Multimed Tools Appl (2016) 75:12919–12940
DOI 10.1007/s11042-015-2548-y

Latent semantic learning with time-series cross correlation analysis for video scene detection and classification

Shyi-Chyi Cheng · Jui-Yuan Su · Kuei-Fang Hsiao · Habib F. Rashvand

Received: 19 October 2014 / Revised: 17 January 2015 / Accepted: 5 March 2015 /
Published online: 19 March 2015

Abstract This paper presents a novel, latent semantic learning method based on the proposed time-series cross correlation analysis for extracting a discriminative dynamic scene model to address the recognition problems of video event recognition and 3D human body gesture. Typical dynamic texture analysis poses the problems of modeling, learning, recognizing and synthesizing the images of dynamic scenes based on the autoregressive moving average (ARMA) model. Instead of applying the ARMA approach to capture the temporal structure of video sequences, this algorithm uses the learned dynamic scene model to semantically transform video sequences into multiple scenes with a lower computational effort. Therefore, to generate a discriminative dynamic scene model with space-time information preserved is crucial for the success of the proposed latent semantic learning. To achieve the goal, the k-medoids clustering with appearance distance metrics first used to partition all frames of training video sequences, regardless of their scene types, to provide an initial key-frame codebook. To discover the temporal structure of the dynamic scene model, we develop a time-series cross correlation analysis (TSCCA) to the latent semantic learning, with an alternating dynamic programing (ADP) to embed the time relationship between the training images into the dynamic scene model. We also tackle the problem of dynamic programming, which is supposed to produce large temporal misalignment for periodic activities. Moreover, the discriminative power of the model is estimated by a deterministic projection-based learning algorithm. Finally, based on the learned dynamic scene model, this paper uses a support vector

S.-C. Cheng · J.-Y. Su
Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung City, Taiwan

S.-C. Cheng
e-mail: csc@mail.ntou.edu.tw

J.-Y. Su
e-mail: rysu@mail.mcu.edu.tw

K.-F. Hsiao (✉)
Department of Information Management, Ming-Chuan University, Taipei, Taiwan
e-mail: kfhsiao@mail.mcu.edu.tw

H. F. Rashvand
Advanced Communication Systems, University of Warwick, Coventry, UK
e-mail: H.Rashvand@ieee.org

machine (SVM) with a two-channel string kernel for video scene classification. Two test datasets, one for video event classification and the other for 3D human body gesture recognition, are used to verify the effectiveness of the proposed approach. Experimental results demonstrate that the proposed algorithm obtains good performance in terms of classification accuracy.

Keywords Time-series cross correlation analysis · Dynamic scene model · K-medoids clustering · SVM · Dynamic programming

1 Introduction

To semantically annotate a video sequence is a fundamental process in computer vision and machine learning due to its potential for semantic video database indexing and retrieval, intelligent video surveillance, and advanced man–machine interfaces [3, 9, 29]. Early work in video classification is focused on recognizing the scene and objects shown in a representative key-frame of a video shot, thus the temporal information of video segments are lost. This limits their applications on un-edited videos that do not contain obvious shots [42]. To solve the problem, various approaches typically employ machine learning tools to learn statistical spatial and temporal causality among video frames [2, 7, 26, 34, 37, 47, 48].

Image classification benefits video classification when we consider the problems of spatially and temporally categorizing multiple video sequences of dynamic scenes [17, 39]. For the purpose of image and video classification, many authors had incorporated a variety of features into their approaches [9]. These features were drawn from three modalities: text, audio, and visual. The bag-of-words (BoW) approach [34] is a simple but effective scheme to represent a video frame as a BoW histogram, though it lacks spatial and temporal statistical information in image modeling. Remarkable advances in acquiring technology equipped with shape modeling and rendering tools have facilitated multimedia applications of designing and manipulating complete 3D models of real-world objects. The analysis of digital 3D models is a challenging research problem in computer vision due to its potential applications in areas ranging from Human Computer Interface (HCI) [32, 35] to cognitive psychology of robotics [23]. The use of 3D visual features open up new possibilities of dealing with video classification but also presents some unique challenges. The evolution of low cost depth sensors, such as Microsoft Kinect and its real-time 3D Motion capture (MoCap) system [32], has enabled the enhancement of the user's experience with games, serious games, and presentation software [38].

Regardless of the types of features used, the common approach to video classification is the use of machine learning techniques to train classifiers for annotation video sequences. The research works for video classification in the existing literature solve two different problems. The first refers to annotating an entire video with a single classifier, while the other refers to classifying segments of a video, such as identifying abnormality activities in surveillance videos [13, 33]. The task of classifying an entire video attempts to classify a video into one of the several broad categories, such as the event type involved in a news video. However, when the scenes contain multiple segments of different movie genres, this process become more complicated. This is because classical constraints such as the use of a single classifier to annotate the entire video are no longer valid. To tackle the difficulty, one can apply concept detectors [31, 45] to divide a complex video into multiple semantic segments and annotate these segments with a multi-class classifier. However, the high variation in video frames seriously degrades the accuracy of existing concept detectors.

Many previous researches regard video classification, or more general video event recognition as classification over key-frame sequences by considering different feature sets to train Hidden Markov Models (HMM) or similar algorithms [36, 38, 47]. Video events are spatial-temporal patterns. Two issues are crucial to the success of video classification: the representation of suitable spatial-temporal features and the modeling of dynamic patterns. In video classification, it is a common practice to locate spatial-temporal interest points like STIP [27], and use HOF [28] or HOG [19] to represent local spatial-temporal patterns. Motivated from dynamic texture research, we can model the video sequences from each category as an autoregressive moving average (ARMA) model [12, 18, 39]. However, in many cases, certain constraints could be specified on the parameters of the ARMA model, which may result in nonlinear topological space in video classification. Although the complex and non-linear dynamics can be characterized by a Convolutional Neural Network [22], it is generally difficult to learn these models from the limited amount of training data.

Another approach to modeling video sequences is dynamic temporal warping (DTW), which aligns frames in a video to a class-specific model template [5, 17], and the detected video scene (segment) or object can be used to annotate the video. A template model might be a key-frame in video event classification or a key pose in human action recognition. The task of the DTW-based video classification involves recovering not only the spatial alignment, as in the case of image classification, but also the temporal alignment. For periodic actions such as “waving”, DTW is likely to produce large temporal misalignment which may ruin the classification accuracy [30, 43]. This poses additional challenges compared to image classification. Moreover, the discriminative power of the model template also affects the performance of the DTW-based video classification schemes. If the input video were temporally aligned with the model template, the video classification problem could be reduced to an image classification problem. Thus, the main challenge is to decide which pair of frame-models should be aligned. To deal with this issue, the DTW-based approaches optimally aligned video frames and template models to choose the transformation that minimizes the sum-of-errors over all the frames. Thus, the DTW-based approaches are computationally expensive in general. Caspi et al. considered a video sequence as a space-time volume and volume-to-volume matching techniques can be applied using either the entire volume or a collection of sub-volumes, or point trajectories that address the problem of spatial-temporal alignment [10, 11]. However, these methods involve an optimal search for both the spatial and temporal registration parameters over the entire video, which results in high computational complexity again.

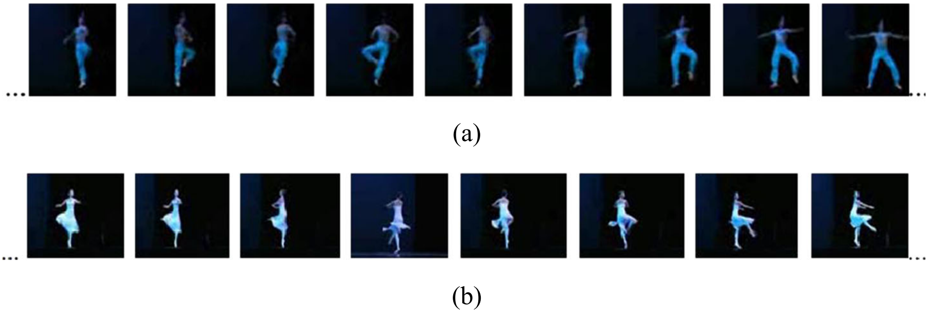
In this paper, we propose a DTW-based video detection and classification based on the time-series cross correlation analysis with a novel latent semantic learning to generate the discriminative dynamic scene model for recognizing the event exhibited by a video sequence. The whole video classification framework consists of three steps: 1) partitioning a video sequence into multiple dynamic scenes with the learned dynamic scene model; 2) encoding the detected video scenes with sets of moving key-frames; 3) using a SVM classifier with a two-channel string kernel to annotate each video scene. In learning time, to ensure the detected dynamic scenes preserve the class semantics and temporal structures in video sequences, the use of a time-series correlation analysis together with a quality measurement with dynamic programming allows the optimal alignment between the dynamic scene model and the input video sequences. The alignment results can then be used to generate the final dynamic scene model with the help of the latent semantic learning.

We have validated our proposed approach with extensive tests carried out on well-known benchmark datasets as well as with comparisons to related methods. The obtained results have clearly shown the success and effectiveness of our solution, even in the presence of noise or periodic activities in the datasets. The contributions of the proposed video classification framework include: 1) the latent semantic learning enhances the discriminative power of the dynamic scene model; 2) both spatial and temporal structures in training video sequences are preserved by the dynamic scene model; 3) class information is used in the latent semantic learning in order to preserve scene semantics; 4) the video scene detection is simple and fast, which tackle the difficulties of the DTW-based video classification. The rest of this paper is organized as follows. In Section 2, the approach related to the DTW-based video classification is described. Section 3 presents the learning of the dynamic scene model with the help of the latent semantic learning. Section 4 shows the implementation of the framework and applications. Section 5 presents the experimental results followed by the conclusion to summarize the main contributions and suggest possible future work in Section 6.

2 DTW-based video scene detection and classification

A video scene class such as dancing can be represented by a sequence of moving snippets (shots) which refer to activities observed in 3–5 frames. To represent every snippet with a key-frame, the event in the video sequence can be represented with a compact set of moving key-frames, shown in Fig. 1. As mentioned above, we can compute the distance between two video scenes by temporally aligning their key-frames using the DTW algorithm. Obviously, the accuracy of DTW-based video comparison degrades significantly when it generates large temporal misalignment. To tackle the difficulty, for each video scene class, we use the learned class-specific template sequence to optimally locate the temporal boundaries of the target video scenes in a test video sequence using the proposed DTW-based video alignment algorithm.

As shown in Fig. 2, this approach extracts multiple video scenes in a video even though some of them might be false positives, which can be further eliminated with the SVM classifier. Thus, the proposed approach can avoid generating large temporal misalignment even when the test video sequence contains periodic or multiple activities. However, the approach has two potential problems to deal with: 1) the performance degrades dramatically when the types of starting key-frames of the target video scenes and the corresponding class template are different; 2) the computational


(a)

(b)

Fig. 1 Representing video scenes as sets of moving key-frames: (**a**) partial frames of a "dancing" video; (**b**) partial frames of the other "dancing" video

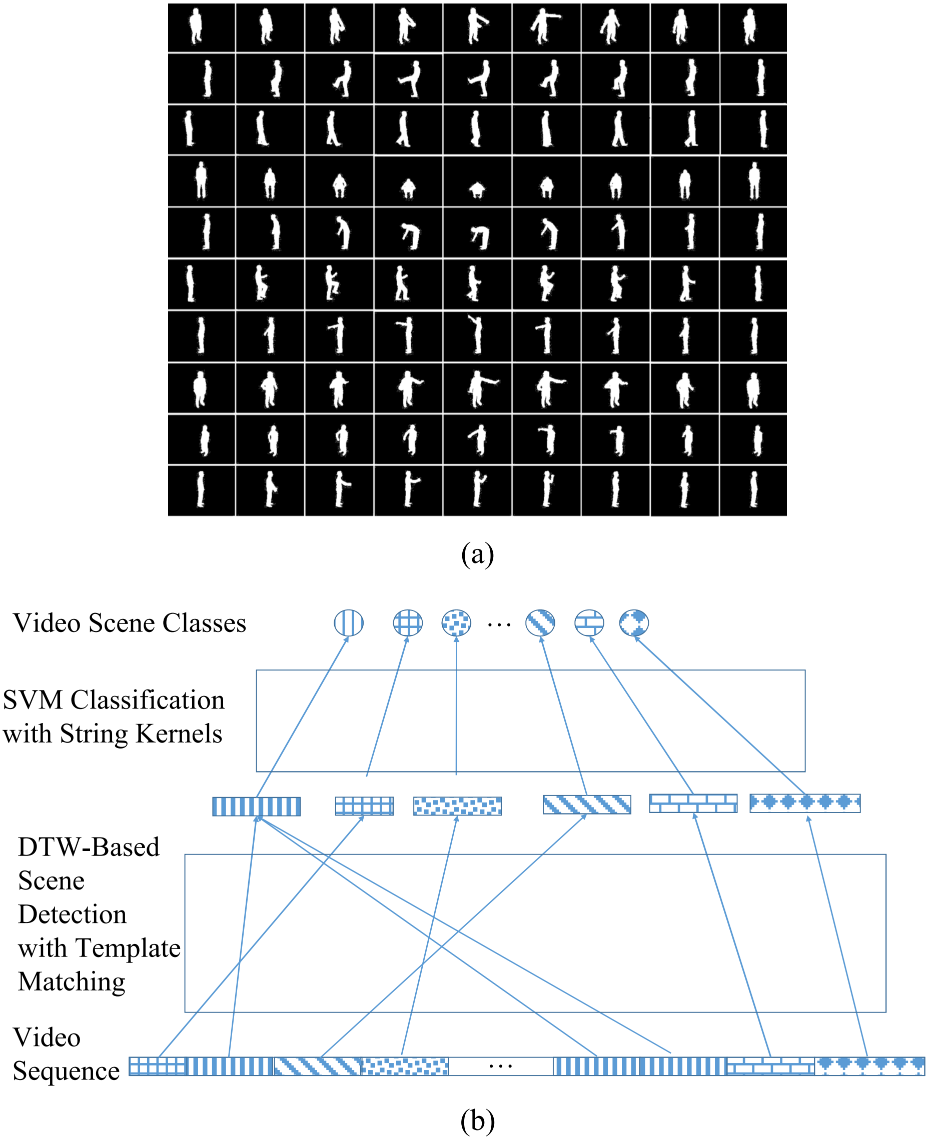

(a)
Video Scene Classes
···
SVM Classification with String Kernels
DTW-Based Scene Detection with Template Matching
Video Sequence
···
(b)

Fig. 2 The DTW-based video scene detection with template matching for classification: (**a**) the class-specific templates of dynamic key-frames of the 3D gesture recognition dataset [1]: *rows* from *top* to *bottom* are "Opening," "Kicking" "Walking," "Crouching down," "Picking up," "Marching," "Pointing at," "Pushing," "Moving apart," and "Pulling;" (**b**) the steps to extract video scenes in a video sequence for classification

complexity to compute the DTW distances is high. These two issues should be seriously considered in the approach in order to annotate a video sequence in a low computational effort.

Let V=$\{F_1,F_2,\ldots,F_n\}$ be a video sequence consisting of n frames, assuming V contains multiple video scenes of different video classes. Let $V^{'} = \left\{F_t, F_{t+1}, \ldots, F_{n'}\right\}$, $n^{'}>s$, be a video scene in V, starting at time t and ending at time $n^{'}$. To keep the notations simple and without loss of generality, we assume that $t=1$. A video frame is visually represented as BoW histogram [34]. To quantify the similarity between two video frames, F_1 and F_2, the well-known Bhattacharyya distance [6] is used to compare two video frames in terms of their BoW histograms H_1 and H_2:

$$D_F(F_1,F_2)=\sqrt{1-\sum_i \frac{\sqrt{h_{F_1}(i)\times h_{F_2}(i)}}{\sqrt{\sum_j h_{F_1}(j)\times \sum_j h_{F_2}(j)}}} \tag{1}$$

Given a class template $T=\{K_i\}_{i=1}^{m}$ of m video frames, the DTW algorithm defines a warping path of length l between T and a candidate video scene $V^{'}$ in V to compute the temporal distortions using dynamic programming. To align T with $V^{'}$, a cost matrix M is designed to define a mapping between T and $V^{'}$: $\boldsymbol{P}=\{p_1,p_2,\ldots,p_l\}$, where p_i indexes the cost of an alignment. We assume the boundary conditions of P to be $p_1 = (1, 1)$ and $p_\mathrm{l} = (m, n^{'})$ that defines the warping path of minimal cost:

$$DTW\left(\boldsymbol{T},\boldsymbol{V}^{'}\right)=\arg\min_{l}\left\{\frac{1}{l}\sqrt{\sum_{t=1}^{n^{'}}D_F(p_t)}\right\} \tag{2}$$

where $D_F\ (p_t)$ of $p_\mathrm{t}=(i,\ j)$ is the cost to align between $K_i$in T and F_j in $V^{'}$. Dynamic programming uses the following recurrent equation:

$$M(i,j)=D_F\left(K_i,F_j\right)+\min(M(i–1,j–1),M(i,j–1)+M(i–1,j)) \tag{3}$$

to compute the cost of M$(i,\ j)$ which represents the minimal cost to align between the sub-sequences $\{K_1,K_2,\ldots,K_i\}$ and $\{F_1,F_2,\ldots,F_j\}$. In this case, we have

$$DTW\left(\boldsymbol{T},\boldsymbol{V}^{'}\right)=\frac{1}{l}M\left(m,n^{'}\right) \tag{4}$$

Given the streaming nature of our problem, the input video sequence V has no definite length and may contain several occurrences of the class-specific video scene $V^{'}$. Given the class template T, the worst case of the computational complexity to detect all occurrences in V with the above DTW algorithm would be $O(n^2\times m\times n')$. To speed up the algorithm, we propose a histogram-based video scene detection, of encoding every candidate video scene as a key-frame-based integral histogram using a learned frame codebook C. Note that because of noise perturbations and speed variation of movements, two different video classes may contain some identical key-frames and instances of the same class may be slightly different. To overcome this problem, all training video frames regardless of their classes are grouped, based on similarity criteria of their visual features (cf. (1)), into a set of k clusters. In practice, we have achieved this using the K-*medoids* algorithm [5], which uses the medians of these clusters as the representative key-frames to construct the video frame codebook $\mathrm{C}=\left\{\widetilde{K}_i\right\}_{i=1}^{k}$.

A video-encoding process is proposed to transform T into a key-frame-based integral histogram using Eq. (1) and C. To achieve the goal, for each video frame of T, we first search its nearest neighbor in C in order to construct an indexing set, denoted by $\widetilde{\mathrm{T}}$, that presents the sequence of time-ordered representative key-frames to describe the temporal structure of T:

$$\widetilde{\mathrm{T}} = \left\{ r(K_i) \middle| r(K_i) \ = \ \arg\min_{j} D_F\left(K_i, \widetilde{K}_j\right), \ j = 1, \dots, m \right\} \tag{5}$$

Where $r(K_i)$ is the cluster number to indicate that the nearest neighbor of the video frame $\mathrm{K_i}$ in C. Considering the consecutive frames of the same cluster index to be a shot, $\widetilde{\mathrm{T}}$ segments T into a sequence of time-ordered shots with boundary points $\boldsymbol{B}=\{\tau_i\}_{i=1}^{s+1}$, where s is the number of shots, $\tau_1=1$, and $\tau_{s+1}=m+1$. Let $\Delta \boldsymbol{T}$ be the length of a class-specific template T. Using B, we define the key-frame-based histogram W of T, which is a histogram given by

$$W^{\mathrm{T}} = \left\{ w_i \middle| w_i = \frac{\tau_{i+1} - \tau_i}{\Delta \boldsymbol{T}}, i = 1, \dots, s \right\} \tag{6}$$

Finally, the temporal structure of T is represented as a triple: $\overline{\boldsymbol{T}} = \left\{ W^{\mathrm{T}}, \overline{K}^{\mathrm{T}}, I^{\mathrm{T}} \right\}$, where $\overline{K}^{T} = \left\{ \overline{K}_j \right\}_{j=1}^{s}$ is the key-frame sequence consisting of the shot key-frames of T and $I^{T}=\{I_j\}_{j=1}^{s}$ I_j is the sequence of shot cluster indexes of T.

To ensure the alignment accuracy between T and a test video V with dynamic programming, two difficulties issue from V of starting activities from a key-frame that is different from the starting key-frame of T or having repeated activities, to ensure both T and V are starting from similar key-frames. We cyclically permute the frames of V until the starting frame of V is similar with that of T. Thus, given the starting key-frame $\overline{K}_1$ of T, the corresponding starting frame of V is defined by

$$F_s = \arg\min_{F_i \in \mathrm{V}} D_F\left(\overline{K}_1, F_i\right) \tag{7}$$

Next, the permuted video sequence V is encoded as a triple $\overline{\boldsymbol{V}} = \left\{ W^{\mathrm{V}}, \overline{K}^{\mathrm{V}}, I^{\mathrm{V}} \right\}$. The ending shot v^{*} in the permuted V to construct the candidate video sequence $\mathrm{V}^{'}$ is then given by:

$$\begin{aligned} v^{*} &= \arg\min_{u} DTW\left(W^{\mathrm{T}}, W^{\mathrm{V}}(u)\right), \\ e &= n \times \sum\nolimits_{i=1\dots,v^{*}} w_i^{\mathrm{V}} \end{aligned} \tag{8}$$

where $W^{\mathrm{V}}(u)$ is the key-frame-based histogram of the video sequence consisted of the first u shots of the permuted video sequence V and e is the ending position to construct the candidate video scene $\mathrm{V}^{'}$, consisted of frames with time indexes from 1 to e in the permuted video sequence V.

The class-specific video scene detection is simple and thus fast. For instance, if the cluster index list of a video sequence V is V={8,8,8,8,1,1,1,1,2,2,2,2,5,5,5,5,6,6,6}. V is separated into 5 shots with four snippet intervals: [1, 4], [5, 8], [9, 12], [13, 16], and [17, 19]. That is, V is encoded by 5 triples: $\overline{\mathrm{V}} = \left\{ \left\{ \frac{4}{19}, F_2, 8 \right\}, \left\{ \frac{4}{19}, F_6, 1 \right\}, \left\{ \frac{4}{19}, F_{11}, 2 \right\}, \left\{ \frac{4}{19}, F_{14}, 5 \right\}, \left\{ \frac{3}{19}, F_{18}, 6 \right\} \right\}$ using C. Assume that the given class template T defines an activity with the starting frame of cluster index 2. In this case, the video V obviously cannot be properly normalized because it's starting frame is of a different cluster index, i.e., 8. However, the frames in the third shot of V are obviously more similar with the starting frame of T. To tackle this difficulty, we circularly shift every triple in $\overline{\mathrm{V}}$ to obtain a new triple sequence: $\overline{\mathrm{V}} = \left\{\left\{\frac{4}{19}, F_{11}, 2\right\}, \left\{\frac{4}{19}, F_{14}, 5\right\}, \left\{\frac{3}{19}, F_{18}, 6\right\}, \left\{\frac{4}{19}, F_{2}, 8\right\}, \left\{\frac{4}{19}, F_{6}, 1\right\}\right\}$.

Recently, Ballan et al. have successfully used the kernel string approach in video event classification, which accounts for the temporal progression of the activities [3]. Let $\mathrm{V}_1^{'}$ and $\mathrm{V}_2^{'}$ be two detected candidate video scenes. The kernel used in the support vector machine (SVM) to classify input video scene is a Gaussian function:

$$G\left(\mathrm{V}_1^{'}, \mathrm{V}_2^{'}\right) = \exp\left(\text{-}\gamma DP\left(\mathrm{V}_1^{'}, \mathrm{V}_2^{'}\right)\right) \tag{9}$$

where γ is the regulation factor and $DP(\mathrm{V}_1^{'}, \mathrm{V}_2^{'})$ is the distance between $\mathrm{V}_1^{'}$ and $\mathrm{V}_2^{'}$, which is given by

$$DP\left(\mathrm{V}_1^{'}, \mathrm{V}_2^{'}\right) = aDTW\left(W^{\mathrm{V}_1^{'}}, W^{\mathrm{V}_2^{'}}\right) + bDTW\left(\overline{K}^{\mathrm{V}_1^{'}}, \overline{K}^{\mathrm{V}_2^{'}}\right),\ a + b = 1 \tag{10}$$

where a and b are the weightings of the alignment cost between $\mathrm{V}_1^{'}$ and $\mathrm{V}_2^{'}$ using the key-frame-based histogram and the visual features of key-frames, respectively. In practice, we set both a and b to be 0.5. Figure 3 shows the summarized block flowchart of the DTW-based video classification. Obviously, the dynamic scene model consisted of the key-frame codebook C, the class templates Ts, and the SVM classifier, is the core of the video classification framework.

3 Latent semantic learning with time-series cross correlation analysis

We consider class templates as the latent variables, which preserve spatial and temporal structures of video classes. Thus, the latent semantic learning to provide an effective dynamic scene model is the key to success for the DTW-based video classification. The learning procedure is composed of two alternating steps: given current class templates and training video sequences, the time-series cross correlation analysis (TSCCA) is first used to capture the temporal structures of video classes; what follows is the updating of the dynamic scene model.

Figure 4 shows the direct implementation of learning the dynamic scene model (DSM) with the time-series cross correlation analysis (TSCCA), which consists of four steps, video encoding, DTW-based video scene detection, video scene clustering, and dynamic scene model updating. The first steps of the TSCCA are the same as those described in the previous section. Thus, the details of these two steps are skipped in this section.

First of all, the frame cookbook C and class templates $\{\boldsymbol{T}_i^{(0)}\}_{i=1}^{c}$ are given to the TSCCA to learn a discriminative DSM for video classification. As mentioned above, C is a codebook of k

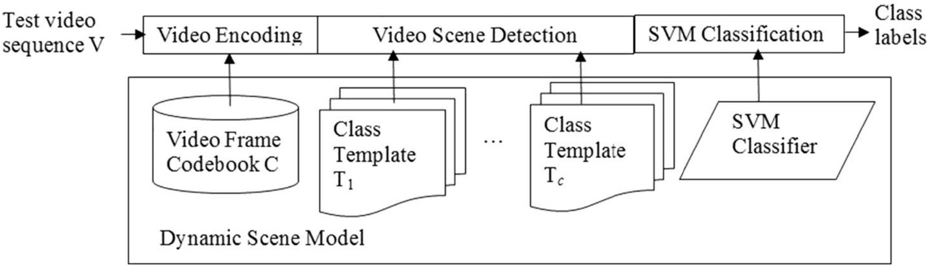


Fig. 3 The block flowchart of the DTW-based video classification

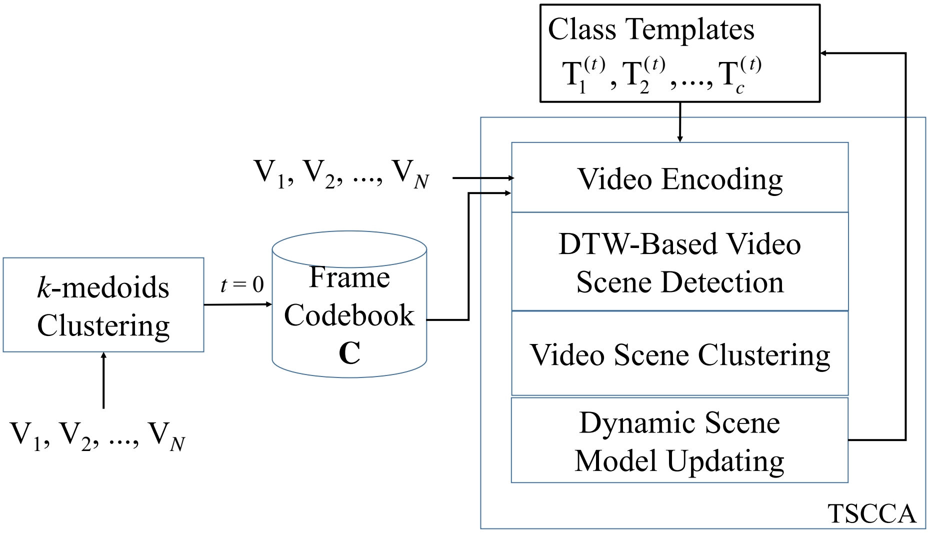


Fig. 4 The approach to learning the dynamic scene model with the time-series correlation analysis: $V_1, \ldots, V_N$ are the training sequences

frame clusters with each of them represented by a key-frame $\widetilde{K}_i$, i=1,…,k performing the *K-medoids* algorithm on all training video frames. Obviously, C preserves the spatial similarity between two video frames without guaranteeing their temporal relationship. To tackle the difficulty, we introduce the class templates to preserve temporal structures in video sequences. To simplify the learning, for each video class, we select a training video of a single activity in the class as the initial class template. The TSCCA is then used to modify the class templates $\{\boldsymbol{T}_i^{(t)}\}_{i=1}^{c}$ until the convergence of these class templates.

Let V=$\{F_1, F_2, \ldots, F_n\}$ be a video sequence consisting of n frames, where each frame is visually represented as a BoW histogram [34]. The learning of dynamic-scene model aims at discovering the linear dependencies between two different but related views of the same underlying semantics [8]. It is used to detect the DSM with maximum cross correlation between linear combinations of frames in class templates T_s and sequences in a training video dataset. In the TSCCA, the objective is to respectively represent the candidate video scene V' in V and T as two weighted BoW histograms $\overline{H}^{V'} = \left\{\left(w_i^{V'}, H_i^{V'}\right)\right\}_{i=1}^{n'}$ and $\overline{H}^{T} = \left\{\left(w_i^{T}, H_i^{T}\right)\right\}_{i=1}^{m}$, where $H_i^{V'}$ $\left(H_i^{T}\right)$ is the BoW histogram to visually represent the i-th key-frame $\overline{K}_i^{V'}$ $\left(\overline{K}_i^{T}\right)$ of V' (T), such that the correlation (similarity) between V' and T is mutually maximized. In other words, the aim is to maximize the correlation between the time-series sequences $\overline{H}^{V'}$ and $\overline{H}^{T}$:

$$\rho = \max_{\overline{H}^{T}, \overline{H}^{V'}} G\left(\overline{H}^{T}, \overline{H}^{V'}\right) \tag{11}$$

where $G\left(\overline{H}^{T}, \overline{H}^{V'}\right)$ is the Gaussian function output, shown in (9), to obtain the maximal correlation between $\overline{H}^{V'}$ and $\overline{H}^{T}$. Note that the Gaussian function is used to the SVM classifier for further video classification. Thus, using the class template T, the underlying concept of the optimization function defined in (9) is to detect the video scene $\mathrm{V}^{'}$ in a test video V with the maximal probability to be classified as the class of T.

To solve the optimization problem of TSCCA defined in (11), the approach starts by obtaining an initial value for the sequence of weighted BoW histograms of a class template T, i.e., $\overline{H}^{\mathrm{T}}$ with the video encoding process, which encodes T as a triple $\overline{\boldsymbol{T}}=\left\{W^{\mathrm{T}},\overline{K}^{\mathrm{T}},I^{\mathrm{T}}\right\}$ using the current frame codebook C. Now, the maximization problem can be written as:

$$\overline{H}^{\mathrm{V}'}=\arg\max_{\mathrm{V}'\in\mathrm{V}} G\left(\overline{H}^{\mathrm{T}},\overline{H}^{V'}\right) \tag{12}$$

However, the optimization goal is not easy to achieve using (12). Instead of using (12) directly, to reduce the optimization effort, for each video class, the proposed DTW-based video scene detection is used to obtain the sub-optimal candidate video scene $\mathrm{V}^{'}$.

An obvious deficiency in the TSCCA is the learned initial class templates which might be of poor discriminative power in video scene classification since the above time-series cross correlation analysis maximizes the correlation between two related views in the same class. The discriminative power of the class templates decreases to ignore the space-time frame variants of training sequences. To tackle this problem, the semantic latent learning should be extended to update the class templates with a video scene clustering. Without loss of generality, we assume every training video sequence V is annotated by a single class label. Let $\left\{\overline{H}_i^{\mathrm{V}}\right\}_{i=1}^{c}$ be the set of sequences of weighted BoW histograms to represent the detected video scenes of V using different class templates T_i, $i=1, \ldots, c$. Using (9), we can compute the alignment result between the class templates and these candidate video scenes with DTW, in terms of weighted BoW histograms, the video V can be classified as

$$i=\arg\max_{j} G\left(\overline{H}_j^{\mathrm{T}},\overline{H}_j^{\mathrm{V}}\right) \tag{13}$$

Obviously, the discriminative power of T_j is not good enough when $class(\boldsymbol{T}_i)\neq \boldsymbol{V}\notin class(\boldsymbol{V})$, where class(.) returns the corresponding class label. To enhance the discriminative power of class templates, we first perform the video scene clustering using (13) to partition all training video sequences into c clusters, in which all miss-classified video sequences are filtered, shown in Fig. 5. For each video class *i*, we collect all miss-classified detected video scenes labeled as class *i* to construct a new video scene cluster. Accordingly, for each video scene cluster VC, we select the median as the new class template:

$$\boldsymbol{T}^{(t+1)}=\arg\min_{\mathrm{V}_i'\in\mathrm{VC}}\sum_{\mathrm{V}_j'\in\mathrm{VC}} DP\left(\boldsymbol{V}_i^{'},\boldsymbol{V}_j^{'}\right) \tag{14}$$

where DP(.,.) is the distance with dynamic programing defined in (10). In practice, the final number of class templates used for each video scene class is 2, which can be extended to use more class templates on those misclassified patterns if the frame variations in a class are large.

The TSCCA starts a new loop to update the current class templates until all class templates are un-changed again. The following algorithm summarizes the latent semantic learning with TSCCA.

3.1 Algorithm TSCCA

Input: a set of training video sequences V_i annotated by class(V_i), $i=1,\ldots, N$; a frame codebook C.

Output: a set of class templates T_j, $j=1, \ldots, 2^*\mathrm{c}$.

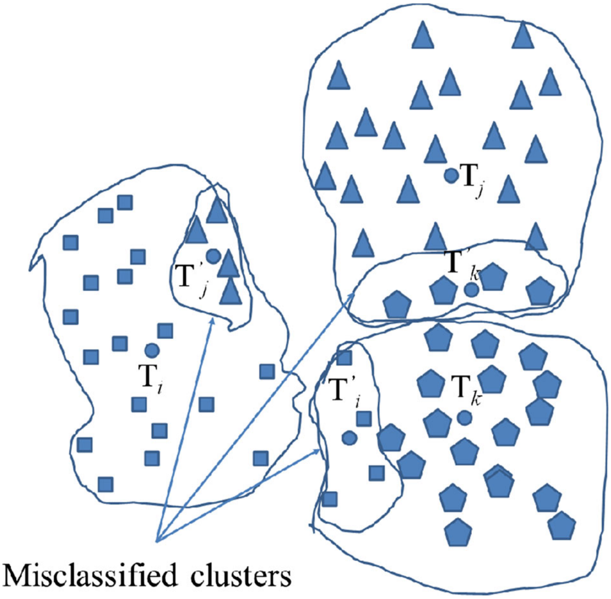


Fig. 5 The class template updating with the video scene clustering, which searches the misclassified video scene clusters and create new class templates to represent them

```
Method:
Step 1. Initialization: randomly select a training video of a single activity in each class as the initial class templates $\{\boldsymbol{T}_i^{(0)}\}_{i=1}^{c}$, which are encoded as the set of triples $\overline{\boldsymbol{T}}_i^{(0)} = \left\{ W^{\mathrm{T}_i^{(0)}}, \overline{K}^{\mathrm{T}_i^{(0)}}, I^{\mathrm{T}_i^{(0)}} \right\}, i = 1, \ldots, c$
Step 2. $t$=0; T$_{\text{current}}$ = $c$; for $i$ =1 to $c$ do $class(\boldsymbol{T}_i^{(0)})=i$;
Step 3. Repeat
    for $i$ =1 to $c$ do
        ${}_i= {}'_i=\{\}$;
    for $i$ =1 to $N$ do
        detect the set of candidate video scenes $\{\boldsymbol{V}'_{i.j}\}_{j=1}^{c}$ in V$_i$ using the DTW-based video scene detection;
        encode $\{\boldsymbol{V}'_{i.j}\}_{j=1}^{c}$ as the set of triples: $\overline{\boldsymbol{V}}'_{i,j} = \left\{ W^{\overline{\mathrm{V}}'_{i,j}}, \overline{K}^{\overline{\mathrm{V}}'_{i,j}}, I^{\overline{\mathrm{V}}'_{i,j}} \right\}, j = 1, \ldots, c$;
        compute the class label $j$ for V$_i$ using (13);
        if $class$(V$_i$) = $j$ then ${}_j= {}_j\cup\{i\}$;
        else $\Gamma'_{l_i} = \Gamma'_{l_i}\cup\{i\}$;
    for $i$ =1 to $c$ do
        $j$=2*$i$-1;
        compute new class template $\boldsymbol{T}_j^{(t+1)}$ using (14) and ${}_i$;
        $class(\boldsymbol{T}_j^{(t+1)})=class(\boldsymbol{T}_i^{(t)})$;
        $j$=$j$+1;
        compute new class template $\boldsymbol{T}_j^{(t+1)}$ using (14) and ${}'_i$;
        $class(\boldsymbol{T}_j^{(t+1)})=class(\boldsymbol{T}_i^{(t)})$;
until all class templates are stable.
```

4 Implementation and applications

To verify the effectiveness of the proposed approach, the recognition problems of video event recognition and 3D human body gesture are addressed. Although these 2 applications are different in video frame representation, the issues in annotating a video event and a 3D human body gesture are similar. Thus, to apply the approach to recognizing them, we need to select suitable frame representations for individual applications. To simplify the implementation, the frames in both applications are represented as the BoW histograms.

A common BoW approach to model a frame of video events is to extract features from a local appearance, i.e., a video patch, in all training video sequence of an event category. A codebook of local appearances is then built up using a clustering algorithm on the extracted features of video patches to learn the appearance variability of an event category. A local appearance codebook consists of multiple codewords, where each of them is determined by the mean features of a video patch cluster. Based on this codebook, we could compute a BoW histogram of a frame by mapping each video patch to a codeword. Thus, these histograms could then be used as feature vectors to characterize the activity of a video sequence.

To apply the BoW approach for modeling 3D shapes, the training shapes are transformed to a set of surfaces [16] and the descriptors of these surfaces are extracted. Note that these descriptors are invariant with respect to scaling, rotation and translation. The descriptors are then clustered into n clusters and the average surface descriptors of the clusters are adapted as the codewords of the local appearance codebook. The 3D shape is then represented as a BoW histogram of the codewords.

Although the frame representations of the 2 applications are different, they contain the same dynamic structures so that our process model is suitable for these different applications. To apply our model, initially all frames of the training video are clustered through a *K-medoids* algorithm, and the video frame codebook are constructed with the key frames of the clusters. After the TSCCA steps as shown in Fig. 4, the class-specific templates are obtained. In the scene detection phase, each frame of the test video is encoded as one of the key frames in the video frame codebook. The latter steps are shown in Fig. 3.

Although the frames could be represented with a more sophisticated form such as HOF [28], HOG [19], or BOR [21], in this paper, the frames in the applications above are represented with simple BoW histograms. Based on our framework, even frames represented with the simple BoW could achieve good result for video scene detection and classification.

5 Experimental results

The performance of the proposed approach is verified according to the two applications, the video event classification and 3D human body gesture recognition. For video event classification, the UCF sports action dataset [40] and a subset of TRECVID 2005 [1] are adapted to analyze the performance. The camera motion and background clutter cause the UCF dataset to be very challenging. Evaluations were done with a five-fold cross-validation. For TRECVID 2005, five classes, including Exiting Car, Running, Walking, Demonstration or Protest and Airplane Flying, are selected for testing. In this dataset, videos show high intra-class variability due to their variation in length. Experiments are performed using 3-fold cross-validation for this dataset.

In the first experiment, the comparison of Hough forest [46], Rodriguez et al.'s method [40], Wang et al.'s method [44], Yeffet & Wolf method [49], Kovashka & Grauman [25] and the SVM classifiers with the proposed string kernels on the UCF dataset are shown in Table 1.

The mean classification accuracy obtained by the proposed method (83 %) largely outperforms the compared methods except the method proposed by Kovashka & Grauman [25] which uses HoG3D [24], HoF and HoG [28] for feature extraction and integrates multiple kernel learning (MKL) [20, 41] for distance metrics selection. These processes generally extract more features from video than basic BoW and need more computation. We believe that using more features would also increase the classification accuracy for our method. Figure 6 shows the confusion matrix for the proposed SVM classifiers on the UCF dataset. The improvement in accuracy is due to the fact that the SVMs have a better performance on the four most critical actions: Diving, Riding, Swings, and Weight-lifting. The worst classification results are due to the fact that it has an extremely large variability in the part of the action.

The performance comparison using TRECVID 2005 in Fig. 7 shows the improvement obtained using the SVM classifiers, based on the proposed temporally normalized string kernels, with respect to the traditional string kernels [4] and Wang's method [34] which uses traditional BoW representations. For multiclass classification, the LIBSVM implementation [14] using the "one-against-all" approach is adapted in this experiment. Figure 7a reports the global accuracy (58 %) and the confusion matrix for the proposed method. The improvement in accuracy is due to the fact that the proposed method has a better performance on the most critical events. The extremely large variability containing multiple persons and multiple backgrounds in the part of the Demonstration or Protest event causes the worst classification results.

MAP is the mean of average precision scores over a set of queries. Figure 7b reports the comparison results of MAP between a traditional BoW approach [34], the SVM classifiers in [4], and the proposed method. Our method outperforms the traditional bag-of-words approach in all classes, with 37.5 % improvement of the MAP. Based on the dataset TRECVID 2005, the mean accuracy obtained by the proposed method (0.44) largely outperforms that obtained using the Wang's method (0.32) and traditional SVMs with string kernels (0.35) [4]. The other result and discussion can be seen in [15].

For 3D human body gesture recognition, the performance of the proposed approach is verified by a public 3D gesture recognition dataset [1], composed of 10 different actions performed by 5 different actors (4 men and 1 woman), each of them performs the same action 5 times [1, 36]. The sequences are classified into ten action classes: "Opening," "Kicking," "Walking," "Crouching down," Picking up," "Marching," "Pointing at," "Pushing," "Moving apart," and "Pulling." All of them start and end at a neutral position. In a room, eight cameras are simultaneously used to acquire the 3D gestures for each actor. These cameras extract eight silhouette images coming from different views of an actor. These images are then integrated to form a 3D gesture using the volume intersection method. Each actor is requested to perform the same gesture in slightly different ways starting from a different point in the room and with 15 an orientation different from the previous one. The total number of 3D data sequences is 250. Figure 2a shows the set of ten performed actions with 9 templates that represent its time evolution.

Table 1 Results per video sequence (in % of accuracy) on the UCF sports dataset

Method	Average performance
Proposed	**83.0**
Hough forest [46]	79.0
Rodriguez et al. [40]	69.2
Wang et al. [44]	81.3
Yeffet & Wolf [49]	79.2
Kovashka & Grauman [25]	87.27

	Diving	Golf	Kick	Lift	Riding	Run	Skate	Swing 1	Swing 2	walk
Diving	**1**	0	0	0	0	0	0	0	0	0
Golf	0.1	**0.7**	0	0	0	0.1	0	0	0	0.1
Kick	0	0	**0.87**	0	0	0.05	0	0	0.03	0.05
Lift	0	0	0	**0.65**	0	0	0	0.2	0	0.15
Riding	0	0	0	0	**0.9**	0	0.1	0	0	0
Run	0	0	0.03	0	0	**0.73**	0.08	0	0	0.12
Skate	0	0	0	0	0	0.05	**0.85**	0	0	0.1
Swing 1	0	0	0	0.02	0	0	0	**0.95**	0.03	0
Swing 2	0	0	0	0.1	0	0	0	0	**0.9**	0
walk	0	0	0.02	0	0	0.1	0.13	0	0	**0.75**

Fig. 6 Confusion matrix of the proposed SVM classifier with string kernels for the UCF sports dataset

First of all, we present some experimental results to illustrate the effectiveness of the proposed surface descriptor using the multiple principal planes analysis. Figure 8 shows the results of 3D shape approximation using the multiple principal planes analysis. Obviously, to decrease the average shape error, we need a large amount of principal planes to approximate a 3D shape. However, this results in a large amount of surface descriptors to be extracted for training the surface descriptor codebook. To be a trade-off between the representation quality and learning speed, we use 700 principal planes to approximate a 3D shape since in this case the curve in Fig. 8b is almost flat.

As mentioned above, we use a simple k-means clustering algorithm to obtain the surface descriptor codebook and the video frame (pose) codebook C. The video frame codebook C clustered through *K-medoids* algorithm can be directly used to detect the templates in the input 3D shape sequences for SVM training and classification.

Table 2 shows the relationship between these two parameters, ns and nt, and the classification rate of the gesture recognition when we use the learned time-ordered video frame codebook C to detect the dynamic poses of input gesture sequences. The average classification rate can achieve 98 % when we set the sizes of the surface descriptor codebook and the initial pose codebook to be 100 and 50, respectively. Figure 9 shows the partial templates in the learned time-ordered pose codebook. Similar templates coming from different instances and gesture classes are correctly grouped into a cluster. This verifies the effectiveness of the proposed surface descriptor which is robust to shape rotation. On the other hand, the shapes of different poses but with the same time index are grouped into the same hyperedge.

We use the leave-one-out principle to train and test the gesture classifier. Figure 10 shows the confusion matrix of the proposed classifier. To compare the performance comparison between the proposed approach and Pierobon's method [36], three types A, B, and C of classification errors are defined. Type A error is defined as misclassification of an enrolled action performed by the unknown (skipped during training) actor; type B error is misclassification of the 'unknown' gesture, i.e. a sequence of the 'unknown' class, is classified as one

	Existing Car	**Running**	**Walking**	**Demonstration or Protest**	**Airplane Flying**
Existing Car	**0.54**	0.15	0.03	0.15	0.13
Running	0.04	**0.61**	0.12	0.20	0.03
Walking	0.00	0.02	**0.64**	0.03	0.13
Demonstration or Protest	0.15	0.20	0.11	**0.49**	0.05
Airplane Flying	0.10	0.10	0.13	0.05	**0.62**

(a)

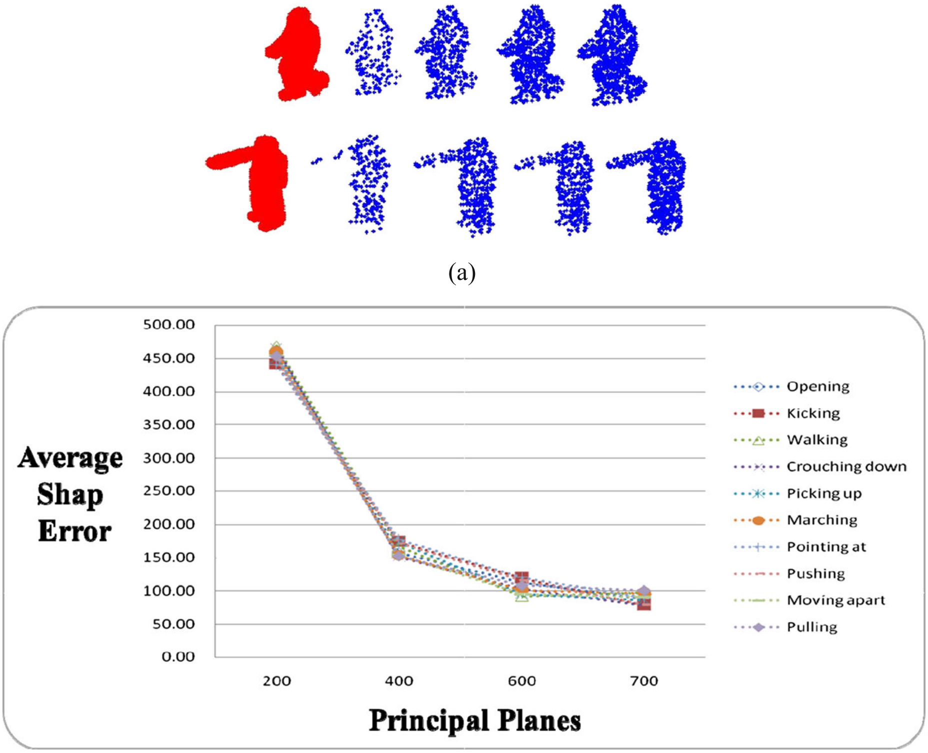
Classes	*BoW* [4]	*SVM* [47]	*Proposed*
Existing Car	0.25	0.37	**0.43**
Running	0.57	0.36	**0.49**
Walking	0.28	0.29	**0.38**
Demonstration or Protest	0.32	0.38	**0.47**
Airplane Flying	0.17	0.34	**0.43**
MAP	0.32	0.35	**0.44**

(b)

Fig. 7 Performance comparison using TRECVID 2005: (**a**) Confusion matrix of the proposed SVM string classifier; (**b**) mean average precision (MAP). MAPs for traditional BoW approach equal to 0.32, 0.35 for traditional SVM with string kernel, and 0.63 for the proposed method

known classes (false positive error); type C error is misclassification of an enrolled action performed by the unknown (skipped during training) actor as an ‘unknown’ gesture (false negative error). Table 3 shows the performance comparison in terms of these three error types. Accordingly, the proposed approach outperforms Pierobon’s method.

In the proposed method, the alignment of input video and the class templates is performed through temporally aligning their key-frames using the DTW algorithm. It will need a lot of computing when the alignment is directly based on the video content since the time complexity of DTW is O(n^2). In our method, a video sequence V is encoded by a sequence of triples and



(a)

(b)

Fig. 8 Examples of shape approximation using the proposed multiple principal planes analysis. *Rows* in (**a**) are leaded by the original 3D shape and followed by the approximated shapes with 200, 400, 600, and 700 principal planes. The *diamond markers* indicate the centers of the detected principal planes. The figure in (**b**) depicts the average shape errors using different numbers of principal planes to approximate 3D data sequences in all classes

the DTW is performed on the encoded representations so that the computing needs is greatly reduced.

6 Conclusion

In this paper, a DTW-based video detection and classification model based on the time-series cross correlation analysis (TSCCA) is proposed. The TSCCA retains time space correlation. This model can be applied to many applications with similar dynamic structures. The latent semantic learning is used to generate the discriminative dynamic scene model for recognizing the event exhibited by a video sequence. There are 3 steps in the video classification framework: 1) using the learned dynamic scene to partition an input video sequence into multiple dynamic scenes; 2) using sets of moving key-frames to encode the detected video scenes; 3) using a SVM classifier with a two-channel string kernel to annotate each video scene. In learning time, to ensure the detected dynamic scenes preserve the class semantics and temporal structures in video sequences, the use of a time-series correlation analysis together with a quality measurement with dynamic programming allows the optimal alignment between

Table 2 The relationship among parameters n_s (number of surface clusters), n_t (number of templates), and the classification rate using the dynamic-pose model learning

Gesture class	k_p	50	100	200	Gesture Class	k_p	50	100	200
	k_s					k_s			
Opening	50	92 %	88 %	88 %	Marching	50	92 %	92 %	92 %
	100	92 %	88 %	88 %		100	96 %	92 %	92 %
	200	92 %	88 %	88 %		200	96 %	92 %	92 %
Kicking	50	92 %	96 %	92 %	Pointing at	50	100 %	96 %	96 %
	100	96 %	96 %	96 %		100	100 %	100 %	100 %
	200	96 %	92 %	96 %		200	100 %	100 %	100 %
Walking	50	100 %	93 %	88 %	Pushing	50	100 %	100 %	88 %
	100	100 %	93 %	81 %		100	100 %	100 %	100 %
	200	100 %	93 %	81 %		200	100 %	100 %	100 %
Crouching down	50	100 %	100 %	100 %	Moving apart	50	96 %	92 %	92 %
	100	100 %	100 %	100 %		100	96 %	92 %	92 %
	200	100 %	92 %	96 %		200	96 %	92 %	92 %
Picking up	50	100 %	100 %	100 %	Pulling	50	96 %	96 %	88 %
	100	100 %	100 %	100 %		100	96 %	96 %	88 %
	200	100 %	100 %	100 %		200	88 %	88 %	88 %

the dynamic scene model and the input video sequences. The alignment result is used to generate the final dynamic scene model with the help of the latent semantic learning.

The proposed approach has been validated with extensive tests and compared with related methods. The results prove that our solution is successful and effective even in the presence of noise or periodic activities in the datasets. The contributions of the proposed video detection and classification framework include: 1) the latent semantic learning enhances the discriminative power of the dynamic scene model; 2) both spatial and temporal structures in training video sequences are preserved by the dynamic scene model; 3) class information is used in the latent semantic learning in order to preserve scene semantics; 4) the video scene detection is simple and fast, which tackles the difficulties of the DTW-based video classification.

The main drawback of the proposed approach is that the start frames of the input video affect the performance of DTW. This is partially resolved in our approach. Since the variations of start fames in the input video are still large in many applications, the start frame may be

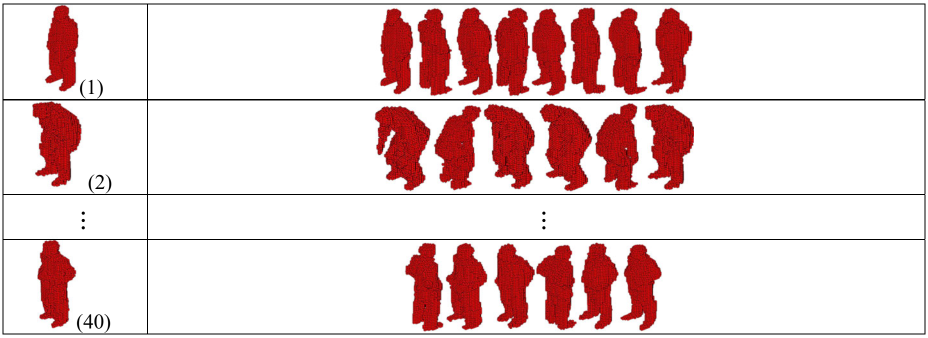


Fig. 9 The pose codebook

	Opening	**Kicking**	**Walking**	**Crouching down**	**Picking up**	**Marching**	**Pointing at**	**Pushing**	**Moving apart**	**Pulling**
Opening	**0.92**	0	0	0	0	0	0	0	0	0.08
Kicking	0	**0.96**	0	0	0	0.04	0	0	0	0
Walking	0	0	**1**	0	0	0	0	0	0	0
Crouching down	0	0	0	**1**	0	0	0	0	0	0
Picking up	0	0	0	0	**1**	0	0	0	0	0
Marching	0	0.04	0	0	0	**0.96**	0	0	0	0
Pointing at	0	0	0	0	0	0	**1**	0	0	0
Pushing	0	0	0	0	0	0	0	**1**	0	0
Moving apart	0.04	0	0	0	0	0	0	0	**0.96**	0
Pulling	0.04	0	0	0	0	0	0	0	0	**0.96**

Fig. 10 The confusion matrix of the proposed classifier

Table 3 Performance comparison between the proposed and Pierobon's method using three error types A, B, and C

Classes	Type A error		Type B error		Type C error	
	Pierobon's method	Proposed method	Pierobon's method	Proposed method	Pierobon's method	Proposed method
Opening	2	2	2	0	3	0
Kicking	0	1	1	0	1	0
Walking	1	0	3	0	3	0
Crouching down	0	0	0	0	0	0
Picking up	0	0	0	0	0	0
Marching	2	1	4	0	3	0
Pointing at	1	0	1	0	1	0
Pushing	0	0	2	0	1	0
Moving apart	2	1	1	0	1	0
Pulling	0	1	1	0	1	0
Total	8	6	15	0	14	0

aligned to wrong positions and cause the wrong results. Another problem is that what is the optimized number of templates for representing a class? These can be the future result topics to solve such problems.

Acknowledgments This work was supported in part by Ministry of Science and Technology, Taiwan, under grant numbers MOST 103-2221-E-019-018-MY2 and MOST 103-2511-S-130-003. The study has benefitted from Mr. Yun-Lun Chen for his technical support.

References

1. 3D Gesture Recognition Database (2014). Available: http://www-dsp.elet.polimi.it/ispg/index.php/description.html. Accessed 11 Jan 2015
2. Ali S, Shah M (2010) Human action recognition in videos using kinematic features and multiple instance learning. IEEE Trans Pattern Anal Mach Intell 32(2):288–303. doi:10.1109/TPAMI.2008.284
3. Ballan L, Bertini M, Bimbo AD, Seidenari L, Serra G (2011) Event detection and recognition for semantic annotation of video. Multimedia Tools Appl 51(1):279–302
4. Ballan L, Bertini M, Bimbo AD, Serra G (2010) Video event classification using string kernels. Multimedia Tools Appl 48(1):69–87
5. Barnachon M, Bouakaz S, Boufama B, Guillou E (2014) Ongoing human action recognition with motion capture. Pattern Recogn 47(1):238–247
6. Bhattacharyya A (1943) On a measure of divergence between two statistical populations defined by their probability distributions. Bull Calcutta Math Soc 35(1):99–109
7. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. 1395–1402 Vol. 1392. doi:10.1109/ICCV.2005.28
8. Branco JA, Croux C, Filzmoser P, Oliveira MR (2005) Robust canonical correlations: a comparative study. Comput Stat 20(2):203–269
9. Brezeale D, Cook DJ (2008) Automatic video classification: a survey of the literature. IEEE Trans Syst Man Cybern Part C Appl Rev 38(3):416–430. doi:10.1109/TSMCC.2008.919173
10. Caspi Y, Irani M (2002) Spatio-temporal alignment of sequences. IEEE Trans Pattern Anal Mach Intell 24(11):1409–1424. doi:10.1109/TPAMI.2002.1046148
11. Caspi Y, Simakov D, Irani M (2006) Feature-based sequence-to-sequence matching. Int J Comput Vis 68(1): 53–64
12. Chan AB, Vasconcelos N (2008) Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE Trans Pattern Anal Mach Intell 30(5):909–926. doi:10.1109/TPAMI.2007.70738
13. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3). doi:10.1145/1541880.1541882
14. Chang C-C, Lin C-J. (2001) LIBSVM: a library for support vector machines. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm. Accessed 11 Jan 2015
15. Chen Y-L, Cheng S-C, Chen PY-P (2012) Reordering video shots for event classification using bag-of-words models and string kernels. In: Proceeding of the 27th Conference on Image and Vision Comuputing
16. Cheng S-C, Kuo C-T, Wu D-C (2010) A novel 3D mesh compression using mesh segmentation with multiple principal plane analysis. Pattern Recogn 43(1):261–279
17. Chuang C-H, Cheng S-C, Chang C-C, Chen PY-P (2014) Model-based approach to spatial-temporal sampling of video clips for video object detection by classification. J Vis Commun Image Represent 25(5):1018–1030
18. Doretto G, Chiuso A, Wu YN, Soatto S (2003) Dynamic textures. Int J Comput Vis 51(2):91–109
19. Felzenszwalb PF, Girshick RB, McAllester D (2010) Cascade object detection with deformable part models. In: Computer Vision and Pattern Recognition (CVPR), 2010 I.E. Conference on. 2241–2248. doi:10.1109/CVPR.2010.5539906
20. Han D, Bo L, Sminchisescu C (2009) Selection and context for action recognition. In: ICCV
21. Hu R, Wang T, Collomosse J (2011) A bag-of-regions approach to sketch-based image retrieval. In: Image Processing (ICIP), 2011 18th IEEE International Conference on. 3661–3664. doi:10.1109/ICIP.2011.6116513
22. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Li F-F (2014) Large-Scale Video Classification with Convolutional Neural Networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 I.E. Conference on. 1725–1732. doi:10.1109/CVPR.2014.223

23. Klank U, Zia M, Beetz M (2009) 3D model selection from an internet database for robotic vision. In: Robotics and Automation, 2009. ICRA '09. IEEE International Conference on. 2406–2411. doi:10.1109/ROBOT.2009.5152488
24. Kläser A, Marszalek M, Schmid C (2008) A spatio-temporal descriptor based on 3D-gradients. In: British Machine Vision Conference
25. Kovashka A, Grauman K (2010) Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: IEEE Conference Computer Vision and Pattern Recognition (CVPR). 2046–2053
26. Laptev I, Caputo B, Schuldt C, Lindeberg T (2007) Local velocity-adapted motion events for spatio-temporal recognition. Comput Vis Image Underst 108(3):207–229
27. Laptev I, Lindeberg T (2005) On space-time interest points. Int J Comput Vis 64(2–3):107–123
28. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. 1–8. doi:10.1109/CVPR.2008.4587756
29. Lavee G, Rivlin E, Rudzsky M (2009) Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. IEEE Trans Syst Man Cybern Part C Appl Rev 39(5):489–504
30. Li L, Prakash B (2011) Time series clustering: complex is simpler! Proceedings of the 28th International Conference on Machine Learning (ICML-11):185–192
31. Ma Z, Yang Y, Sebe N, Zheng K, Hauptmann AG (2013) Multimedia event detection using a classifier-specific intermediate representation. IEEE Trans Multimedia 15(7):1628–1637. doi:10.1109/TMM.2013.2264928
32. Microsoft (2012) Kinect sdk. Available: http://www.microsoft.com/en-us/kinectforwindows/develop/. Accessed 11 Jan 2015
33. Nam Y, Rho S, Park J (2012) Intelligent video surveillance system: 3-tier context-aware surveillance system with metadata. Multimedia Tools and Applications 57(2):315–334
34. Niebles J, Wang H, Fei-Fei L (2008) Unsupervised learning of human action categories using spatial-temporal words. Int J Comput Vis 79(3):299–318
35. Park JH, Rho S, Jeong CS, Kim J (2013) Multiple 3D object position estimation and tracking using double filtering on multi-core processor. Multimedia Tools and Applications 63(1):161–180
36. Pierobon M, Marcon M, Sarti A, Tubaro S (2007) A human action classifier from 4-D data (3-D+time) based on an invariant body shape descriptor and Hidden Markov Models. In: Proc. International Conference on Signal Processing and Multimedia Applications. 406–413
37. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990
38. Raptis M, Kirovski D, Hoppe H (2011) Real-time classification of dance gestures from skeleton animation. In: Proc. of the 2011 ACM Siggraph/Eurographics Symposium on Computer Animation - SCA'11. 147–156
39. Ravichandran A, Vidal R (2011) Video registration using dynamic textures. IEEE Trans Pattern Anal Mach Intell 33(1):158–171. doi:10.1109/TPAMI.2010.61
40. Rodriguez MD, Ahmed J, Shah M (2008) Action mach a spatiotemporal maximum average correlation height filter for action recognition. In: Int'l Conf. Computer Vision and Pattern Recognition
41. Sun J, Wu X, Yan S, Cheong L-F, Chua T-S, Li J (2009) Hierarchical spatio-temporal context modeling for action recognition. In: CVPR
42. Wang F, Jiang Y, Ngo C (2008) Video event detection using motion relativity and viosual relatedness. In: Proceedings of ACM Multimedia. 239–248
43. Wang J, Liu Z, Wu Y, Yuan J (2012) Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition (CVPR), 2012 I.E. Conference on. 1290–1297. doi:10.1109/CVPR.2012.6247813
44. Wang H, Ullah MM, Kläser A, Laptev I, Schmid C (2009) Evaluation of local spatio-temporal features for action recognition. Proceeding 20th British Machine Vision Conference
45. Weng M-F, Chuang Y-Y (2012) Cross-domain multicue fusion for concept-based video indexing. IEEE Trans Pattern Anal Mach Intell (TPAMI) 34(10):1927–1941. doi:10.1109/TPAMI.2011.273
46. Yao A, Gall J, Gool LV (2010) A Hough transform-based voting framework for action recognition. In: IEEE Conf. Computer Vision and Pattern Recognition
47. Yao B, Nie B, Liu Z, Zhu S-C (2014) Animated pose templates for modeling and detecting human actions. IEEE Trans Pattern Anal Mach Intell 36(3):436–452. doi:10.1109/TPAMI.2013.144
48. Yao B, Zhu S-C (2009) Learning deformable action templates from cluttered videos. In: Computer Vision, 2009 I.E. 12th International Conference on. 1507–1514. doi:10.1109/ICCV.2009.5459277
49. Yeffet L, Wolf L (2009) Local trinary patterns for human action recognition. In: Computer Vision, 2009 I.E. 12th International Conference on. 492–497. doi:10.1109/ICCV.2009.5459201

SHYI-CHYI CHENG received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan in 1986, and the M.S. and Ph.D. degrees in Electronics Engineering and Computer Science and Information Engineering in 1988 and 1992, respectively, both from National Chiao Tung University, Hsinchu, Taiwan. From 1992 to 1998, he was a Technical Staff at the Chunghwa Telecom Laboratories, Taoyuan, Taiwan. He joined the faculty of Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan from 1999 to 2005. He is currently a Professor with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan. His research interests include multimedia databases, image/video compression and communications, and intelligent multimedia systems.

Jui-Yuan Su, Assistant Professor of the Department of Communication Management at Ming Chuan University, Taiwan, has received his B.S. in Computer Science and Engineering from Tatung Institute of Technology (Tatung University) in September 1993, and M.S. in Computer Science from National Chiao-Tung University in September 1995. He is currently doing his Ph.D. research in the Department of Computer Science and Engineering, National Taiwan Ocean University. His research interests include intelligent multimedia system, sensor technologies, internet/ network computing, and embedded system.

Kuei-Fang Hsiao, Associate Professor of the Department of Information Management at Ming Chuan University (MCU), Taiwan, received B.Sc. degree from National Taiwan University, Taiwan in 1991 to follow her Postgraduate and Research Qualifications at the University of Manchester, UK, awarded M.Ed. and Ph.D. in 1993 and 1998 in Information Education, respectively. Dr Hsiao joined MCU since Aug 1998 to build her own research area for applications of computerised systems of using the augmented reality (AR) to elevate the deteriorating health of students in the very competitive educational environment dominating the country. Further extension of applications of AR and Sensor technologies includes medical and psychological health of elderlies and use of smartphones.

Professor Rashvand is a Chartered Engineer and Life member of the IEEE. Following his distinguished engineering qualifications in the early 1970s in association with the University, PTT, NTT, KTT and many Industries helped building a new Telecom Research Centre (ITRC). He then joined the University of Kent at Canterbury for his highly commented degrees in 1977 and 1980. After 20 years of mix and solid experience in industrial and educational organizations he received his professorship of ‘Networks, Systems & Protocols’ from German Ministry of Education in 2001. His experience includes Racal, Vodafone, Nokia and Cable & Wireless and Universities of Tehran, Zambia Southampton, Reading, Portsmouth, Coventry University, Magdeburg and Warwick with overlapping 10 years with the Open University. Chief Editor since 1998 for 15 years including IEE-COM/IFS, IET-COM and IET-WSS. Director of Advanced Communication Systems, University of Warwick has Special Interests in technological innovation, wireless sensors and distributed systems.