_1, \ldots, u_{M-1})$ are defined as follows:

$$\begin{aligned} g(x, y) &= (f_y(x) - f_1(x), \ldots, f_y(x) - f_{y-1}(x), \\ &\qquad f_y(x) - f_{y+1}(x), \ldots, f_y(x) - f_M(x)), \\ \mathrm{sign}(u) &= \begin{cases} 1 & \text{if } u_{\min} = \min(u_1, u_2, \ldots, u_{M-1}) \geqslant 0, \\ -1 & \text{if } u_{\min} < 0. \end{cases} \end{aligned} \tag{2}$$

As mentioned before, the most prominent feature of ψ-learning is the direct consideration of GE. Defined as the probability of misclassification, GE yielded by an M-class classifier is

$$\mathrm{Err}(\mathbf{f}) = P\left[Y \neq \arg\max_{i=1,\ldots,M} f_i(X) \right].$$

It can be shown that with the notations of $g(x, y)$ and $\mathrm{sign}(u)$ GE can be rewritten as

$$\mathrm{Err}(\mathbf{f}) = \tfrac{1}{2} E[1 - \mathrm{sign}(g(X, Y))].$$

2.1. Multi-category ψ-learning

Seeking a vector $\mathbf{f}$ to minimize GE is the ultimate goal for any learning algorithm. For example, in the coding system described above,[1] the cost function of the well-known linear SVM can be rewritten as [26]

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \sum_{j=1}^{2} \|w_j\|^2 + C \sum_{i=1}^{N} F_{\mathrm{SVM}}(f_{y_i}(x_i) - f_{3-y_i}(x_i)), \\ &\text{subject to} \quad \sum_{j=1}^{2} f_j(x) = 0 \quad \text{for } \forall x, \end{aligned} \tag{3}$$

where N is the number of training samples and the sum-to-zero constraint is invoked to eliminate the redundancy in (f_1, f_2). The parameter C is a regularizer that controls the relative importance between the separating margin and the training error which are reflected in the quantities $\frac{1}{2}\sum_{j=1}^{2}\|w_j\|^2$ and $\sum_{i=1}^{N} F_{\mathrm{SVM}}$, respectively. Here the so-called hinge loss $F_{\mathrm{SVM}}(u) = 0$ if $u \geqslant 1$, and $2(1-u)$ if $u \leqslant 1$ is a convex upper envelope of $F_{\mathrm{GE}} = (1 - \mathrm{sign}(u))$. However, as shown in Fig. 1(a) and (b) there is significant difference between this convex envelope and $(1 - \mathrm{sign}(u))$ itself especially when $u < 0$, which corresponds to the inevitable misclassifications in non-separable cases. Motivated by this consideration, Shen et al. [26,27] proposes to replace F_{SVM} with a non-convex ψ function as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \sum_{j=1}^{2} \|w_j\|^2 + C \sum_{i=1}^{N} \psi_b(f_{y_i}(x_i) - f_{3-y_i}(x_i)), \\ &\text{subject to} \quad \sum_{j=1}^{2} f_j(x) = 0 \quad \text{for } \forall x. \end{aligned} \tag{4}$$

[1] Conventionally, the formulation of SVM is expressed in the coding system where the class label $y \in \{-1, 1\}$.

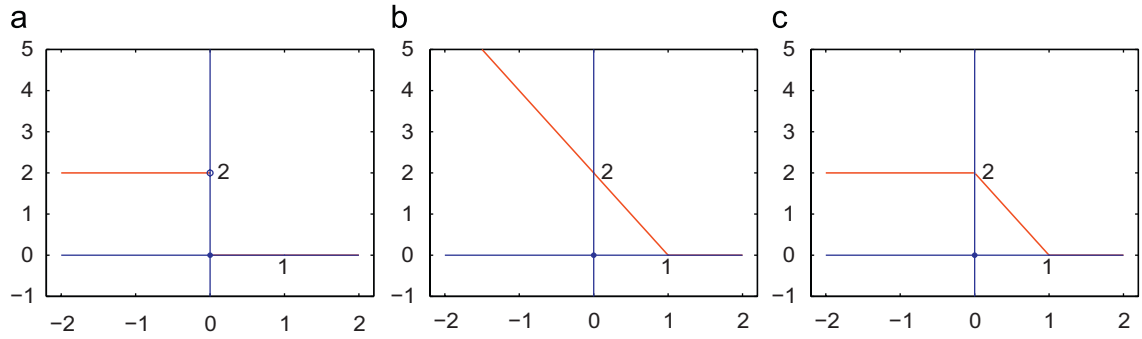

Fig. 1. F_{GE} function for GE, F_{SVM} function for SVM and ψ_b function for binary ψ-learning: (a) $F_{\mathrm{GE}} = 1 - \mathrm{sign}(u)$; (b) F_{SVM} function and (c) ψ_b function.

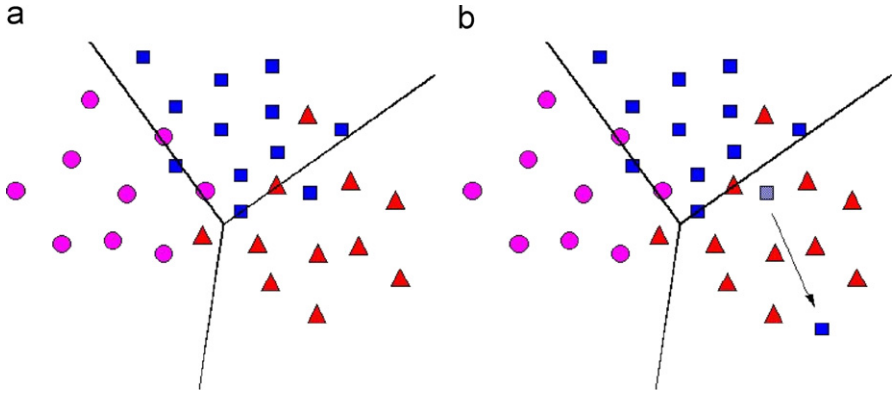

Fig. 2. Illustration of the robustness of multi-class ψ-learning in non-separable cases. (a) Shows a simple non-separable three-class training set. The classification boundaries are depicted as the solid lines. In (b) a misclassified sample, represented by the blue square, is moved much farther away from the boundaries and becomes an outlier. For learning algorithms that penalize the misclassification using the hinge loss function such as SVM, this change will affect the resulted classification boundaries a great deal because the penalty is proportional to the distance from the sample to the hyperplanes. ψ-learning, on the other hand, still counts it as a single misclassification as before, and consequently the boundaries are not much affected.

Here ψ_b can be any function satisfying $R \geqslant \psi_b(u) \geqslant 0$ if $u \in [0\ \tau]$ and $\psi_b(u) = 1 - \mathrm{sign}(u)$ otherwise, where $\psi_b(u)$ is non-increasing in u and $\tau \in (0\ 1]$. An example of such a function is shown in Fig. 1(c). Evidently because of the constant penalty for misclassification, ψ_b is much closer to $(1 - \mathrm{sign}(u))$ than F_{SVM}, which explains why ψ-learning is expected to deliver higher accuracy performance for the non-separable case. A graphical illustration is given in Fig. 2.

In analogy to Eq. (4) which is for binary classification, the multi-category ψ-learning is formulated as

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\sum_{j=1}^{M}\|w_j\|^2 + C\sum_{i=1}^{N}\psi(g(x_i, y_i)), \\ &\text{subject to} \quad \sum_{j=1}^{M} f_j(x_i) = \sum_{j=1}^{M}(w_i^{\mathrm{T}}x_i + b_i) = 0. \end{aligned} \tag{5}$$

Here again C is the regularizer while the ψ function is a multivariate version of ψ_b with $(M-1)$ arguments which is defined as

$$\begin{cases} R \geqslant \psi(u) > 0 & \text{if } u \in (0\ \tau_1] \times \cdots \times (0\ \tau_{M-1}], \\ \psi(u) = 1 - \mathrm{sign}(u) & \text{otherwise}, \end{cases} \tag{6}$$

where $0 < \tau_1, \ldots, \tau_{M-1} \leqslant 1$, and $\psi(u)$ is restricted to be non-increasing in each u_j for $u \in (0\ \tau_1] \times \cdots \times (0\ \tau_{M-1}]$. The multi-category ψ-learning preserves the desired properties of its binary counterpart. More specifically speaking, for any x satisfying $\mathrm{sign}(g(x, y)) = -1$, ψ assigns a constant penalty for the misclassification which is in the same spirit as GE. As a result, it is less sensitive to outliers and offers better learning ability. The cost, however, is the computational advantage since ψ is not a convex function any more. Fortunately the selection of the ψ function is relatively flexible. To utilize the difference convex (d.c.) decomposition which is a global optimization strategy, a specific ψ function

$$\psi(u) = \begin{cases} 0 & \text{if } u_{\min} \geqslant 1, \\ 2 & \text{if } u_{\min} < 0, \\ 2(1 - u_{\min}) & \text{if } 0 \leqslant u_{\min} < 1 \end{cases} \tag{7}$$

is chosen for implementations [26].

2.2. Theoretical advantage of multi-category ψ-learning

GE is the ultimate measure for any classifier, and the optimal performance of classification is achieved by the Bayes classifier $\mathbf{f} = (P_1(x), P_2(x), \ldots, P_M(x))$ with $P_j(x) = P(y = j|x)$ in the

sense that GE is minimized by $\bar{\mathbf{f}}$. In other words, we have

$$\mathrm{Err}(\mathbf{f}) \geqslant \mathrm{Err}(\bar{\mathbf{f}}) = \min \tfrac{1}{2} E[1 - \mathrm{sign}(g(X, Y))]. \quad (8)$$

For a learning algorithm, how to construct the function $\mathbf{f}$ in the absence of the knowledge of $P(X, Y)$ and how the resulted $\mathbf{f}$ statistically approaches the optimal performance $\mathrm{Err}(\bar{\mathbf{f}})$ are two equally important issues. The lack of statistical learning theories for multi-category classification manifests the immaturity of this area. Only recently theoretical analysis of margin-based classification has been investigated, most of which is focused on the asymptotical scenario. Therefore practical performances of multi-category approaches in general remain empirical and theoretically unclear. Fortunately a statistical learning theory has been developed for multi-category ψ-learning which provides insight of ψ-learning's performance with respect to the choice of tuning parameter C, the training size N as well as the number of classes M [26], and we summarize it as follows:

(1) ψ-Learning estimates the Bayes classifier $\bar{\mathbf{f}}$ as opposed to the conditional probability. However, the optimal classification performance of $\bar{\mathbf{f}}$ is realized via the $\psi(u)$ function which differs from $1 - \mathrm{sign}(u)$. In other words, as the number of training samples N goes to infinity, $\mathrm{Err}(\mathbf{f})$ approaches $\mathrm{Err}(\bar{\mathbf{f}})$ asymptotically.
(2) The non-asymptotic rate of convergence of $e(\mathbf{f}, \bar{\mathbf{f}}) = \mathrm{Err}(\mathbf{f}) - \mathrm{Err}(\bar{\mathbf{f}})$ is investigated. For example, it is shown that the convergence rate decreases as the number of classes M increases although the order remains the same for finite M. It provides insight of the performance difference between ψ-learning and the optimal Bayes classifier under practical circumstances.
(3) Unlike the binary case, the optimal performance of linear learning may not be achieved at large C for multi-category problems.

For details of the theory, the readers are referred to Ref. [26].

3. Multi-object extraction using multi-category ψ-learning

3.1. Related work

Filtering and association and *representation and localization* are two major techniques for object tracking [29]. Rooted in the control theory, the former technique deals with the dynamics of the objects while the latter heavily relies on image processing technologies. The way these two techniques are combined and weighted is application dependent. For example, the filtering and association method prevails in the application of aerial video surveillance because the motion of the objects is the major concern. For content-based video processing, on the other hand, objects of interest are usually heterogeneous in spatial features, non-rigid in the temporal domain, yet rich of visual information. For this reason representation and localization is the technique used most in VO extraction, and therefore the emphasis of this paper.

For VO extraction using the *representation and localization* technique, a reference model representing the object must first be created which can be done either in an automatic [2–7] or semiautomatic fashion [8–10]. A variety of models has been proposed including: 2D mesh [30,31], binary model [32], color histogram [29], deformable templates [33], corners and lines [34], active contour [9], 2D regions [35], etc. To localize the object in subsequent frames, a typical approach is to place the model to its possible positions and locate it where the best match is found. To measure the quality of the match between the model and the object candidates, a similarity function is defined, which traditionally considers only the information of the object such as spatial similarity and temporal consistency.

The importance of integrating the background information in the matching process is demonstrated in Ref. [29]. More specifically it takes into account the dissimilarity between the object and the background by down-weighting the colors that appear in both classes in the similarity function such that the object is represented only by the salient parts. Avidan [13] extends this idea by explicitly treating tracking as a classification problem. Single object tracking, for example, requires identifying each pixel as object or background, and therefore can be formulated as a binary classification problem. In the meantime, the same spirit appears in Refs. [11,12] for the task of VO extraction where the tracking results are required to be pixel-wise accurate.

In spite of the differences in detailed algorithms, these classification-based approaches are encompassed in a generic four-step model: (1) construction of feature vectors, (2) training of classifiers, (3) classifications applied to the new frame, and (4) object generation. The first step is to design a feature representation for every pixel. It may be the raw chromatic values such as RBG [11], the histogram of colors [13], or the coefficients of the DCT transform of the block centering at the pixel [12]. Then in the second step a classifier is trained and the classification function is obtained to discriminate the pixels that belong to the object from those that belong to the background. Different classifiers have been attempted such as neural networks [11], ψ-learning [12] or even an ensemble of linear classifiers [13]. The third step is to evaluate the classification function at every pixel in the subsequent frames, and the final step is to generate the tracked object based on the classification results for which the way of implementation varies. For example, in Ref. [13] a so-called confidence map is first produced according to the classification results, and tracking is then realized by locating the object where the peak of the confidence map occurs. The output of the tracker, however, is a rectangle that tightly encloses the object of interest. For the task of VO extraction the fourth step can even be skipped [11] since after the classification step we already know for every pixel if it belongs to the object. However, for efficiency purpose it is not necessary to do the classification pixel-by-pixel. By exploiting the spatial redundancy, we introduce the block-level classification instead and design a pyramid refining scheme to refine the boundary in an efficient and scalable manner [12].

Accuracy and complexity are two critical issues for VO extraction which have to be traded off in practice, and the major advantage of the classification-based methods is the potential to

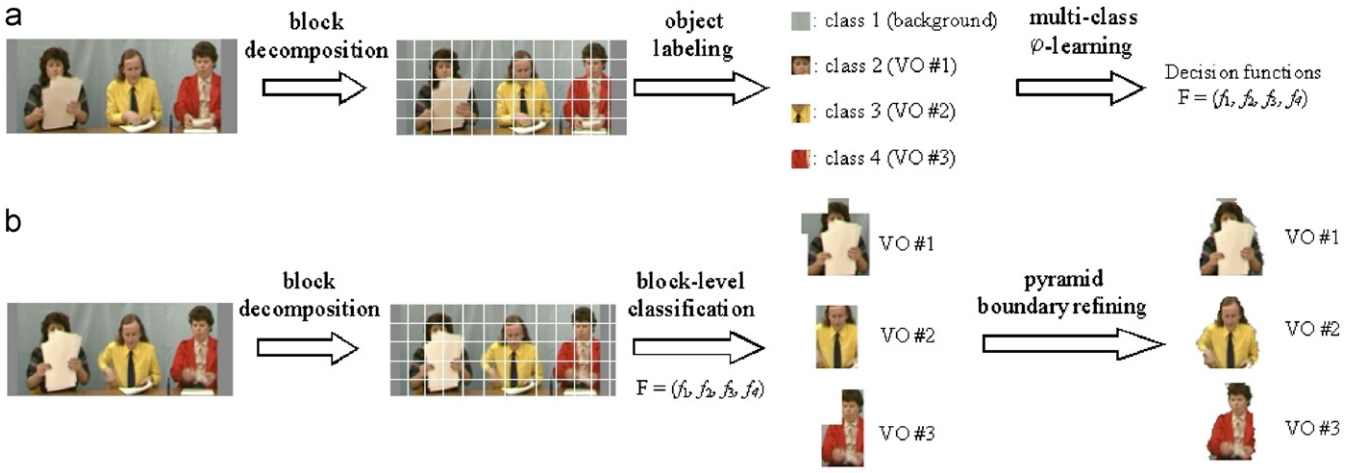

Fig. 3. An overview of the proposed approach for multiple VO extraction. (a) The training phase and (b) the tracking phase.

achieve both. The methods are accurate because powerful classifiers are designed for the purpose of 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., $w^{\mathrm{T}}x + b$ for linear SVM while the time-consuming processes of object modeling, extracting and searching are circumvented.

3.2. The approach

As mentioned before, the choice of the learning algorithm is key to the success of the current approach because the performance of the algorithm is directly related to the classification accuracy. Considering the single VO extraction as an example, the background and the object are often not separable. As pointed out in Section 2, ψ-learning aims at the minimization of GE and therefore has the advantages in non-separable cases. For this reason, a method for single VO extraction that employs binary ψ-learning as the classifier is proposed in Ref. [12]. To tackle the challenging task of multi-object extraction, multi-category ψ-learning has to be employed.

As shown in Fig. 3, our approach is semiautomatic and consists of two phases: the training phase and the tracking phase. At the training phase the user manually outlined the objects of interest in the first frame, and at the tracking phase the proposed approach tracks and extracts the objects automatically for the rest of the video sequence. For classification at the pixel level, individual pixels are represented by pixel-wise color or intensity information, which; however, would result in misclassifications due to the negligence of the spatial relationship among pixels. Another concern is the size of the training set. If every pixel is included, it would contain too many training samples to yield a quick training especially when the frame size is large. The same efficiency issue exists if we do the pixel-by-pixel classification in the tracking phase. Fortunately, in most video sequences there is abundant spatial correlation that we can take advantage of to make the approach more efficient. Let p denote a pixel and $N(p, d)$ the set of pixels within a small distance d from p. Due to the spatial correlation of images, the class labels as well as the feature vectors of p and $N(p, d)$ tend to be similar to each other. Based on this observation, we introduce the concepts of *object blocks* and *background blocks*, and suggest the representation and classification to be done at the block level as follows.

Suppose we have M VOs of interest. The training phase begins with dividing the first frame, chosen as the training frame, into $(M+1)$ types of blocks (the number of different VOs plus background) depending on which object or background the pixel at the center of the block belongs to. In the standard video processing algorithms such as MPEG, 8×8 has been a common choice for the block size. In our approach, an odd number of pixels is preferred. That is because the intended classification is at the pixel level and the classification of each pixel is performed by classifying the block in which the pixel is at the center, i.e., the pixel to be classified is the centering pixel of the block. Consequently, the block size is chosen as 9×9 which is closest to 8×8, and evidently the number of blocks determines the size of the training set.

We use the same method as in Ref. [12] to represent each block as well as the centering pixels. Namely, discrete cosine transform (DCT) is first applied to each block and then based on the DCT coefficients $c(i, j)$ the local and neighboring features are constructed for each block:

$$\vec{f}_{local} = (f_0, f_1, f_2, f_3)^{\mathrm{T}} = \begin{pmatrix} c(0,0) \\ \sqrt{\sum_{j=1}^{8} c(0,j)^2} \\ \sqrt{\sum_{i=1}^{8} c(i,0)^2} \\ \sqrt{\sum_{i=1}^{8}\sum_{j=1}^{8} c(i,j)^2} \end{pmatrix}. \quad (9)$$

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

The neighboring features $\vec{f}_{neighbor}$ are extracted from neighbors which are eight 9×9 blocks that are adjacent to the

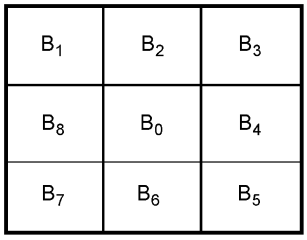

Fig. 4. Eight-connected neighboring blocks of block B_0.

block under study in the vertical, horizontal and diagonal directions as shown in Fig. 4. With $avg(B_i)$ denoted as the average intensity of block B_i we compute the neighboring features as

$$\vec{f}_{neighbor} = \begin{pmatrix} avg(B_1 + B_2 + B_3) \\ avg(B_3 + B_4 + B_5) \\ avg(B_5 + B_6 + B_7) \\ avg(B_7 + B_8 + B_1) \end{pmatrix}. \quad (10)$$

The calculations given above only consider the grayscale information. When the video sequence is chromatic, we compute Eqs. (9) and (10) for each color component and then concatenate the vectors, respectively, to form the chromatically local and neighboring features. As a result, if the video sequence is chromatic, the dimensions of the feature vectors increase. That will prolong the training time of the classifier, but the trained classifier will become more effective as more information is included in the feature vectors.

Now with the training data in place, the next step is to train the machine by solving the optimization problem Eq. (5), which yields $(M + 1)$ decision functions that separate the M objects as well as the background.

In the tracking phase each subsequent frame is also divided into 9×9 blocks, and for each block the $M + 1$ decision functions are evaluated to decide what object the centering pixel belongs to, which consequently determines the class label of the block. Then the tracking mask of every object is formed by the blocks that have been classified in the corresponding class. At this point the resolution of object's boundary is as large as the size of the block, but by applying a *pyramid boundary refining algorithm* [12] the object boundary can be refined and the pixel-wise accuracy can be achieved. The details of the latter algorithm can be found in Ref. [12].

4. Experimental results

In this section we apply the proposed multiple VO extraction method to three standard MPEG-4 test video sequences, which exhibit varieties of temporal and spatial characteristics. These sequences are *Students*, *Trevor* and *Sun Flower Garden*, respectively. The performance comparisons are made between multi-category ψ-learning and three popular multi-class algorithms, namely one-vs-all, one-vs-one and DAG [22]. The performance of one non-classification based-method is also presented to show the robustness of the proposed method.

Table 1
The average run time

	Student	*Trevor*	*Sun Flower Garden*
# of classes	3	4	3
Frame size	144 × 176	72 × 176	120 × 176
Average (s)	1.27	1.01	0.62

4.1. Computational complexity

During the training phase, the unconstrained optimization algorithm proposed in Ref. [26] is adopted to minimize the cost function of Eq. (5). The parameter C in Eq. (5) is empirically chosen as C=0.25. All experiments are carried out on a Pentium IV 2.5 GHz PC and the average execution time per frame is shown in Table 1.

The number of VOs and the frame size are two critical factors that determine the execution time of the algorithm. Assume there are M VOs classes, L blocks in each frame, and each block is represented by a d-dimensional feature vector x. In the tracking phase we need to evaluate $M + 1$ functions (M functions for VOs and one for background) $f_i = w_i^{\mathrm{T}} x + b_i$, each of which performs d multiplications to determine the class label of a block. As a result, the computational complexity is $L(M + 1)d = O(M)$, which is a linear function of the number of objects M and gives the approach low complexity and good scalability. After the block-level classification, boundary refining is executed to generate the VOs with pixel-level accuracy. So the total complexity is $LMd + T_{\mathrm{r}}$, where T_{r} denotes the computations consumed by the boundary refining algorithm that is proportional to the size of the VOs.

4.2. Subjective evaluations

The first one to test is *Students*. As the major content of this sequence, the two students are chosen as two objects of interest, and along with the background this is a three-class classification problem. As one can see from the original frames shown in columns (a) and (d) of Fig. 5, *Students* is a typical sequence of slow but heterogeneous motion. For example the male student turns the head and moves his hands while his body stays still most of the time. The extracted objects are shown in columns (b), (c), (e) and (f), respectively. One can see that the proposed method works well, which discriminates the body parts of the students as well as their faces. The latter is not an easy task since the skin color is very similar between the two students.

Another sequence containing three people is also tested, and the three people are considered as three objects which makes it a four-class classification problem. The original frames and the extracted objects are shown in Fig. 6. Unlike the *Students* sequence, the objects in this sequence change the appearance a great deal. Taking the lady who sits at the farthest right as an example, her face changes from frontal to left-side view. Besides, the man in the middle is originally seated but finally standing. As seen in Fig. 6, the main body of the objects are successfully extracted although the boundaries of the objects are not perfectly separated due to classification errors.

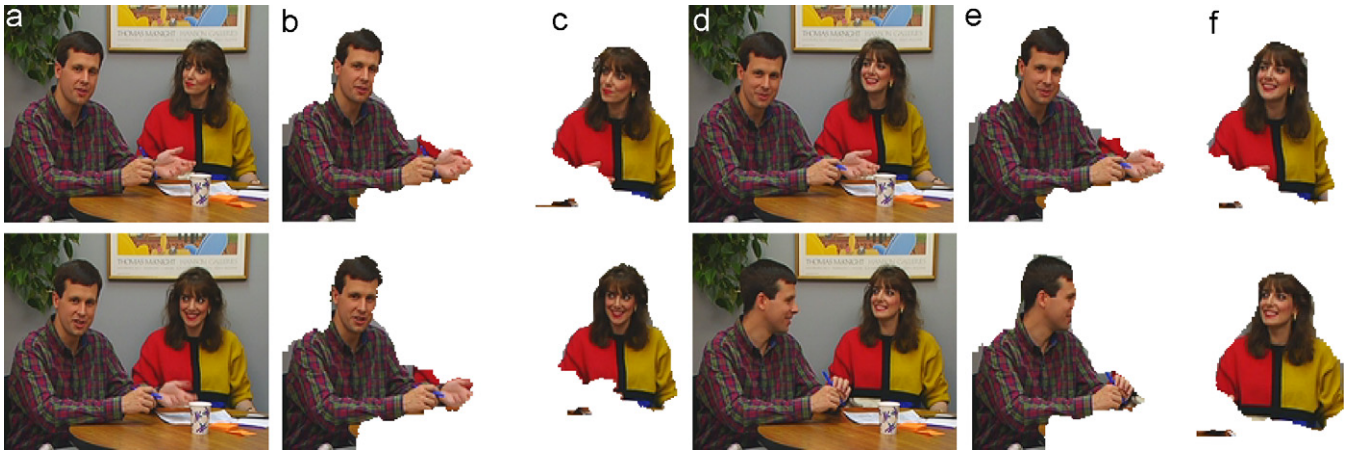

Fig. 5. The extraction performance of *Students*. Columns (a) and (d) display the original frames. Columns (b) and (e) display the extracted VO #1. Columns (c) and (f) display the extracted VO #2.

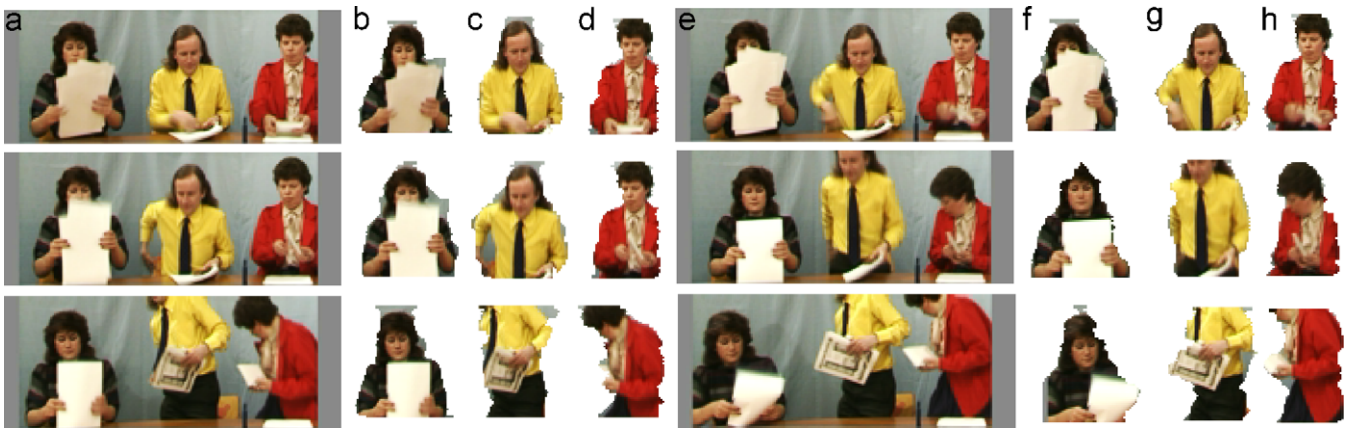

Fig. 6. The extraction performance of *Trevor*. Columns (a) and (e) are the original frames. Columns (b) and (f) are the extracted VO #1. Columns (c) and (g) are the extracted VO #2. Columns (d) and (h) are the extracted VO #3.

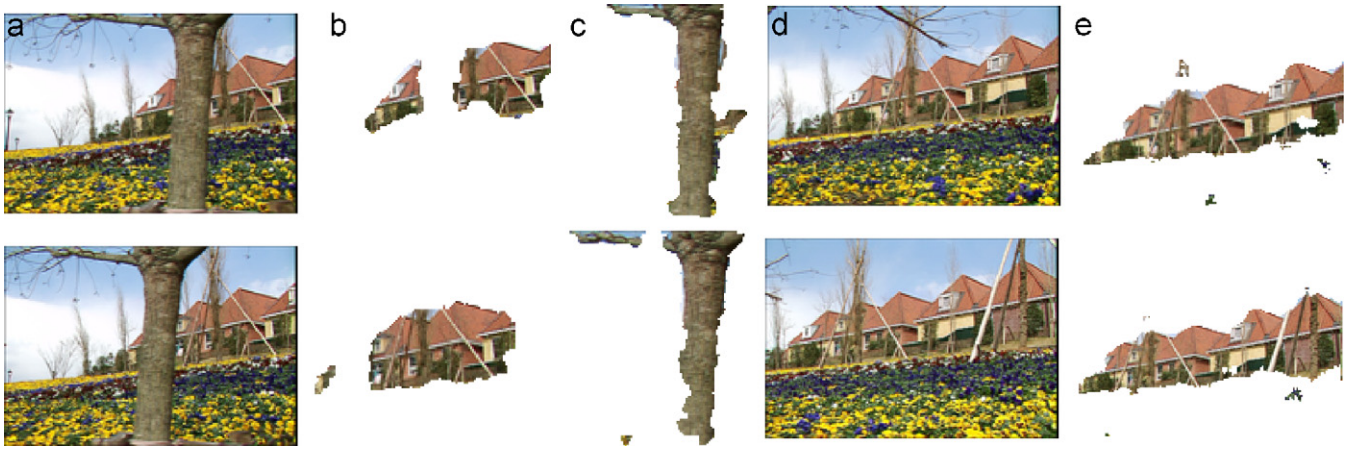

Fig. 7. The extraction performance of *Sun Flower Garden*. Columns (a) and (d) are original frames. Columns (b) and (e) are the extracted VO #1. Column (c) is the extracted VO #2.

Among the sequences tested in the experiments, *Sun Flower Garden* is most challenging. Different from the previous video-conference kind of sequences, it displays a natural scene that is rich of colors and textures with a non-stationary camera. There are two objects of interest: the house and the tree. For the first few frames, the house is occluded by the tree. Two

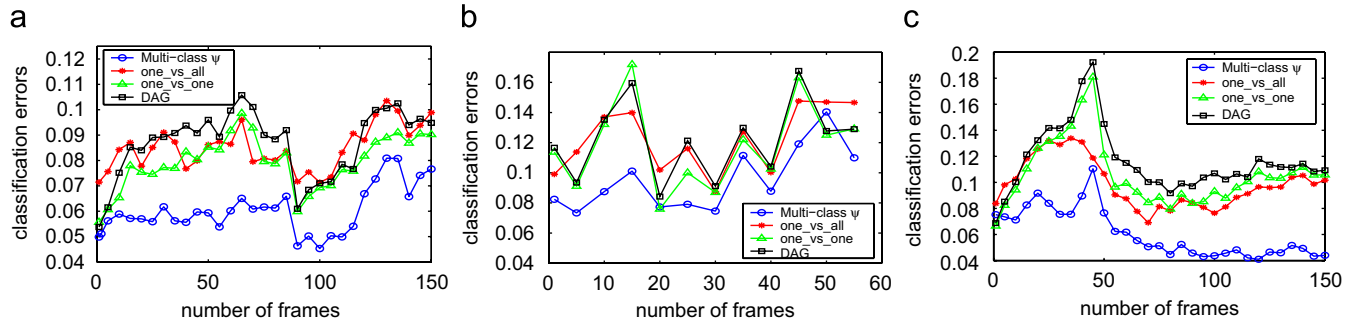

Fig. 8. The comparison of classification errors between multi-category ψ-learning, one-vs-all, one-vs-one and DAG. SVM is the underlying binary classifier employed by one-vs-all, one-vs-one and DAG. (a) *Students*, (b) *Trevor* and (c) *Sun Flower Garden*.

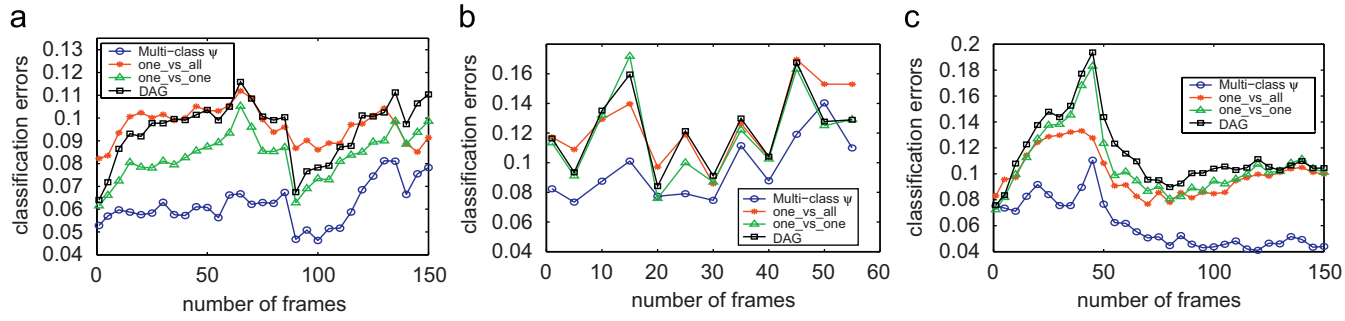

Fig. 9. The comparison of classification errors between multi-category ψ-learning, one-vs-all, one-vs-one and DAG. Binary ψ-learning is the underlying binary classifier employed by one-vs-all, one-vs-one and DAG. (a) *Students*, (b) *Trevor* and (c) *Sun Flower Garden*.

of such frames are shown in column (a) of Fig. 7, and the two extracted objects (house and tree) by using ψ-learning are shown in columns (b) and (c), respectively. With the camera moving, the tree shifts toward the left side of the frame and finally disappears as in column (d). From that point on, only the house can be extracted by the proposed method as shown in the last column of Fig. 7.

4.3. Performance comparison

For their simplicity and effectiveness, one-vs-all, one-vs-one and DAG are three widely used multi-category algorithms. Suppose we have M classes. One-vs-all constructs M binary classifiers $f_i^{\mathrm{OVA}}(x)$ with the ith one separating class i from all the remaining classes. One-vs-one and DAG, on the other hand, construct $M(M-1)/2$ decision functions $f_{i,j}(x)$, each of which is responsible for the binary classification task between class i and j. At the classification step, one-vs-all classifies a sample x to the class for which $f_i^{\mathrm{OVA}}(x)$ produces the highest value while one-vs-one follows a voting strategy. As for DAG, it builds a DAG using the $M(M-1)/2$ binary classifiers as the internal nodes. The classification is achieved by going through a path from the root of the graph to a leaf node which indicates the predicted class [22].

The classification error is the metric employed to compare the performances of different learning schemes. So now we compute the classification errors to see how multi-category ψ-learning performs against these three popular methods. To do so, we first manually outlined the objects of interest in each frame, which serves as the ground truth to compare with the objects extracted by the proposed approach. The classification errors are then computed as the number of wrongly classified pixels divided by the number of pixels per frame.

In Fig. 8, the classification errors yielded by all the four methods are displayed every five frames where SVM is the underlying binary classifiers. For the training of each SVM, the classification accuracy is estimated by testing different values of $C \in [2^{12}, 2^{11}, \ldots, 2^{-2}]$, and the best one is chosen for the performance comparison. As one can see, for all the three sequences multi-category ψ-learning achieves the lowest classification errors almost for every test frame. Although the training is conducted only once by using the first frame, the superior generalization ability of multi-category ψ-learning enables it to survive nearly the whole sequence.

Multi-category ψ-learning as one sees in its formulation (Eq. (5)) has two methodological features: (1) direct consideration of GE by using the ψ function and (2) collective consideration of all the classes at once. In order to further show the advantage of the second feature, we conduct another series of experiments which replaces SVM with binary ψ-learning in the one-vs-all, one-vs-one and DAG methods. The errors are compared in Fig. 9. As mentioned before, one-vs-one and DAG are based on pairwise classification, for which smaller sample sets are used to train each classifier [18]. One-vs-all, similar to multi-category ψ-learning, does utilize all the training sam-

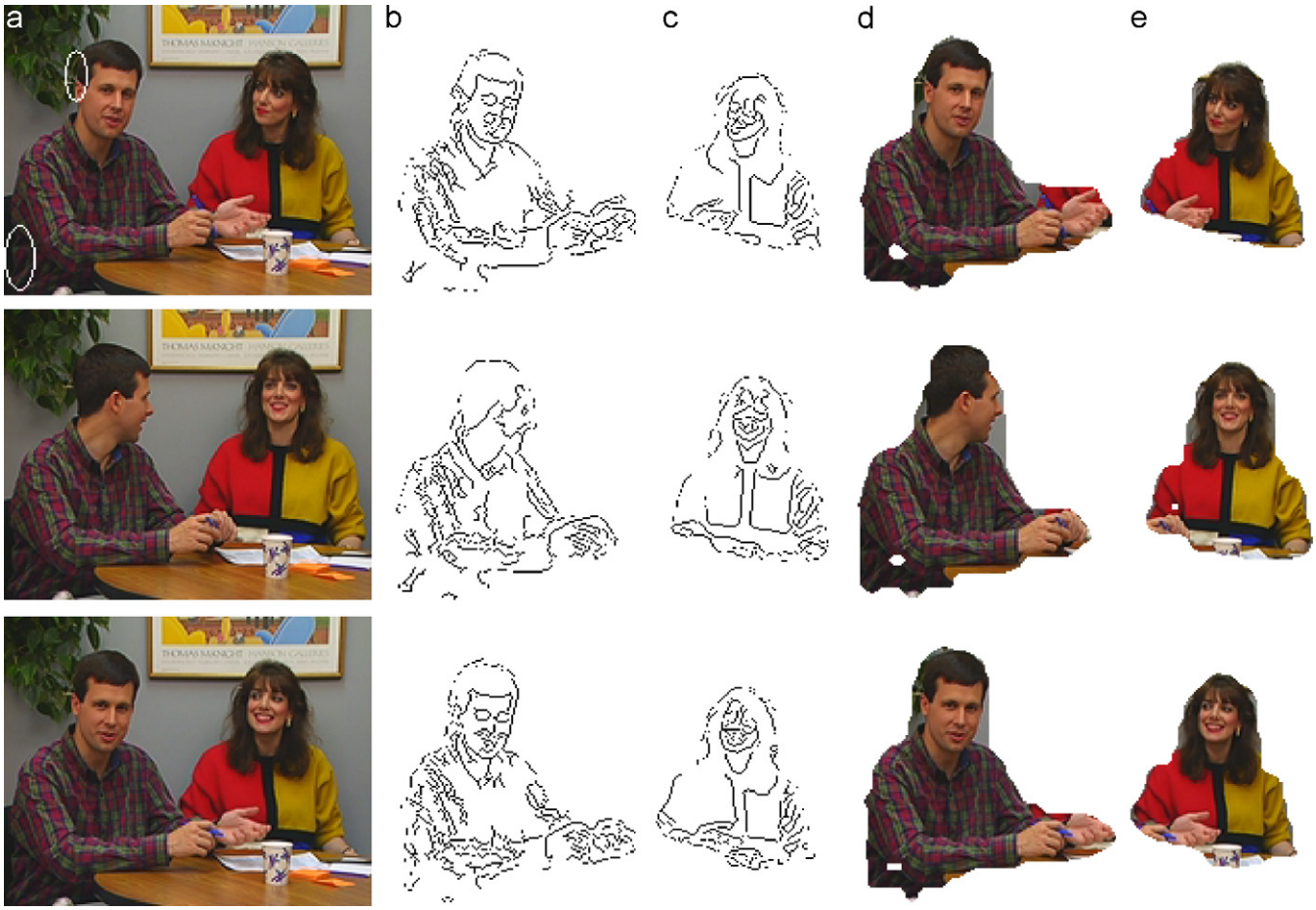

Fig. 10. The extracted performance using 2D binary model. (a) The original frame. (b) The binary model of VO #1. (c) The binary model of VO #2. (d) The extracted VO #1. (e) The extracted VO #2.

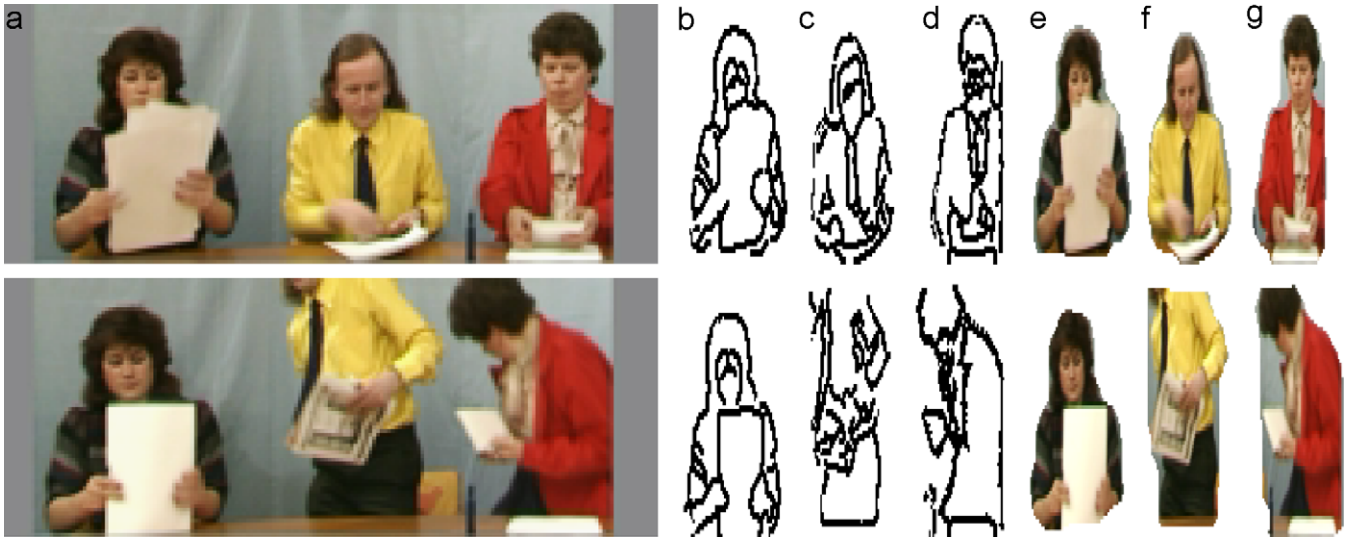

Fig. 11. The extracted performance using 2D binary model. (a) The original frame. (b) The binary model of VO #1. (c) The binary model of VO #2. (d) The binary model of VO #3. (e) The extracted VO #1. (f) The extracted VO #2. (g) The extracted VO #3.

ples to train, but the class mutual exclusiveness is overlooked. Thus the performance of one-vs-all degrades when there does not exist a dominating class in the sense that the conditional probability of each class is less than 0.5 [12,18]. In contrast, all the mutual information among classes is considered at once in multi-category ψ-learning, which is the main reason that it

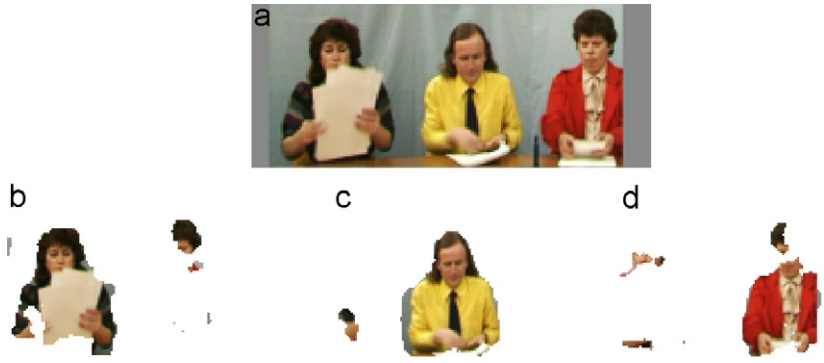

Fig. 12. The extracted performance by using block size 5×5. (a) Frame 2, (b) VO #1, (c) VO #2 and (d) VO #3.

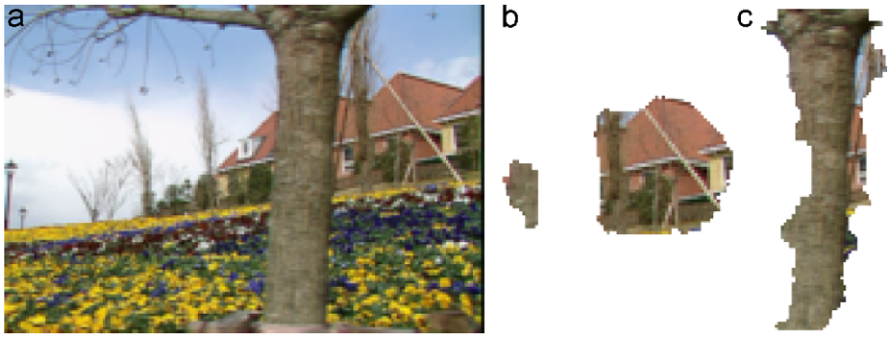

Fig. 13. The extracted performance by using block size 17×17. (a) Frame 3, (b) VO #1 and (c) VO #2.

beats the other three methods by large margin even when binary ψ-learning has been employed as the base classifier in this series of experiments.

It can be observed in Fig. 8(c) and 9(c) that after frame 50, the classification errors yielded by all the four methods drop significantly in *Sun Flower Garden*. This is because as the tree shifts toward the left side of the frame and finally disappears in around frame 50, the original three-class problem reduces to a binary-class problem. With one fewer class to differentiate, the learning approach is able to yield higher classification accuracy.

So far the performance comparison is done within the group of classification-based approaches by employing different multi-class classifiers. To further understand the strength of the proposed method, we now present the performance of a non-classification-based approach developed by Meier et al. [32]. This approach is selected for the following reasons:

(1) It belongs to the "representation and localization" category, which is the same technique as we use here.
(2) Meier's approach is illustrated using the single object model, but its extension to the scenarios of multiple objects is straightforward.
(3) It has few parameters to tune which makes it easy to implement and fair to compare.

Assuming a stationary background, the core algorithm of Meier's approach is to represent each VO by a 2D binary model and localize the VO by matching the model against subsequent frames using the Hausdorff distance. After the best match is found in every frame, the model is updated to accommodate for the change in shape. The initial model, which is derived automatically in the paper, is provided manually in our implementation for the first frame as it is done in our approach. Fig. 10 shows the extracted VO of sequence *Students* when Meier's approach is applied.

Because the binary model consists of the edge pixels of the object, it has difficulty locating the boundary of the object where the edge information is not strong enough or the textured background is present such as the two regions circled in the first row of Fig. 10(a). As a result, the lower left part of the male student's body is missing from the object while part of the background (the leave area around his head) is included in the extracted results. For the same reason, this approach performs well for sequence *Trevor*, of which the background is very simple and flat as shown in Fig. 11. We cannot obtain the results of sequence *Sun Flower Garden* because, unlike *Students* and *Trevor*, *Sun Flower Garden* is a sequence with non-stationary background due to the motion of camera which breaks the assumption of stationary background. In order to handle this type of video sequences, Meier et al. [32] states that a different method for moving background filtering other than the one presented in the paper is necessary.

How to extract the corresponding VO based on the binary model of the current frame is another critical step for this approach, or more generally the approaches using edge-represented models. One possible way is to find the first and

last edge pixels of each row and assign all the pixels in between as to VO. Then the same procedure is repeated for each column [32]. It works well when the contour of the VO is convex. Otherwise, part of the background will be assigned as the VO. Such an example is shown in the second row of Fig. 11(f) and (g).

As demonstrated before, our approach works effectively on all the three video sequences and all the VOs of different shapes, which shows its advantage in terms of robustness.

It is worth pointing out that the original approach by Meier is for single VO extraction, and we simply extend it to deal with multiple VOs by matching the corresponding models individually. Therefore, due to the errors introduced in the process of model updating it is possible for some areas to appear in multiple VOs, especially when they overlap. For instance, the left hand of the male student is included in the extracted male as well as female students. This problem evidently demonstrates that multiple VO extraction is not just a straightforward extension from its binary counterpart and special handling is necessary.

4.4. Effect of the block size

The objects are represented and classified at the block-level, and for the reason stated in 3.2 the block size has been fixed as 9×9 during the experiments. Evidently the choice of block size affects the performance. As the block size goes smaller, the features extracted from the block contain less distinguishing information of the blocks, which will degrade the performance of the classification. This effect can easily be perceived in the extreme when the block is sized down to a single pixel, which is not discriminating enough to represent different classes.

We performed an experiment on smaller blocks to test the effect of the reduced size. Some results obtained by using 5×5 on *Trevor* is shown in Fig. 12, where the hair of VO #3 is misclassified as part of VO #1, and parts of VO #1 and #3 are misclassified as VO #2, all due to the reduction of the block size. To the contrary, the blocks can be as big as the whole object, which will defeat the purpose of the current approach since the training of classifiers needs multiple samples of the same classes. Generally speaking, by incorporating more information of the spatial structures around the pixels, large blocks are more distinguished from object to object. The downside is the higher computation load and the lack of robustness to occlusions. For example, when we test the block size of 17×17 on the sequence *Sun Flower Garden* on frame 3, which is only 2 frames later than the training frame, we see in Fig. 13 that a significant portion of the house has already been misextracted. According to our experience, the size 9×9 works well for the three video sequences we experiment with, but it would not be a universal choice.

5. Conclusions

VO extraction is of great importance for content-based video analysis, and a great deal of research has been performed for single object extraction. Unfortunately, multi-object scenario which is more realistic and imposes a much greater challenge. Following the idea that handles VO extraction as a classification problem, this paper aims to tackle multiple object extraction by solving a multi-class classification problem using multi-category ψ-learning which is a newly developed learning algorithm for classification. The performances of other three popular multi-category classifiers as well a non-classification-based approach are compared against ψ-learning, which shows the advantage of this new learning machine. The proposed method is of low computational complexity which scales well when the number of objects increases.

It can be observed that even when the camera is in motion and the training is only done once, the tracking results are still of good quality. We believe that this is because there is no significant change of the video content 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.

It is also worth pointing out that the proposed approach relies only on the spatial information. Video sequences, however, provide temporal information which is useful for objects and background separation. Therefore one interesting research topic is to take advantage of the temporal redundancy between frames to further improve the efficiency and accuracy of the proposed algorithm.

Acknowledgments

This work was supported in part by the US National Science Foundation under Grant IIS-0328802, and in part by the Chinese Natural Science Foundation under Grant 60632040.

References

[1] D.G. Prerz, C. Gu, M.T. Sun, Semantic video object extraction using four-band watershed and partition lattice operators, IEEE Trans. Circuits Syst. Video Technol. 11 (5) (2001) 603–618.
[2] D. Wang, Unsupervised video segmentation based on watersheds and temporal tracking, IEEE Trans. Circuits Syst. Video Technol. 8 (5) (1998) 539–546.
[3] I. Kompatsiaris, M.G. Strintzis, Spatialtemporal segmentation and tracking of objects for visualization of videoconference image sequences, IEEE Trans. Circuits Syst. Video Technol. 10 (8) (2000) 1388–1403.
[4] Y. Deng, B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, IEEE Trans. Pattern Anal. Mach. Intell. 23 (8) (2001) 800–810.
[5] A. Neri, S. Colonnese, G. Russo, P. Talone, Automatic moving objects and background separation, Signal Process. 66 (2) (1998) 219–232.
[6] C. Kim, J.N. Hwang, Fast and automatic video object segmentation and tracking for content-based applications, IEEE Trans. Circuits Syst. Video Technol. 12 (2) (2002) 122–129.
[7] S.Y. Chien, S.Y. Ma, L.G. Chen, Efficient moving object segmentation algorithm using background registration technique, IEEE Trans. Circuits Syst. Video Technol. 12 (7) (2002) 577–586.
[8] C. Gu, M.C. Lee, Semiautomatic segmentation and tracking of semantic video objects, IEEE Trans. Circuits Syst. Video Technol. 8 (5) (1998) 572–584.
[9] S. Sun, D.R. Haynor, Y. Kim, Semiautomatic video object segmentation using VSnake, IEEE Trans. Circuits Syst. Video Technol. 13 (1) (2003) 75–82.
[10] C. He, J. Dong, Y.F. Zheng, S.C. Ahalt, Object tracking using the Gabor wavelet transform and the golden section algorithm, IEEE Trans. Multimedia 4 (4) (2002) 528–538.

[11] A. Doulamis, N. Doulamis, K. Ntalianis, S. Kollias, An efficient fully unsupervised video object segmentation scheme using an adaptive neural-network classifier architecture, IEEE Trans. Neural Networks 14 (3) (2003) 616–630.
[12] Y. Liu, Y.F. Zheng, Video object segmentation and tracking using ψ-learning classification, IEEE Trans. Circuits Syst. Video Technol. 15 (7) (2005) 885–899.
[13] S. Avidan, Ensemble tracking, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, June 2005, pp. 20–25.
[14] C. Cortes, V.N. Vapnik, Support vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[15] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[16] J. Weston, C. Watkins, Multi-class support vector machines, Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, 1998.
[17] Y. Lin, Support vectors machines and the Bayes rule in classification, Data Min. Knowl. Discovery 6 (2002) 259–275.
[18] Y. Lee, Y. Lin, G. Wahba, Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, J. Am. Stat. Assoc. 99 (465) (2004) 67–81.
[19] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res. 2 (2001) 265–292.
[20] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–286.
[21] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.
[22] J.C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin DAG's for multiclass classification, Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, MA, 2000, pp. 547–553.
[23] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[24] C. Hsu, C. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Networks 13 (2) (2002) 415–425.
[25] T. Zhang, Statistical analysis of some multi-category large margin classification methods, J. Mach. Learn. Res. 5 (2004) 1225–1251.
[26] Y. Liu, X. Shen, H. Doss, Multicategory ψ-learning, J. Am. Stat. Assoc. 101 (474) (2006) 500–509.
[27] X. Shen, G. Tseng, X. Zhang, W.H. Wong, On ψ-learning, J. Am. Stat. Assoc. 98 (463) (2003) 724–734.
[28] Y. Liu, X. Shen, H. Doss, Multicategory ψ-learning and support vector machine: computational tools, J. Comput. Graphical Statist. 14 (1) (2005) 219–236.
[29] D. Comanicu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 564–577.
[30] Y. Altunbasak, A.M. Tekalp, Occlusion-adaptive, content-based mesh design and forward tracking, IEEE Trans. Image Process. 6 (9) (1997) 1270–1280.
[31] P.V. Beek, A.M. Tekalp, N. Zhuang, I. Celasun, M. Xia, Hierarchical 2D mesh representation, tracking and compression for object-based video, IEEE Trans. Circuits Syst. Video Technol. 9 (2) (1999) 353–369.
[32] T. Meier, K.N. Ngan, Automatic segmentation of moving objects for video object plane generation, IEEE Trans. Circuits Syst. Video Technol. 8 (5) (1998) 525–538.
[33] Y. Zhong, A.K. Jain, M.P. Dubuisson-Jolly, Object tracking using deformable templates, IEEE Trans. Pattern Anal. Mach. Intell. 22 (5) (2000) 544–549.
[34] H. Wang, M. Brady, Real-time corner detection algorithm for motion estimation, Image Vision Comput. 13 (1995) 695–703.
[35] Y. Tsaig, A. Averbuch, Automatic segmentation of moving objects in video sequences: a region labeling approach, IEEE Trans. Circuits Syst. Video Technol. 12 (7) (2002) 597–612.

About the Author—YI LIU received the Ph.D. degree in Electrical and Computer Engineering from The Ohio State University, Columbus, Ohio in 2006. Her B.S. and M.S. degrees were received from the Department of Information Science and Electronic Engineering, 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.

About the Author—YUAN F. ZHENG 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, Columbus, Ohio in 1980 and 1984, respectively.
From 1984 to 1989, he was with the Department of Electrical and Computer Engineering at Clemson University, Clemson, South Carolina. Since August 1989, he has been with The Ohio State University, where he is currently Winbigler Professor of Electrical and Computer Engineering. He is on leave at the Shanghai Jiao Tong University, Shanghai, China in the academic year 2004–2005 and continues his participation and collaboration there. His research interests include two aspects. One is in *wavelet transform* for image and video compression for internet and satellite communications. Current efforts focus on content-based compression, 3D wavelet transformation and video object tracking. The other is in robotics which includes robots for biological applications, multiple robots coordination, legged robots, human-robot coordination and personal robotics. He is currently on the Editorial Board of *International Journal of Multimedia Tools and Applications*, on the Editorial Board of *Autonomous Robots*, an associated editor of the *International Journal of Intelligent Automation and Soft Computing*, on the Editorial Board of *International Journal of Intelligent Control and Systems* and on the Editorial Board of *International Journal of Control, Automation, and Systems*.
Professor Zheng was Vice-President for Technical Affairs of the IEEE Robotics and Automation Society from 1996 to 1999. He was an associate editor of the IEEE Transactions on Robotics and Automation between 1995 and 1997. He was the Program Chair of the 1999 IEEE International Conference on Robotics and Automation, held in Detroit, MI, on May 10–15, 1999. Professor Zheng received the Presidential Young Investigator Award from President Ronald Reagan in 1986.

About the Author—XIAOTONG SHEN is Professor of Statistics, University of Minnesota. He received Ph.D. degree in Statistics from University of Chicago in 1991. He was a faculty member in the Department of Statistics, The Ohio State University. Since 2003, he was with the School of Statistics, University of Minnesota. He is a Fellow of the American Statistical Association, and a Fellow of Institute of Mathematical Statistics.