Pattern Recognition Letters 34 (2013) 1936–1944

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Temporal segmentation and assignment of successive actions in a long-term video

Guoliang Lu *, Mineichi Kudo, Jun Toyama

Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan

ARTICLE INFO

Article history:
Available online 10 November 2012

Keywords:
Action segmentation
Characteristic frames
Viterbi algorithm

ABSTRACT

Temporal segmentation of successive actions in a long-term video sequence has been a long-standing problem in computer vision. In this paper, we exploit a novel learning-based framework. Given a video sequence, only a few characteristic frames are selected by the proposed selection algorithm, and then the likelihood to trained models is calculated in a pair-wise way, and finally segmentation is obtained as the optimal model sequence to realize the maximum likelihood. The average accuracy on IXMAS dataset reached to 80.5% at frame level, using only 16.5% of all frames in computation time of 1.57 s per video which has 1160 frames on the average.

1. Introduction

Human activity analysis has been an attractive and popular research topic in recent two decades. Most previous works are concentrating on recognizing classes/categories of actions performed in an input video, independently of background. In these works, many significant progresses have been reported with satisfactory experimental results, but their experiments are mostly carried out under well-controlled situations such as seen in WEIZMANN (Blank et al., 2005) and KTH (Schuldt et al., 2004) datasets where short-term clips of single action (manually segmented/aligned) are provided. In real-world applications, however, human activity is observed in a continuous flow of multiple actions. Moreover, in general we cannot assume any prior knowledge of categories, temporal or spatial extents of performed action(s). A human activity is something like follows: a person steps into a room, picks up something to drink from a refrig, sits down on a sofa for a little break and stands up. Given such a video containing a variety of actions in a successive way (walking, picking up, sitting down and standing up, etc.), we have to segment it into individual actions as a natural demand as seen in action-based video index/classification, event recording and vision-surveileance management.

One common and standard approach is as follows: First, in the training phase, a set of features (e.g., interested point (Kovashka and Grauman, 2011), HoG (Thurau and Hlavác, 2008), optical flow (Fathi and Mori, 2008)) from each frame in the training sequences is extracted, and then individual actions are modeled using these features by some statistical or geometrical methods, e.g, HMM (Ahmad and Lee, 2008), SVM (Hoai et al., 2011). When a newly observed sequence is appeared in the evaluation phase, all frames of the sequence are firstly evaluated their probabilities according to the learned action models and segmentation result is obtained by solving a global optimization problem (Hoai et al., 2011; Lv and Nevatia, 2007) or a local optimization problem (Ogale et al., 2007; Jia and Yeung, 2008). This approach has succeeded in some practical problems (e.g., view-invariance (Weinland et al., 2007), activity modeling (Wang and Suter, 2007), fast matching (Shakhnarovich et al., 2003)). However, there are still some issues to be considered in order to increase its practical value (Poppe, 2010). In this study, we consider two aspects as below:

(a) It is redundant to use every frame in a video sequence, because neighboring frames are highly correlated (very similar) on the temporal domain. Moreover, such a *frame-by-frame* comparison is computationally expensive. In fact, it is sometimes reported that only a few frames in an input video are sufficient for action discrimination (Schindler and Van-Gool, 2008; Weinland and Boyer, 2008).

(b) Single-frame based representation is sufficient for modeling of human actions only where videos contain one single action. For videos containing more than one action, this approach would not work, since different actions can share very similar frames in part (Fig. 1).

To cope with these problems, the following two techniques are proposed in this study (Fig. 2):

(a) Given a long-term video sequence, just a few frames are selected by a martingale framework proposed in our prior work (Lu et al., 2012), which is executed without requiring any prior knowledge of possibly performed action(s). Such frames are called *characteristic frames*, here.

[*] Corresponding author. Address: Graduate School of Information Science and Technology, Hokkaido University, Kitaku Kita14 Nishi9, Sapporo 060-0814, Japan. Tel.: +81 11 706 6854; fax: +81 11 706 7393.

E-mail addresses: luguoliang@main.ist.hokudai.ac.jp (G. Lu), mine@main.ist.hokudai.ac.jp (M. Kudo), jun@main.ist.hokudai.ac.jp (J. Toyama).

0167-8655/$ - see front matter Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.10.023

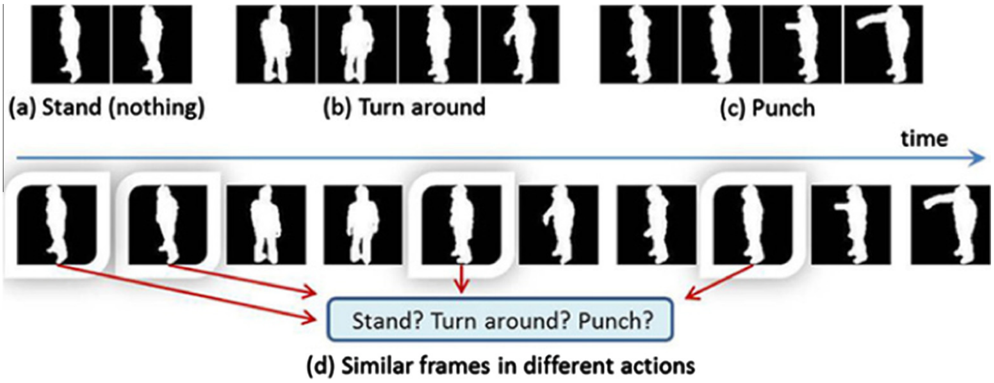

(a) Stand (nothing)
(b) Turn around
(c) Punch
time
Stand? Turn around? Punch?
(d) Similar frames in different actions

Fig. 1. Typical frames taken from actions of **stand** (a), **turn-around** (b) and **punch** (c). For the case of videos containing one single action, these frames can be recognized correctly. However, in a clip (d) containing multiple successive actions, similar frames appearing in different action contexts are not identified correctly sometimes.

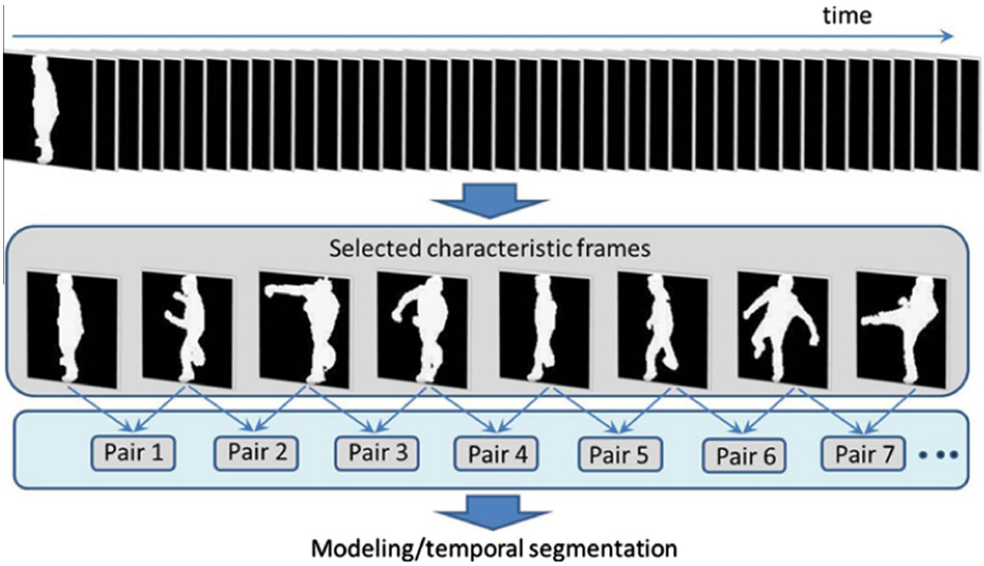

time
Selected characteristic frames
Pair 1
Pair 2
Pair 3
Pair 4
Pair 5
Pair 6
Pair 7
Modeling/temporal segmentation

Fig. 2. The proposed process: given a long-term video sequence, only a small number of characteristic frames are selected, and then pairwise based comparison to the models of actions has carried out.

(b) For modeling/segmenting actions, pairwise-frame representation using characteristic frames is employed to describe the given video sequence.

Since we use a pairwise-frame representation instead of the single-frame based representation, the time differentiated information in two neighboring characteristic frames brings a higher level of discriminative information among actions. In addition, a smaller number of frames selected in the whole video sequence brings an efficiency.

The rest of this paper is organized as follows: Section 2 reviews the related works. In Section 3, selection procedure of characteristic frames is presented. Section 4 indicates how to model a human activity using characteristic frames. The detailed description on the temporal segmentation of successive actions is given in Section 5. Section 6 presents the experimental results on IXMAS dataset, followed by the discussion in Section 7. Section 8 concludes this paper and shows the future work.

2. Related work

Recent efforts on successive actions segmentation fall into five approaches roughly (Hoai et al., 2011). As the first approach, change-point detection based actions segmentation (Xuan and Murphy, 2007; Harchaoui et al., 2009) is the most popular and is based on change-point analysis with a sliding window along the time extent. Xuan and Murphy (2007) modeled the joint density of vector-valued observations using undirected Gaussian graphical models by which the location of change points are detected by computing the MAP segmentation. Harchaoui et al. (2009) proposed a test statistic based upon the maximum kernel Fisher discriminant ratio as a measure of homogeneity between segments, which allows to build a statistical hypothesis test procedure for detecting change-points from an unlabeled sample of observations. The change-point detection can detect local changes in one action, but it is often weak for detection of global changes in the whole video.

Cyclic motion analysis is the second approach to segment events by analyzing periodicity of cyclic events (Laptev et al., 2005; Cutler and Davis, 2000). Laptev et al. (2005) exploited periodicity as a cue and detected periodic motions in complex scenes. Cutler and Davis (2000) proposed an assumption that the self-similarity measure of periodic motion is also periodic and that periodic motion can be detected and characterized by applying time–frequency analysis. Cyclic motion analysis based events segmentation is applicable for repetitive actions with discriminative cyclic. However, actions appearing in a complex cyclic manner are difficult to be segmented correctly.

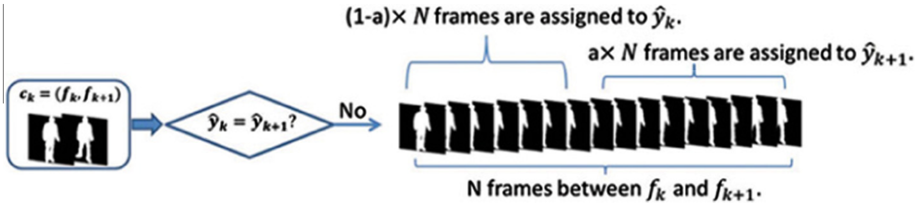

(1-a)× N frames are assigned to $\hat{y}_k$.
a× N frames are assigned to $\hat{y}_{k+1}$.
$c_k = (f_k, f_{k+1})$
$\hat{y}_k = \hat{y}_{k+1}$?
No
N frames between f_k and f_{k+1}.

Fig. 3. Temporal segmentation with the most likely path (ψ^*). Assuming that $(\hat{y}_k, \hat{y}_{k+1})$ is the searched model label of one pair $c_k = (f_k, f_{k+1})$ in (ψ^*) and there are N frames between the two characteristic frames f_k and f_{k+1} in this pair, if $\hat{y}_k = \hat{y}_{k+1}$, the N frames are assigned to $\hat{y}_k$; Otherwise, these frames are separated into actions of $\hat{y}_k$ or $\hat{y}_{k+1}$ according to a pre-determined ratio a.

Third, action segmentation by clustering or by grouping frames has also proposed in which a cluster is expected to represent a single action (Zelnik-Manor and Irani, 2006; Loui and Savakis, 2000). Zelnik-Manor and Irani (2006) proposed a statistical behavior-based distance measure between video sequences which captures the similarities in their behavioral contents, by which frames are grouped into similar behavior frames. Loui and Savakis (2000) created an event segmentation algorithm to automatically cluster pictures into events and sub-events for albumin on the basis of date/time meta data information as well as color content of the pictures. This approach, however, lacks a mechanism to incorporate the dynamics of temporal events in the clustering process (Hoai et al., 2011).

The fourth approach is exemplar-matching based approach and it is a way to segment successive actions by evaluating similarity of each newly arrived frame to exemplars representing each of actions (Lv and Nevatia, 2007). Lv and Nevatia (2007) modeled a human action as a series of synthetic 2D human poses as exemplars and segmentation is accomplished under constraints on the transition of the synthetic poses which is represented by a graphical model called Action Net. One drawback of this approach is that this method does not consider the temporal correlation between successive poses, consequently, e.g., **stand** action could not be discriminated from the **standing** poses/frames appearing in another action (e.g., the action of **turn-around**, as seen in Fig. 1).

The last approach is learning based approach which employs a classifier for segmentation such as SVM (Hoai et al., 2011), HMM (Boykin and Merlino, 2000). Hoai et al. (2011) proposed an approach using the spatial bag-of-words model for representing each frame, in which classification is performed by a multi-class SVM. Boykin and Merlino (2000) applied HMM to perform event (story and advertisement) segmentation. One drawback of these learning based approach is the requirement of fully labeled data for training. Unfortunately labeling of every frame is typically expensive and requires much burden of human inspectors. In comparison with other ways, this approach often gives a fast and natural solution for action segmentation.

In this study, we employ the learning-based approach for temporal segmentation of successive actions, since it has advantages on practical efficiency which is most expected in our work.

3. Selection of characteristic frames in a video

An efficient way of selecting characteristic frames in a given video sequence has been proposed by the authors (Lu et al., 2012). That selection way is supported by two basic ideas. The first one is, an observed video sequence can be sufficiently characterized by few characteristic frames for describing basic actions; The other one is, by considering the input video sequence as a set of data streams in which successive frames are almost the same, the characteristics frames can be detected as the change frames between two successive streams. Such change frames are detected by testing *exchangeability* of a Martingale. This idea is based on the following three points:

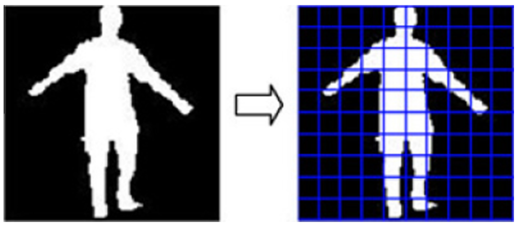


Fig. 4. Block-based representation of a normalized silhouette with a size of 100×100 pixels (center-orientation). First, the silhouette extracted in one frame f_i is divided into nonoverlapping blocks with a fixed size of $B \times B$ pixels. Then, in each block (bin), feature is computed as $b(i) = \frac{\#\{i\}}{m}, i \in \{1, 2, \ldots, D\}$ assuming there are D blocks in this silhouette, where $\#\{i\}$ is the sum of foreground pixels in the ith block and m is the maximum value of all blocks; Finally, this frame f_i is represented by a vector of $h_i = [b(1), b(2), \ldots, b(D)]^T$.

(a) The changes are detected by testing the null hypothesis that all n (strangeness) values $s_1, s_2, \ldots, s_n$ are exchangeable in the index, through the corresponding exchangeability martingale $M_1, M_2, \ldots, M_n$, where M_n is a measurable function of $s_1, s_2, \ldots, s_n$ satisfying

$$M_n = E(M_{n+1} | M_1, M_2, \ldots, M_n). \quad (1)$$

(b) The following Doob's inequality can be used for rejecting this null hypothesis for a large value of M_n:

$$P(\exists n | M_n \geqslant \lambda) \leqslant 1/\lambda. \quad (2)$$

(c) This (exchangeability) martingale is constructed from a p-value, the probability of obtaining a test statistic at least as extreme as the one that was actually observed, and the p-value is obtained by a strangeness value appropriately determined in each specific application.

We here describe the outline of this selection way as below. For the details, please see the previous work Lu et al. (in press).

3.1. Frame representation

Given a video sequence F of n frames, i.e., $F = \{f_1, f_2, \ldots, f_n\}$, a time series of block-based features (as illustrated in Fig. 4) $H = \{h_1, h_2, \ldots, h_n\}$ is calculated, where h_i is the extracted feature from frame f_i.

3.2. Strangeness evaluation of observed frames

Given the feature series up to the $(i-1)$th frame $H_{i-1} = \{h_1, h_2, \ldots, h_{i-1}\}, i \leqslant n$, the strangeness s_i of a newly observed frame h_i against a model generated from H_{i-1} is measured as

$$s_i = s(H_{i-1}, h_i) = ||h_i - \mu_{i-1}||, \quad \mu_{i-1} = \sum_{j=1}^{i-1} h_j / (i-1), \quad (3)$$

where μ_{i-1} is the center of model and $|| \cdot ||$ is a metric defined as Euclidean distance in this study.

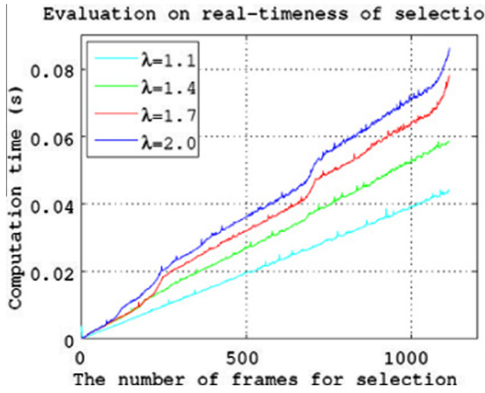


Fig. 5. Computation time of the procedure of selecting characteristic frames as the number of frames increases. The values of λ is $\lambda = 1.1, 1.4, 1.7, 2.0$. It is noted that the computation time shown here includes only two components of strangeness calculation and change detection.

On the basis of the strangeness values $s_1, s_2, \ldots, s_n$, we construct a family of Martingale indexed by $\epsilon \in [0,1]$ as Vovk et al. (2003), as seen in Fig. 6:

$$M_n^{(\epsilon)} = \prod_{i=1}^{n} (\epsilon \hat{p}_i^{\epsilon-1}), \tag{4}$$

where the $\hat{p}_i$'s are the $\hat{p}$-values calculated from

$$\hat{p}_i(\{h_1, h_2, \ldots, h_i\}, \theta_i) = \frac{\#\{j : s_j > s_i\} + \theta_i \#\{j : s_j = s_i\}}{i}. \tag{5}$$

Here, the value of θ_i is a random value uniformly distributed over $[0,1]$ (fixed to 0.5 in the following experiments due to the relatively less number of data) and $\#\{j\}$ indicates the number of j. It is noted that $M_i^{(\epsilon)} = \epsilon \hat{p}_i^{\epsilon-1} M_{i-1}^{(\epsilon)}$. Therefore, no re-computation is needed for $\hat{p}_j, j \in \{1, 2, \ldots, i-1\}$ in order to compute $M_i^{(\epsilon)}$. The initial Martingale value is set to $M_0^{(\epsilon)} = 1$.

3.3. Detection on change frames

When a new frame f_n represented by h_n with strangeness value s_n is observed, a Martingale test below takes place to decide whether a change occurs or not:

$$M_n^{(\epsilon)} \geqslant \lambda \quad \text{or} \quad 0 < M_n^{(\epsilon)} < \lambda, \tag{6}$$

where λ is a positive threshold. A *change* is detected, at time n, if $M_n^{(\epsilon)} \geqslant \lambda$, otherwise *no change* is detected. Once a *change* is detected, this frame h_n is selected to be one characteristic frame and then a new Martingale starts with this frame by initializing the Martingale value (as illustrated in Fig. 6). Here it is noted that more changing points, i.e., more characteristic frames, are detected if we adopt a lower value of λ and less for a larger value. Specially, all frames will be theoretically selected as characteristic frames when $\lambda \leqslant 1$.

4. Supervised training for human activity model

We will describe a way to learn models for individual actions and transitions between two successive actions from a collection of videos including many successive actions. Given a video sequence $F = \{f_1, f_2, \ldots, f_n\}$ of n frames with correct action labels $\{y_1, y_2, \ldots, y_n\}$, we extract m characteristic frames $F^c = \{f_{c_1}, f_{c_2}, \ldots, f_{c_m}\} \subseteq F$ $(m \leqslant n)$ by above Martingale test. For simplicity, we regard F^c as F. Then we couple the characteristic frames pairwise such as

$$G = \{(f_1, f_2), (f_2, f_3), \ldots, (f_{m-1}, f_m)\}, \tag{7}$$

as well as the corresponding label pairs

$$L = \{z_1 = (y_1, y_2), \ldots, z_{m-1} = (y_{m-1}, y_m)\}, y_i \in \zeta, \tag{8}$$

where ζ is the label set of possible actions. It should be noted that y_i can be identical to y_{i+1}.

The characteristic frame set G is furthermore converted to a feature representation set

$$H = \{c_1 = (h_1, h_2), \ldots, c_{m-1} = (h_{m-1}, h_m)\}, \tag{9}$$

where h_i is the block-based feature representation of ith characteristic frame f_i and c_i is the pair of h_i and h_{i+1}.

By collecting all training video sequences $F_1, F_2, \ldots, F_N$, we have $C = \bigcup_{j=1}^{N} H_j$ where H_j is the jth feature representation set in (9). If $y_i = y_{i+1}$ for a couple of frames $c_i = (h_i, h_{i+1}) \in C$, then it means that more than one characteristic frame are chosen from one action y_i (i.e., y_{i+1}), while in the case of $y_i \neq y_{i+1}$, the corresponding pair f_i and f_{i+1} (h_i and h_{i+1}) shows a transition from one action y_i to another action y_{i+1}. Moreover, the pairs from the same action imply multiple appearances of the action, since characteristic frames are usually dissimilar to each other to some extent. In contrast, the transitional pairs between two different actions are somewhat similar to each other because of the continuity of frames. We model all those training pairs with Gaussian Mixture Model (GMM). A pair $c_i = (h_i, h_{i+1})$ is assumed to be generated according to a Gaussian mixture distribution of K components in $2D$ dimensions (here, D is the number of blocks/bins in one frame feature-representation h, as seen in Fig. 4) as below:

$$p(c_i|\varphi_t) = \sum_{k=1}^{K} w_k^t N(c_i|\mu_k^t, \Sigma_k^t), \tag{10}$$

where the parameter set $\varphi_t = \{w_k^t, \mu_k^t, \Sigma_k^t\}_{k=1}^{K}$ is the one for the tth trained model. In this paper, the value of K is chosen from 3 to 6 when c_i is a one-action pair (i.e., $y_i = y_{i+1}$) and $K = 3$ when c_i is a transitional pair (i.e., $y_i \neq y_{i+1}$). In addition, we assume that two frames in one pair are independent, consequently the covariance matrix Σ^t can be represented as $\Sigma^t = \begin{bmatrix} \Sigma^{t,1} & 0 \\ 0 & \Sigma^{t,2} \end{bmatrix}$, where $\Sigma^{t,1}$ is the covariance matrix for the first frame, and $\Sigma^{t,2}$ for the second frame. The Expectation–Maximization (EM) algorithm is used for parameter estimation. It should be noted that theoretically there are $T^*(= T + T(T-1)/2)$ GMMs: T for individual actions and $T(T-1)/2$ for possible pairwise transitions. However, in real-world applications, there is sometimes no transition between some action pairs (e.g., no transition exists from *sit-down* action to *run* action). Therefore, the actual number of necessary models is far less than the theoretical number.

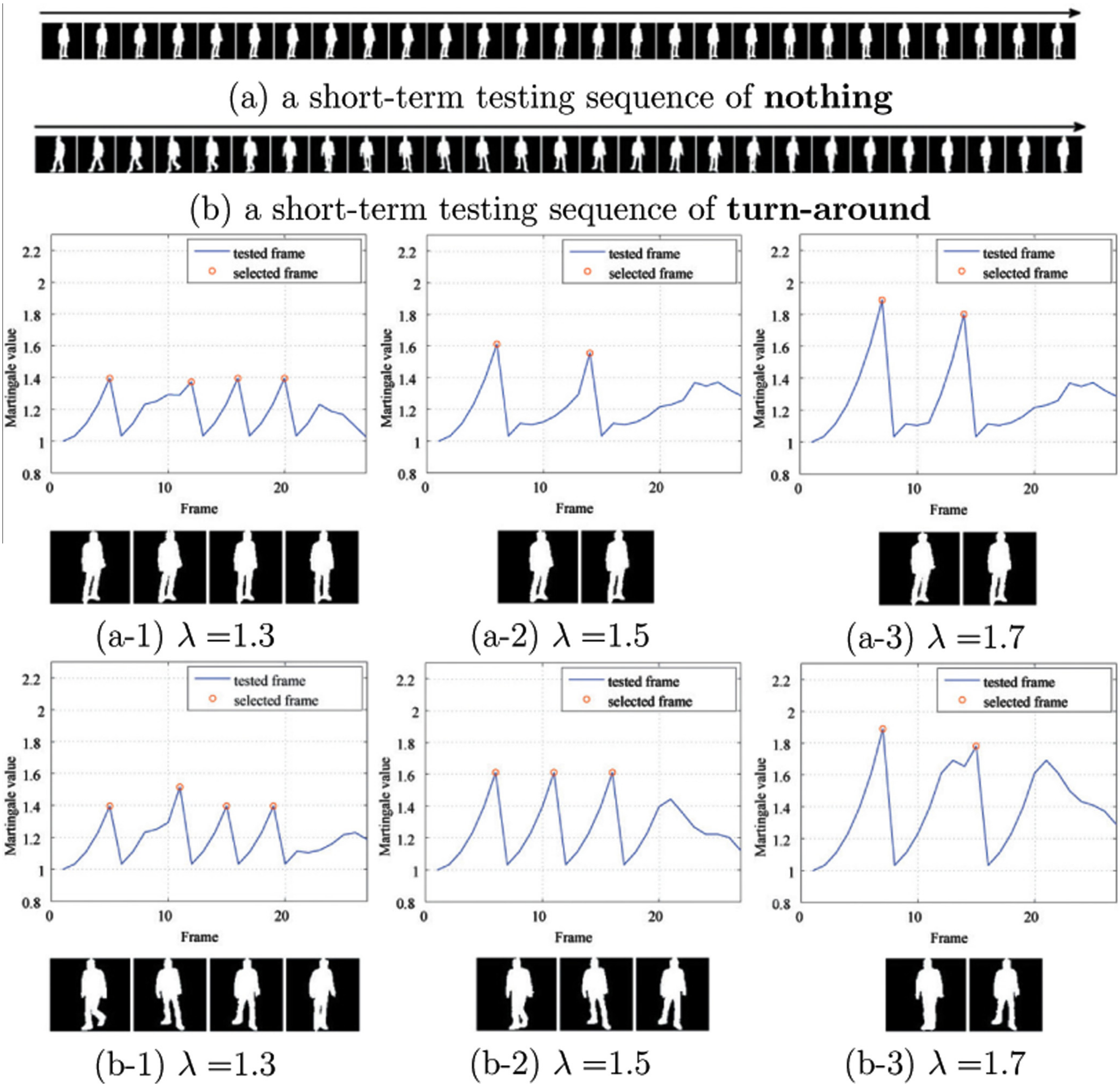
5. Temporal segmentation of successive actions

5.1. Probability computation

With GMMs learned from training sequences, a newly observed video sequence X is processed as follows. The characteristic frames $\{f_1, f_2, \ldots, f_m\}$ are selected firstly and then a series of pairs $H = \{c_1 = (h_1, h_2), c_2 = (h_2, h_3), \ldots, c_{m-1} = (h_{m-1}, h_m)\}$ is generated according to the same procedure as used in the training phase. Then the posterior probability of GMM φ_t given c_i $(i = 1, 2, \ldots, m-1)$ is calculated by Bayes' rule as

$$p(\varphi_t|c_i) = \frac{p(c_i|\varphi_t)p(\varphi_t)}{\sum_{t=1}^{T^*} p(c_i|\varphi_t)p(\varphi_t)}, \tag{11}$$

where T^* is the number of trained models of possible T actions and all possible transitions between them.


(a) a short-term testing sequence of **nothing**

(b) a short-term testing sequence of **turn-around**

(a-1) $\lambda = 1.3$ (a-2) $\lambda = 1.5$ (a-3) $\lambda = 1.7$

(b-1) $\lambda = 1.3$ (b-2) $\lambda = 1.5$ (b-3) $\lambda = 1.7$

Fig. 6. Selecting characteristic frames from videos of **nothing** (**stand**) and **turn-around** actions. From top to bottom: (a) a short-term testing sequence of **nothing**, (b) a short-term testing sequence of **turn-around**, and Martingale values of each frame and selected characteristic frames for $\lambda = 1.3$ (a-1, b-1), 1.5 (a-2, b-2) and 1.7 (a-3, b-3), respectively.

5.2. Searching the most likely path

Our goal is to find the model sequence $\psi = (\varphi_1, \varphi_2, \ldots, \varphi_{m-1})$ under which the probability of the feature sequence $H = \{c_1, c_2, \ldots, c_{m-1}\}$ is maximized. It is obtained by solving

$$\psi^* = \arg\max_\psi P(H|\psi) = \arg\max_\psi P(\psi|H)P(H). \tag{12}$$

We assume a constant prior $P(H)$ and consider only the second degree dependence on the basis of simple hidden Markov models as

$$\begin{aligned} P(\psi|H)P(H) &\propto P(\psi|H) = P(\varphi_1, \varphi_2, \ldots, \varphi_{m-1}|h_1, h_2, \ldots, h_m) \\ &= \prod_{i=1}^{m-1} P(\psi_i|h_i, h_{i+1}) = \prod_{i=1}^{m-1} P(\psi_i|c_i). \end{aligned} \tag{13}$$

Taking the logarithm, we have

$$(\psi^*) = \arg \max_{(\varphi_1, \ldots, \varphi_{m-1})} \sum_{i=1}^{m-1} \log P(\varphi_i|c_i). \tag{14}$$

We solve this problem using Viterbi algorithm. The obtained (ψ^*) is called the *most likely path.*

5.3. Temporal segmentation

Once the most likely path (ψ^*) is obtained for a given video sequence, the obtained series of action labels is used to segment this sequence. Let us assume that we have original N frames of which both ends are characteristic frames f_k and f_{k+1} with estimated action labels $\hat{y}_k$ and $\hat{y}_{k+1}$. We label all the frames with $\hat{y}_k$ if $\hat{y}_k = \hat{y}_{k+1}$, otherwise label the first $(1-a)N$ frames with $\hat{y}_k$ and the remaining aN frames with $\hat{y}_{k+1}$ (Fig. 3). The ratio $a \in [0,1]$ is determined by the empirical ratio obtained from the training videos and their characteristic frames.

6. Experiment

6.1. Database

The proposed framework was validated on the publicly available multi-view IXMAS database (Weinland et al., 2007) containing 180 video sequences ($36\ shots \times 5\ views$) in total. In each sequence, one of 12 actors performed 15 actions in a successive way. This database includes 2D data (the resolution is 160×120 pixels) consisting of image sequences and 2D silhouette sequences.

6.2. Experiments setup and Implement

In the following two sub-sessions we firstly evaluated the efficiency of the way of selecting characteristic frames (Section 6.3), and then compared the segmentation performance with the state-of-art approach with the same task/goal (Section 6.4). In these two experiments, we set $\epsilon = 0.90$ since any value of $\epsilon \in [0.80, 1.00)$ for Martingale series (4) was demonstrated its validation in (Ho and Wechsler, 2010). While, the value of λ in Martingale test (6) was varied from 1 to 2, empirically. The PC for the experiments is CPU3.10 GHz RAM4.0 GB. We implemented the proposed framework in Matlab without any optimization for speeding up the procedure.

As stated earlier, in this study a block-based description is computed frame by frame for frame representation. Some studies use instead flow computation based descriptions (e.g., motion based description (Weinland et al., 2006) and optical flow based description (Fathi and Mori, 2008)) for frame representation, but these descriptions need a specialized hardware for processing videos in real time (Wang et al., 2003). Therefore, we used such a block-based frame representation (Wang and Suter, 2007), as seen in Fig. 4. The computation time shown in Table 1 guarantees the real-timeness of our feature extraction method.

6.3. Evaluation on selection of characteristic frames

In the following sub-sessions, we will confirm the real-timeness of the characteristic frame selection and its robustness against a large variety of actions.

6.3.1. Evaluation on real-timeness

Since we use only a small number of characteristic frames instead of the entire video sequence, the temporal segmentation process is very fast. However, the selection procedure of characteristic frames requires an additional processing time, consequently the real-timeness has to be evaluated. This procedure includes three components (Section 3): frame representation, strangeness calculation and change detection. Since the real-timeness of feature extraction was evaluated already, we will evaluate the additional computation cost of the two residual components.

First, we chose at random a video sequence of a length of 1163 frames. Then, a number of sub-clips of 1 to 1163 frames were extracted. We have measured the time consumed by the frame selection procedure with several values of λ, that is, $\lambda = 1.1, 1.4, 1.7$ and 2.0. The result is shown in Fig. 5 and it is observed that the time increases almost linearly in the number of frames. In addition, the efficiency is almost proportional to the reciprocal value of λ. Since even the largest amount of computation time is less than 0.085 s (for $\lambda = 2.0$ on the sub-clip with 1163 frames), we could conclude that our selection procedure is sufficiently fast for real-time processing.

Table 1
Computation time [ms/frame] of block-based feature (Wang and Suter, 2007).

Size of a block [pixels]	4×4	5×5	10×10	20×20
Computation time	1.9	1.2	0.37	0.13

6.3.2. Evaluation on robustness against various actions

We have already demonstrated in (Lu et al., in press) the validation and priority of the selection procedure of characteristic frames for video sequences containing one single action, but not for long-term video sequences containing multiple successive actions. In such long-term videos, there is a large variety of temporal changes, that is, only a slight change of silhouettes occurs in some actions (e.g., **nothing** (**stand**) in Fig. 6(a)) but a large change can occur in the other actions (e.g., **turn-around** in Fig. 6(b)), which is different from the relatively stable frame-change in videos containing one action. Therefore, a selection way is expected to be robust enough for this frame-change, i.e., the numbers of selected characteristic frames extracted from various actions would be almost the same, or at least, not differ greatly.

Fig. 6 shows the selected frames in two actions of **nothing** and **turn-around** for values of λ = 1.3, 1.5 and 1.7. It is noted that almost the same pose is kept in **nothing**, while large changes of poses are observed in **turn-around**. In Fig. 6(a-1, 2, 3), we see that 2–4 frames are selected even in this almost motionless action. In **turn-around** action in Fig. 6(b-1, 2, 3), 2–4 frames are also selected for several values of λ. From these observations, we can see that our selection procedure is capable to characterize actions with different degrees of frame-change. Here, it is noted that the results in Fig. 6 (a-1, 2, 3) are different from those of most previous works (e.g., in the work (Lv and Nevatia, 2007), only one pose is extracted for modeling of **nothing** action). Our result gives an advantage on discriminating similar frames/poses appearing in different actions using a pairwise-frame representation, as seen in Fig. 7.

Here, it is noted that although any numerical evaluation has not been conducted on these results, the validity of selected characteristic frames is confirmed by the accuracy of action assignment that will be shown in the following.

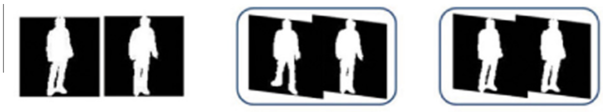
6.4. Performance on temporal segmentation

In the proposed framework, the leading parameters that affect the efficiency and performance are the block size B in frame representation and the Martingale threshold λ for selection of characteristic frames. It is clear that a larger B makes the frame representation more robust against noise but loose precision and resolution to some extent. Similarly, the value of λ controls the balance between efficiency and the amount of information for segmentation.

We have examined several combinations of values of (B, λ). The value of B ranges for 4×4, 5×5, 10×10 and 20×20, and the value of λ ranges from 1 to 2 by step of 0.1. In each combination, the experiment is carried out as follows: for given video sequences, few characteristic frames were firstly selected and then GMMs of pairwise characteristic frames were learned in the training phase and the temporal segmentation was performed for a newly given video sequence in the test phase. Evaluation was made by leave-one-out cross-validation.

6.4.1. Experimental results

For the tested combinations of (B, λ) stated above, the corresponding accuracies of action assignment, compression rates and



(a) Two 'standing' frames. (b) A pair of 'turn around'. (c) A pair of 'nothing'.

Fig. 7. Given one respective frame of **standing** (a), it is difficult to certify the performed action in it. However, a pairwise-frame representation gives a differentiated information in temporal extent, by which we could confirm whether such a **standing** frame is from **turn-around** (b) or **nothing** (c), i.e., the performed action in it is identified correctly.

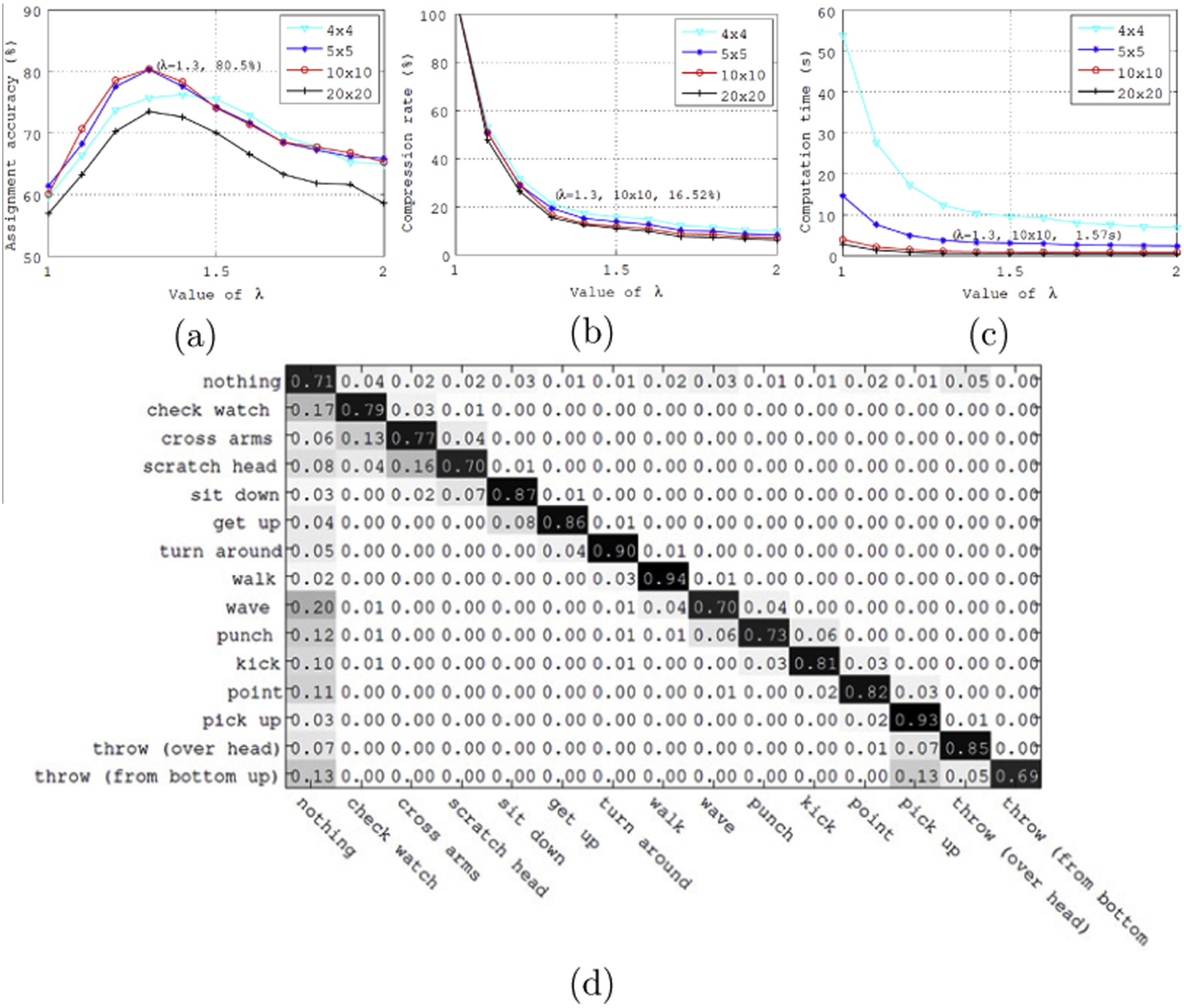
the computation time are shown in Fig. 8. Here, the accuracy of action assignment at frame level is measured as:

$$\text{Accuracy} = \frac{\text{Number of frames with correct assignment}}{\text{Number of total frames of this action}} (\%) \tag{15}$$

The compression rate in this study is defined as:

$$\text{Compression rate} = \frac{\text{Number of selected characteristic frames}}{\text{Number of total frames in a video}} (\%) \tag{16}$$

We can see that the best accuracy of 80.5% is attained at a block size of 10 × 10 and $\lambda = 1.3$ (Fig. 8(a)). To this best accuracy, 16.5% frames are chosen on the average as characteristic frames (Fig. 8(b)) and the computation time is 1.57 s (Fig. 8(c)) including feature extraction, selection of characteristic frames and actions segmentation, except for silhouette-extraction (since the silhouettes have been provided in IXMAS dataset). These performance values show that our proposed framework has a good validation of temporal segmentation of successive actions and a satisfactory efficiency for practical usage. The IXMAS database includes 5 camera views, therefore we can show the corresponding accuracy of each view to the overall best. The result is shown in Table 2. It is observed that the proposed framework performed best in **Cam.4**, which probably means that this top-view gives most discriminative appearances in selected characteristic frames of 15 actions. Fig. 8(d) shows a confusion matrix of action assignment at frame-level with the best accuracy. From Fig. 8(d), we can see that among all 15 action classes, action discrimination of **pick-up** and **walk** is the easiest. It implies that these two actions are very contrastive to other ones. It is also observed that some actions are wrongly assigned



	nothing	check watch	cross arms	scratch head	sit down	get up	turn around	walk	wave	punch	kick	point	pick up	throw (over head)	throw (from bottom
nothing	0.71	0.04	0.02	0.02	0.03	0.01	0.01	0.02	0.03	0.01	0.01	0.02	0.01	0.05	0.00
check watch	0.17	0.79	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
cross arms	0.06	0.13	0.77	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
scratch head	0.08	0.04	0.16	0.70	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
sit down	0.03	0.00	0.02	0.07	0.87	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
get up	0.04	0.00	0.00	0.00	0.08	0.86	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
turn around	0.05	0.00	0.00	0.00	0.00	0.04	0.90	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
walk	0.02	0.00	0.00	0.00	0.00	0.00	0.03	0.94	0.01	0.00	0.00	0.00	0.00	0.00	0.00
wave	0.20	0.01	0.00	0.00	0.00	0.00	0.01	0.04	0.70	0.04	0.00	0.00	0.00	0.00	0.00
punch	0.12	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.06	0.73	0.06	0.00	0.00	0.00	0.00
kick	0.10	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.03	0.81	0.03	0.00	0.00	0.00
point	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.82	0.03	0.00	0.00
pick up	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.93	0.01	0.00
throw (over head)	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.07	0.85	0.00
throw (from bottom up)	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.05	0.69

Fig. 8. Three performance measures are measured as the value of λ increases: (a) the action assignment accuracy; (b) the compression rate and (c) the computation time. The best accuracy 80.5% is achieved at the block size of 10 × 10 pixels and $\lambda = 1.3$ in which the corresponding compression rate is 16.5% and computation time is 1.57s. Here, it is noted that accuracy is 80.2% at the block size of 5 × 5 and $\lambda = 1.3$ which is almost best, however, the corresponding computation time is larger, that is, 3.7 s. In (d), we show a confusion matrix of action assignment at frame level to the best average accuracy in (a) on the whole dataset.

Table 2
Comparison of accuracy (%) with *state-of-the-art* approaches. The bold value is for emerging the best performance among different work.

Method	Level	Actions	Actors	Cam.0	Cam.1	Cam.2	Cam.3	Cam.4	Average
(a) Single-view based action recognition									
Weinland et al. (2006)	Pre-segmented	11	10	–	–	–	–	–	**93.3**
	Action sequence	11	10	–	–	–	–	–	82.3
Lv and Nevatia (2007)	Action sequence	15	10	81.5	82.1	80.1	81.3	78.4	80.6
Yan et al. (2008)	Pre-segmented	13	12	72	53	68	63	–	64.0
Junejo et al. (2008)	Pre-segmented	11	10	76.4	77.6	73.6	68.8	66.1	72.5
Liu and Shah (2008)	Pre-segmented	13	12	76.7	73.3	72.0	73.0	-	73.8
Weinland et al. (2010)	Pre-segmented	11	10	85.8	86.4	88.0	88.2	74.7	84.6
Junejo et al. (2011)	Pre-segmented	11	10	80.0	83.9	80.5	85.5	73.3	80.6
Ramadan and Davis (2011)	Pre-segmented	13	11	85.7	87.3	82.4	86.8	–	85.6
Ours	Action sequence	15	12	77.6	79.3	78.8	83.1	83.6	80.5
Method	Level	Actions	Actors						Average
(b) Multiple-view based action recognition									
Weinland et al. (2007)	Pre-segmented	11	10						81.3
Yan et al. (2008)	Pre-segmented	13	12						78
Lewandowski et al. (2010)	Pre-segmented	12	12						83.1
Iosifidis et al. (2011)	Pre-segmented	11	10						**83.5**
Ours	Action sequence	15	12						81.0

to other actions (e.g., **check-watch** is wrongly assigned to **nothing** at 17%). This is not surprising because the transitional/boundary frames between two successive actions are often similar to each other, e.g., the ending frames of **check-watch** and the frames of **nothing** are naturally similar.

Here, it can be shown theoretically that *no selection* occurs for $\lambda = 1$, as stated in Section 3.3. In this case, two frames in one pair are neighboring in the original video context and thus they are so similar or even considered being duplicated. Using such pairs, the accuracy is low (the accuracy is lower than 65% for all tested block sizes in Fig. 8(a)). This means that our pairwise-frame representation using characteristic frames outperformed the single frame representation.

6.4.2. Comparison

We have compared the proposed framework with eleven *state-of-the-art* approaches. Eight of them are single-view based algorithms and four of them are multiple-view based algorithms. The results are shown in Table 2. The accuracy values were directly translated from the corresponding references.

Let us first evaluate the proposed algorithm in the group of single-view based algorithms. The proposed algorithm is comparable in accuracy to many competitors, but is not the best. Seven of them (Weinland et al., 2006, 2010; Yan et al., 2008; Junejo et al., 2008, 2011; Liu and Shah, 2008; Ramadan and Davis, 2011), however, assume a pre-segmentation by which a long term video sequence has already been segmented into a series of single action clips. This kind of pre-segmentation process usually needs a heavy human work-load, which reduces the practical value of those algorithms. Only three algorithms including ours (i.e., the algorithms of Weinland et al. (2006) (*action sequence*) and Lv and Nevatia (2007), and ours) do not require such a pre-segmentation of successive actions. All these three algorithms show a similar degree of performance. Another concern is the size of experiment. Their sizes are smaller than ours: the algorithm in (Weinland et al., 2006) was tested on 11 actions performed by 10 actors; Lv and Nevatia (2007) tested their algorithm only with 50 video sequences chosen from the overall 180 sequences in IXMAS database, while all 180 sequences containing 15 actions by 12 actors were tested in ours. In this sense, our accuracy is more reliable than theirs.

Next, let us compare the proposed algorithm with multiple-view based algorithms. Although our basic layout is to use a single camera because of the simplicity of facilities and the ease of application, but multiple-view can be utilized even in our framework. We adopted a simple *concatenating* strategy to represent one frame by combining the block-based features (Section 3.1) extracted from the five simultaneously observed images of this frame, provided in the IXMAS database. The results are also shown in Table 2. In spite of the fact that all four competitors assume pre-segmentation, their performances are comparable to ours. The usage of multiple views helped the proposed algorithm only by 0.5. This slight amount of improvement implies that the employed *concatenating* strategy could not extract sufficient information from multiple views.

The benefit of our algorithm is the improved processing speed as a whole. The proposed algorithm uses a smaller number of selected frames, thus it could be faster than any algorithm using entire frames. In addition, the proposed algorithm does not require an extra process for pre-segmentation. We could compare the time of the proposed algorithm with only that of the algorithm in (Lv and Nevatia, 2007). The result is, the average computation time of ours is 1.57 s per video, while theirs is proximately 226 s per video that is obtained by calculation from their reported speed (5.1 frames/s on a PC of CPU P4-3 GHz).

7. Discussion

In the proposed framework for temporal segmentation of successive actions, there is a little difficulty on how to determine the martingale threshold λ for selecting characteristic frames. As seen in Figs. 6 and 8, it is difficult to choose the universally effective value of λ. From Fig. 8(a) and (b), we recommend the Martingale value of $1.25 \leqslant \lambda \leqslant 1.35$ and block size B of 10×10 pixels for practical usage, considering the possibly existing noise in a video sequence.

Many approaches proposed so far need a relatively higher computation cost compared with the proposed framework. Unfortunately we cannot compare the actual computation time, because the faithful implementation of those algorithms and fair execution under the same condition are almost impossible. However, it is clear that only our approach selects a smaller number of frames from the entire video sequence (16.5% on the average). Therefore, it is almost sure that our method is faster than any of them. At least, we could speed up all of them by feeding the selected frames to them instead of the entire frame sequence/series.

8. Conclusion and future work

In this study, we have proposed a novel framework for temporal segmentation of successive actions, as summarized as follows: (1) Given a long-term video sequence, a smaller number of characteristic frames are selected firstly by a change detection algorithm using a Martingale nature, (2) pairwise-frame representation of consecutive characteristic frames is then employed to calculate the likelihood to trained actions models that are constructed for individual actions and transitive actions, and (3) final segmentation is obtained by solving an optimization problem on the basis of frame-based likelihood. The pairwise-frame representation gives a time differentiated information in two neighboring characteristic frames. Therefore, the similar frames appearing in different actions can be discriminated. The selection of characteristic frames brought a high efficiency, and the pair-wise treatment of characteristic frames brought a high performance of accurate segmentation and recognition of human actions.

In the future work, we will evaluate the proposed framework on (1) unconstrained video (e.g., Hollywood dataset (Marszalek et al., 2009)) and (2) a long-term video sequence in realistic life (e.g., Ger'Home datasets (Zouba et al., 2008)).

References

Ahmad, M., Lee, S.W., 2008. Human action recognition using shape and CLG-motion flow from multi-view image sequences. Pattern Recognit. 41 (7), 2237–2252.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R., 2005. Actions as space–time shapes. In: Proc. ICCV, vol.2, pp. 1395–1402.

Boykin, S., Merlino, A., 2000. Machine learning of event segmentation for news on demand. Comm. ACM 43 (2), 35–41.

Cutler, R., Davis, L.S., 2000. Robust real-time periodic motion detection, analysis, and applications. Pattern Anal. Machine Intell. 22 (8), 781–796.

Fathi, A., Mori, G., 2008. Action recognition by learning mid-level motion features. In: Proc. CVPR, pp. 1–8.

Harchaoui, Z., Bach, F., Moulines, E., 2009. Kernel change point analysis. In: Neural Inf. Process. Syst.

Minh Hoai, Zhen-Zhong Lan, De la Torre F., 2011. Joint segmentation and classification of human actions in video. In: Proc. CVPR, pp. 3265–3272.

Ho, S.S., Wechsler, H., 2010. A martingale framework for detecting changes in data streams by testing exchangeability. Pattern Anal. Machine Intell. 32 (12), 2113–2127.

Iosifidis, A., Tefas, A., Nikolaidis, N., Pitas, I., 2011. Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis. Comput. Vision Image Understanding, Special issue on Semant. Understanding Human Behaviors Image Sequences 116 (3), 347–360.

Jia, K., Yeung, D.Y., 2008. Human action recognition using local spatio–temporal discriminant embedding. In: Proc. CVPR, pp. 1–8.

Junejo, I., Dexter, E., Laptev, I., Perez, P., 2008. Cross-view action recognition from temporal self-similarities. In: Proc. ECCV, v2, pp. 293–306.

Junejo, I., Dexter, E., Laptev, I., Perez, P., 2011. View-independent action recognition from temporal self-similarities. Pattern Anal. Machine Intell. 33 (1), 172–185.

Kovashka, A., Grauman, K. 2011. Learning a hierarchy of discriminative space–time neighborhood features for human action recognition. In: Proc. CVPR, pp. 2046–2053.

Laptev, I., Belongie, S.J., Perez, P., Wills, J., 2005. Periodic motion detection and segmentation via approximate sequence alignment. In: Proc.ICCV, vol.1, pp. 816–823.

Lewandowski, M., Makris, D., Nebel, J.C., 2010. View and style-independent action manifolds for human activity recognition. In: Proc. ECCV, pp. 547–560.

Liu, J., Shah, M., 2008. Learning human actions via information maximization. In: Proc. CVPR, pp. 1–8.

Loui, A.C., Savakis, A.E., 2000. Automatic image event segmentation and quality screening for albumin applications. In: Proc. 2000 IEEE Internat. Conf. on Multimedia and Expo, 2000(ICME 2000), vol. 2, pp. 1125–1128.

Lu, G.L., Kudo, M., Toyama, J., 2012. Selection of characteristic frames in video for efficient action recognition. IEICE Transactions on Information and Systems E95-D(10), 2514–2521.

Lv, F., Nevatia, R., 2007. Single view human action recognition using key pose matching and viterbi path searching. In: Proc.CVPR, pp. 1–8.

Marszalek M., Laptev I., Schmid C., 2009. Actions in context. In: Proc. CVPR, pp. 2929–2936.

Ogale, A., Karapurkar, A., Aloimonos, Y., 2007. View-invariant modeling and recognition of human actions using grammars. Dyn. Vis., 115–126.

Poppe, R., 2010. A survey on vision-based human action recognition. Image Vision Comput. 28 (6), 976–990.

Ramadan, S., Davis, L., 2011. Action recognition using partial least squares and support vector machines. In: Proc. 18th IEEE Int. Conf. Image Processing (ICIP), pp. 533–536.

Schindler K., Van-Gool, L., 2008. Action snippets: how many frames does human action recognition require? In: Proc. CVPR, pp. 1–8.

Schuldt, C., Laptev, I., Caputo, B., 2004. Recognizing human actions: a local SVM approach. In: Proc. ICPR, vol.3, pp. 32–36.

Shakhnarovich, G., Viola, P., Darrell, T., 2003. Fast pose estimation with parameter-sensitive hashing. In: Proc. ICCV, pp. 750–757.

Thurau, C., Hlavác, V., 2008. Pose primitive based human action recognition in videos or still images. In: Proc. CVPR, pp. 1–8.

Vovk V., Nouretdinov I., Gammerman A., 2003. Testing exchangeability on-line. In: Proc. 20th Intnat. Conf. on Machine Learning, pp. 768–775.

Wang, L., Suter, D., 2007. Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model. In: Proc.CVPR, pp. 1–8.

Wang, L., Hu, W., Tan, T., 2003. Recent developments in human motion analysis. Pattern Recognit. 36 (3), 585–601.

Weinland, D., Boyer, E., 2008. Action recognition using exemplar-based embedding. In: Proc. CVPR, pp. 1–7.

Weinland, D., Ronfard, R., Boyer, E., 2006. Free viewpoint action recognition using motion history volumes. Comput. Vision Image Understanding 104 (2), 249–257.

Weinland, D., Ronfard, R., Boyer, E., 2006. Free viewpoint action recognition using motion history volumes. Comput. Vision Image Understanding 104 (2–3), 249–257.

Weinland, D., Boyer, E., Ronfard, R., 2007. Action recognition from arbitrary views using 3D exemplars. In: Proc. ICCV, pp. 1–7.

Weinland, D., Özuysal, M., Fua, P., 2010. Making action recognition robust to occlusions and viewpoint changes. In: Proc. ECCV, pp. 635–648.

Xuan X., Murphy, K., 2007. Modeling changing dependency structure in multivariate time series. In: Proc. ICML, pp. 1055–1062.

Yan, P., Khan, S.M., Shah, M., 2008. Learning 4D action feature models for arbitrary view action recognition. In: Proc. CVPR, pp. 1–7.

Zelnik-Manor, L., Irani, M., 2006. Statistical analysis of dynamic actions. Pattern Anal. Machine Intell. 28 (9), 1530–1535.

Zouba, N., Boulay, B., Bremond, F., Thonnat, M., 2008. Monitoring activities of daily living (adls) of elderly based on 3d key human postures. In: Proc. ICVW'08, pp. 37–50.