# Joint Segmentation and Classification of Human Actions in Video

Minh Hoai Zhen-Zhong Lan Fernando De la Torre
Carnegie Mellon University, Pittsburgh, PA 15213
minhhoai@cmu.edu, lansysu@gmail.com, ftorre@cs.cmu.edu

## Abstract

*Automatic video segmentation and action recognition has been a long-standing problem in computer vision. Much work in the literature treats video segmentation and action recognition as two independent problems; while segmentation is often done without a temporal model of the activity, action recognition is usually performed on pre-segmented clips. In this paper we propose a novel method that avoids the limitations of the above approaches by jointly performing video segmentation and action recognition. Unlike standard approaches based on extensions of dynamic Bayesian networks, our method is based on a discriminative temporal extension of the spatial bag-of-words model that has been very popular in object recognition. The classification is performed robustly within a multi-class SVM framework whereas the inference over the segments is done efficiently with dynamic programming. Experimental results on honeybee, Weizmann, and Hollywood datasets illustrate the benefits of our approach compared to state-of-the-art methods.*

## 1. Introduction

The amount of video being captured with video cameras is growing exponentially, and there is a need to develop efficient algorithms for content extraction. Understanding human activities in video plays a key role in many applications such as camera surveillance, video summarization, highlight extraction, and content-based annotation. However, understanding human activities in video is a challenging problem due to the large variability in the temporal scale and periodicity of human actions, the complexity of articulated motion, the exponential nature of all possible movement combinations, and the prevalence of irrelevant background.

Action recognition systems aim at recognizing the classes of the actions present in a video, independently of the background. Much effort in the literature for action recognition attempted to build robustness to background clutter by using temporal segmentation as preprocessing,

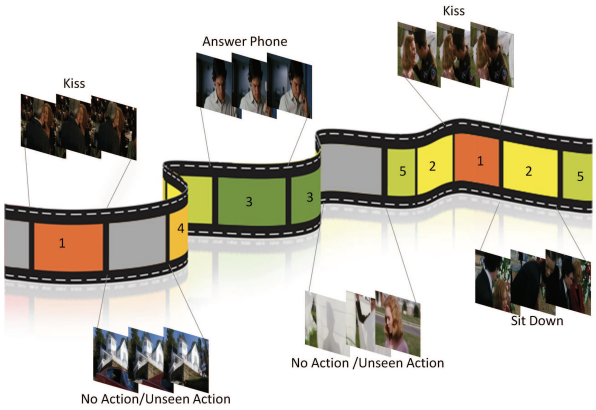


Figure 1. Realistic video contains multiple actions and segmentation is not provided a priori. We propose a discriminative framework for joint segmentation and action recognition in a video sequence. Our method can handle multiple action classes, including the null class of unfamiliar activities.

e.g., [12, 16, 20]. It was hoped that segmentation methods could partition videos into coherent constituent parts, and recognition could then be simply carried out as categorization of the action classes corresponding to the segments. This naive strategy to categorization floundered on the challenges presented by bottom-up temporal segmentation, due to the difficulty of partitioning a video into actions purely based on low-level cues. More importantly, the segmentation process is done independently of the classification, that might result in important loss of information related to the actions.

The difficulty of bottom-up temporal segmentation has been well understood, especially among the speech and Natural Language Processing (NLP) communities. They addressed this issue by proposing various methods where bottom-up segmentation was assisted by concurrent top-down recognition. Most of these methods are based on variants of dynamic Bayesian networks that model the dynamics of the temporal events as transitions in a partially observed state space, e.g., [18, 8]. Although these approaches have achieved high performance in speech and NLP domains, accuracy tends to be much lower in studies for ac-

tion recognition. Action recognition is challenging; temporal hidden state models suffer the drawbacks of needing either an explicit definition of the latent states of all frames, or the need to simultaneously learn a state sequence and a state transition model that fit the data, resulting in a high-dimensional minimization problem with typically many local minima. Furthermore, processing long video sequences requires a null-class model for the background clutter which is often problematic for generative approaches.

Action recognition can be treated as temporal event detection, without the need for explicit segmentation of the whole video sequence. Event detection algorithms operate by evaluating a classifier function at many different segments of the video and then predicting the event presence in segments with high-score. This methodology has been applied with great success to a wide variety of temporal event classes, e.g., [14, 17, 7]. This approach, however, has fundamental drawbacks for action recognition as classes are treated independently and detected actions can potentially overlap each other.

In this paper we propose a novel learning framework that simultaneously performs temporal segmentation and event recognition in time series and its applications to action recognition in video. Our discriminative recognition model is trained using labeled data with a multi-class SVM [4] that maximizes the separating margin between classes. Once the model for all actions has been learned, simultaneous segmentation and recognition is done efficiently using dynamic programming, maximizing the SVM score of the winning class while suppressing those of the non-maximum classes.

Figure 1 illustrates the main idea of our paper. During training, a model for human actions is learned from a set of labeled training samples. Given a testing video with a continuous stream of human activities, our algorithm finds the globally optimal temporal segmentation (i.e., the change points between actions) and class labels. Notably, there exist other supervised learning techniques for joint segmentation and recognition like ours; however, the segmentation inference is often done heuristically without any optimality guarantee, e.g., searching for one temporal scale and thresholding the outputs. Furthermore, many of these methods are based on generative models such as extensions of Hidden Markov Models. A major issue of generative approaches is their limited ability to model the *null* class (no action or unseen actions) due to the large variability of the null class. Our proposed method is based on multi-class SVM [4], a discriminative model that does not suffer from this limitation of generative models.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the new temporal model for joint segmentation and recognition. Experiments on three standard datasets are reported in Section 4. Section 5 concludes and outlines a direction for future work.

## 2. Related work

Much work in the literature on action recognition acts on short video clips, assuming temporal segmentation has been done a priori. Laptev et al. [12] used bag of spatio-temporal interest points to classify realistic human motion from Hollywood movies. Ali & Shah [1] recognized human actions using kinematic features. Dollár et al. [6] proposed cuboid descriptors for categorizing short video clips of animal behavior. Tran & Sorokin [23] used motion context and a learned nearest neighbor metric for classifying actions in YouTube videos of sporting events. Satkin & Herbert [20] observed the imperfection of human-provided segmentation and proposed to crop the video sequences to boost the recognition performance. For a more complete survey, we refer the reader to [24].

Recent effort in recognizing actions from longer video sequences focuses on the event detection approach. Event detection algorithms operate by evaluating a classifier function at many different segments of the video and then predicting the event presence in segments with high-score. Ke et al. [11] detected human actions in crowded video. Nguyen et al. [14] used structured-output SVM to localize the occurrences of facial action units. Duchenne et al. [7] and Nguyen et al. [15] used multiple instance learning to annotate weakly labeled video sequences. Though this methodology of treating action recognition as event detection has been shown to be effective in many cases, it has a major limitation. This approach often treats action classes independently, yielding a separate temporal event detector for every single class. As a result, knowledge about the presence or absence of a particular action do not constrain on those of any other action, and the temporal extents of detected actions can potentially overlap each other.

The literature on segmentation of time series falls in several categories. Change point detection such as [25] and [10] is a popular technique; it works by performing a sequence of change-point analysis in a sliding window along the time dimension. This, unlike our proposed method, only detects local boundaries and does not provide a global model for temporal events. Moreover, it is unclear how this unsupervised approach can cope with the problem of over-segmentation, especially for complex actions which often contain many changes in local motion statistics. Time series segmentation has also been implicitly studied from the perspective of analyzing periodicity of cyclic events, e.g., [5, 19]. Cyclic motion analysis, however, is only interested in extracting segments of repetitive motion; consequently a substantial portion of a signal might not either be segmented or modeled. Segmentation can be done by clustering frames and grouping those that are assigned to the same cluster to form a segment, as in [26]. This approach performs segmentation as a subsequent step of clustering; it lacks a mechanism to incorporate the dynamics of temporal

events in the clustering process.

Existing techniques for joint segmentation and recognition are often based on state-space or generative models. Oh et al. [18] proposed parametric segmental switching linear dynamical system to model honeybee behavior. Fox et al. [8] used HMM with Hierarchical Dirichlet prior. Niebles et al. [17] used probabilistic Latent Semantic Analysis for unsupervised learning of human actions. Brand and Kettnaker [3] trained HMM with entropy minimization and interpreted the hidden states for discovery and segmentation of activities in video. Sminchisescu et al. [22] proposed to use conditional models for human action recognition. Laxton et al. [13] designed a hierarchical structure based on dynamic Bayesian network to decompose complex activities. Zhou et al. [27] combined kernel $k$-means, a generative model, with dynamic time warping for segmenting human motion. In contrast to the aforementioned approaches for joint segmentation and recognition, we propose a discriminative framework that is based on multi-class SVM.

## 3. Joint segmentation and recognition

This section describes our framework for joint video segmentation and action recognition. Our discriminative recognition model is trained using multi-class SVM, and segmentation is done using dynamic programming.

### 3.1. Supervised training with multi-class SVM

Given a collection of training time series $\mathbf{X}^1, \cdots, \mathbf{X}^n$ with known segmentation and class labels, i.e., the change points between actions $0 = s_1^i < \cdots < s_{k_i+1}^i = len(\mathbf{X}^i)$ and the associated class labels $y_1^i, \cdots, y_{k_i}^i \in \{1, \cdots, m\}$ are provided (see Figure 2 for graphical illustration), we can use multi-class SVM [4] to train a model for temporal actions:

$$\underset{\mathbf{w}_j, \xi_t^i \geq 0}{\text{minimize}} \frac{1}{2m} \sum_{j=1}^{m} ||\mathbf{w}_j||^2 + C \sum_{i=1}^{n} \sum_{t=1}^{k_i} \xi_t^i, \text{ s.t.} \quad (1)$$

$$(\mathbf{w}_{y_t^i} - \mathbf{w}_y)^T \varphi(\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}) \geq 1 - \xi_t^i \; \forall i, t, y \neq y_t^i. \quad (2)$$

Here $\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}$ denotes the segment of time series $\mathbf{X}^i$, taken from frame $s_t + 1$ to frame $s_{t+1}$ inclusive. $len(\mathbf{X}^i)$ denotes the length of time series $\mathbf{X}^i$, $\varphi(\cdot)$ is the feature computation function, and $\mathbf{w}_y^T \varphi(\mathbf{X}^i_{(s_t^i, s_{t+1}^i]})$ is the *SVM score* for assigning segment $\mathbf{X}^i_{(s_t, s_{t+1}]}$ to class $y$. Constraint (2) requires segment $\mathbf{X}^i_{(s_t^i, s_{t+1}^i]}$ to belong to class $y_t^i$ with high confidence; in other words, the SVM score for class $y_t^i$ should be relatively higher than that of any other class by a large margin. $\{\xi_t^i\}$ are slack variables which allow for penalized constraint violation. $C$ is the parameter controlling the trade-off between a large margin and less constrained violation.

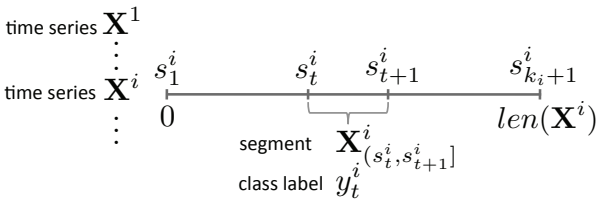


Figure 2. Training data: the change points between actions $s_1^i, \cdots, s_{k_i+1}^i$ and the class labels $y_t^i$ are provided.

### 3.2. Segmentation with non-maxima suppression

Once the weight vectors $\{\mathbf{w}_j\}$ have been learned, we can use them to segment unseen time series $\mathbf{X}$ by finding a set of change points $s_1, \cdots, s_{k+1}$ that:

$$\underset{k, s_t, y_t, \xi_t \geq 0}{\text{minimize}} \sum_{t=1}^{k} \xi_t, \text{ s.t.} \quad (3)$$

$$l_{min} \leq s_{t+1} - s_t \leq l_{max} \; \forall t, \; s_1 = 0, s_{k+1} = len(\mathbf{X}),$$

$$(\mathbf{w}_{y_t} - \mathbf{w}_y)^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}) \geq 1 - \xi_t \; \forall t, y \neq y_t,$$

Observe that the number of segments $k$ is not known in advance and, therefore, needs to be optimized over. In the above formulation, $l_{min}$ and $l_{max}$ are the minimum and maximum lengths of segments, which can be inferred from training data. $\mathbf{w}_y^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]})$ is the SVM score for assigning segment $\mathbf{X}_{(s_t, s_{t+1}]}$ to class $y$. What we propose is to maximize the difference between the SVM score of the winning class $y_t$ and that of any other class $y \neq y_t$, filtering through the Hinge loss. The idea is to seek a segmentation in which each resulting segment is assigned a class label with high confidence. This is very different from what is often done in the literature, e.g., [21], that maximizes the total SVM scores:

$$\underset{k, s_t, y_t}{\text{maximize}} \sum_{t=1}^{k} \mathbf{w}_{y_t}^T \varphi(\mathbf{X}_{(s_t, s_{t+1}]}), \text{ s.t.} \quad (4)$$

$$l_{min} \leq s_{t+1} - s_t \leq l_{max} \; \forall t, \; s_1 = 0, s_{k+1} = len(\mathbf{X}),$$

Different from the above formulation, our segmentation criterion, Eq. (3), requires suppressing the non-maximum classes. To see the difference between these two criteria, consider breaking a time series $AB$ in Figure 3 at either $M$ or $N$. For simplicity, suppose there are only two classes, and the SVM scores of the first and second class for some segments in Figure 3 are in printed in underlined red and overlined blue, respectively. The segmentation criterion of Eq. (4) would prefer to divide $AB$ at $M$ because it leads to higher total SVM scores of the winning classes (total score

of 3.5 = 2.0 + 1.5, 2.0 from segment $AM$ and 1.5 from $MB$). On the other hand, our segmentation criterion does not prefer to cut at $M$ because it cannot confidently classify the resulting segments. To see this, consider the segment $AM$, even though the SVM score of the winning class, class 1, is high, the SVM score of the alternative, class 2, is also similarly high. Our proposed criterion seeks the optimal segmentation that maximizes the difference between the SVM scores of the winning class and the next best alternative, filtering through the robust Hinge loss.

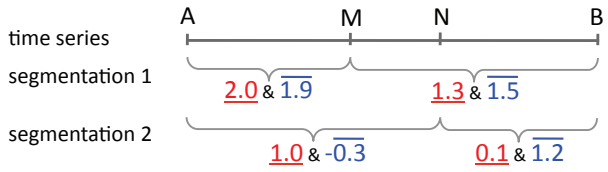


Figure 3. Which segmentation is preferred, breaking time series $AB$ at $M$ or $N$? Suppose there are only two classes, SVM scores of the first and second class for corresponding segments are printed in red and blue, respectively. Our segmentation criterion prefers to cut at $N$ because the resulting segments can be confidently classified. This figure is best seen in color.

In theory, our segmentation criterion is preferred because it optimizes the same objective as that of the training formulation in Eq. (1). In Section 4, we will show empirically the benefits of our approach.

### 3.3. Dynamic programming for segmentation

Segmentation of an unseen time series given the recognition model, $\{\mathbf{w}_i\}$, can be done using dynamic programming. To use dynamic programming for time series $\mathbf{X}$, let us consider the best segmentation for the truncated time series $\mathbf{X}_{(0,u]}$ (ignoring frames from $u+1$ onward), i.e.,

$$f(u) = \min_{k,s_t,y_t,\xi_t \geq 0} \sum_{t=1}^{k} \xi_t, \tag{5}$$
$$\text{s.t. } l_{min} \leq s_{t+1} - s_t \leq l_{max},\ s_1 = 0, s_{k+1} = u\ \forall t,$$
$$(\mathbf{w}_{y_t} - \mathbf{w}_y)^T \varphi(\mathbf{X}_{(s_t,s_{t+1}]}) \geq 1 - \xi_t\ \forall t, y \neq y_t.$$

For every tuple $u \in (0, len(\mathbf{X})], l \in [l_{min}, l_{max}]$ let:

$$\xi(u,l) = \max\{0, 1 - (\mathbf{w}_{\hat{y}} - \mathbf{w}_{\tilde{y}})^T \varphi(\mathbf{X}_{(u-l,u]})\}, \tag{6}$$
$$\text{where} \quad \hat{y} = \operatorname*{argmax}_y \mathbf{w}_y^T \varphi(\mathbf{X}_{(u-l,u]}), \text{ and} \tag{7}$$
$$\tilde{y} = \operatorname*{argmax}_{y \neq \hat{y}} \mathbf{w}_y^T \varphi(\mathbf{X}_{(u-l,u]}). \tag{8}$$

The goal is to compute $f(len(\mathbf{X}))$, which can be done with dynamic programing using the formula:

$$f(u) = \operatorname*{argmin}_{l_{min} \leq l \leq l_{max}} \{\xi(u,l) + f(u-l)\}. \tag{9}$$

The complexity of dynamic programming for segmenting time series $\mathbf{X}$ is $O(m(l_{max} - l_{min} + 1)len(\mathbf{X}))$.

### 3.4. Segment-level feature mapping

This section describes the form of the feature mapping $\varphi(\cdot)$. Following [2] and inspired by HMMs, we propose to use two types of features, interactions between the observation vectors and the set of predefined states as well as the transition between states of neighboring frames along the subsegment. More formally, we consider an additive feature mapping:

$$\varphi(\mathbf{X}_{(s_t,s_{t+1}]}) = \sum_{j=s_t+1}^{s_{t+1}} \begin{bmatrix} \boldsymbol{\phi}^{obs}(\mathbf{x}_j) \\ \boldsymbol{\phi}^{int}(\mathbf{x}_j) \end{bmatrix} \tag{10}$$

Here $\mathbf{x}_j$ denotes the frame $j$ of time series $\mathbf{X}$, $\boldsymbol{\phi}^{obs}(\mathbf{x}_j)$ and $\boldsymbol{\phi}^{int}(\mathbf{x}_j)$ are the observation and interaction feature vectors, respectively. These feature vectors are computed as follows. First we build a dictionary of temporal words by clustering the raw feature vectors from the time series in the dataset. Let $\mathbf{c}_1, \cdots, \mathbf{c}_r$ denote the set of clustering centroids. We consider $\boldsymbol{\phi}^{obs}(\mathbf{x}_j)$ as a $r \times 1$ vector with the $i^{th}$ entry is $\phi_i^{obs}(\mathbf{x}_j) = \mu \exp(-\gamma||\mathbf{x}_j - \mathbf{c}_i||^2)$. Intuitively, the $i^{th}$ entry of observation vector is the pseudo-probability that $\mathbf{x}_j$ belongs to state $i$, which is proportional to how close $\mathbf{x}_j$ to the cluster centroid $\mathbf{c}_i$. The scale factor $\mu$ is chosen such that the sum of the entries of $\boldsymbol{\phi}^{obs}(\mathbf{x}_j)$ is one. The interaction feature vector $\boldsymbol{\phi}^{int}(\mathbf{x}_j)$ is defined as a $r^2 \times 1$ vector, with:

$$\phi^{int}_{(u-1)r+v}(\mathbf{x}_j) = \phi^{obs}_u(\mathbf{x}_j)\phi^{obs}_v(\mathbf{x}_{j-1})\ \forall u, v = 1, \cdots, r.$$

The above quantity is the pseudo-probability for transitioning from state $v$ to state $u$ at time $j$. The interaction feature vector depends on both the observation vectors of the frame $\mathbf{x}_j$ and the preceding frame $\mathbf{x}_{j-1}$.

## 4. Experiments

This section describes experimental results on three standard datasets: honeybee dancing [18], Weizmann [9], and Hollywood [12]. In all experiments we measured the joint segmentation-recognition performance as follows. We ran our algorithm on long video sequences to find the optimal segmentation and class labels. At that point, each frame was associated with a particular class, and the overall frame-level accuracy against the ground truth labels was calculated as the ratio between the number of agreements over the total number of frames. This evaluation criterion is different from recognition accuracy of algorithms that require pre-segmented video clips. As a consequence, our results here are not directly comparable to some published numbers in the literature [12, 9, 20]. However, where available, we included the previously reported results for reference.

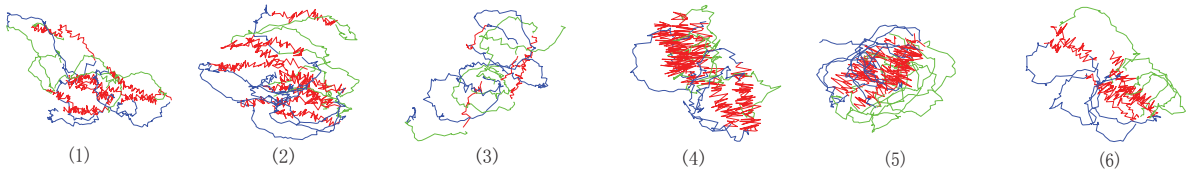


Figure 4. Honeybee dataset—trajectories of dancing bees. Each dance trajectory is the output of a vision-based tracker. The segments are color coded; red, green, and blue correspond to waggle, right-turn, and left-turn, respectively. This figure is best seen in color.

### 4.1. Honeybee dataset

The honeybee dataset [18] contains video sequences of honeybees which communicate the location and distance to a food source through a dance that takes place within the hive. The dance can be decomposed into three different movement patterns: waggle, right-turn, and left-turn. During the waggle dance, the bee moves roughly in a straight line while rapidly shaking its body from left to right; the duration and orientation of this phase correspond to the distance and the orientation to the food source. At the endpoint of a waggle dance, the bee turns in a clockwise or counter-clockwise direction to form a turning dance. These turning dances often shape like a capital $C$. The dataset consists of six video sequences with lengths 1058, 1125, 1054, 757, 609, and 814 frames, respectively.

The bees were visually tracked (Figure 5.a), and their locations and head angles were recorded. The 2D trajectories of the bees in six sequences are shown in Fig. 4. The frame-level feature vector was $[x, y, \sin(\theta), \cos(\theta)]$, where $(x, y)$ was the 2D location of the bee and $\theta$ was the bee's head angle. Once the sequence observations were obtained, the trajectories were preprocessed as in [8]. Specifically, the trajectory sequences were rotated so that the waggle dances had head angle measurements centered about zero radian. The sequences were then translated to center at $(0, 0)$, and the 2D coordinates were scaled to the $[-1, 1]$ range. Aligning the waggle dances was possible by looking at the high frequency portions of the head angle measurements. Following the suggestion of [18], the data was smoothed using Gaussian FIR pulse-shaping filter with 0.5dB bandwidth-symbol time. Figure 5.b shows the correlation between the feature vectors and the labels. Since the lengths of original waggle, right-turn, and left-turn sequences are quite long, we further broke them down into smaller subsequences (maximum length 13) to increase the number of training instances.

Following [18, 8], we adopted the leave-one-out evaluation strategy: training on five sequences and testing on the left-out sequence. Table 1 displays the experimental results of our method along with three state-of-the-art methods. SLDS and PS-SLDS [18] are switching linear dynamical system and parametric segmental switching linear dynamical system, respectively. HDP-HMM [8] is the method combining hierarchical Dirichlet process prior and HMM.

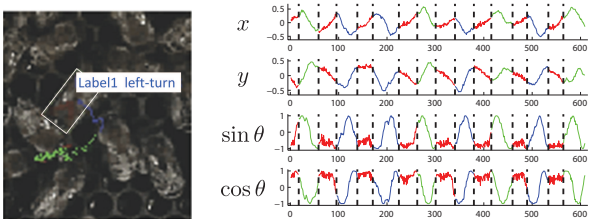


Figure 5. a) Visual tracking: green + blue trajectory and the bounding box for tracking. b) plots of the frame-level features $[x, y, \sin(\theta), \cos(\theta)]$. Red, green, and blue correspond to waggle, right-turn, and left-turn, respectively. This is best seen in color.

Although all methods are supervised learning, the setting of HDP-HMM is slightly different from those of the others. HDP-HMM requires knowing the testing sequences (without labels) at training time. We also implemented MaxScoreSeg(c.f., [21]), a variant of our proposed algorithm, that performed temporal segmentation by maximizing the total SVM scores (Eq. 4) instead of maximizing the assignment confidence (Eq. 3). The reported numbers in Table 1 are frame-level accuracy (%) measuring the joint segmentation-recognition performance as described at the beginning of Section 4. As can be seen, our method achieved similar or better results than state-of-the-art methods on all sequences, and it had the best overall performance. Figure 6 displays side-by-side comparison of the prediction result and the human-labeled ground truth.

| Sequence | 1 | 2 | 3 | 4 | 5 | 6 | Mean |
|---|---|---|---|---|---|---|---|
| SLDS [18] | 74.0 | 86.1 | 81.3 | **93.4** | 90.2 | 90.4 | 85.9 |
| PS-SLDS [18] | 75.9 | 92.4 | **83.1** | **93.4** | 90.4 | 91.0 | 87.7 |
| HDP-HMM [8] | 55.0 | 86.3 | 81.7 | 89.0 | **92.4** | 89.6 | 83.3 |
| MaxScoreSeg | 82.2 | 85.3 | 75.0 | 87.5 | 88.8 | 88.0 | 84.5 |
| Ours | **85.9** | **92.6** | 81.3 | 92.3 | 90.6 | **93.1** | **89.3** |

Table 1. Frame-level accuracy (%) on honeybee dataset. Our method achieved similar and sometimes better results than state-of-the-art methods [18, 8]. Averaged over all six sequences, our method yielded the best result.

### 4.2. Weizmann dataset

The Weizmann dataset contains 90 video sequences (180 × 144 pixels, deinterlaced 50fps) of 9

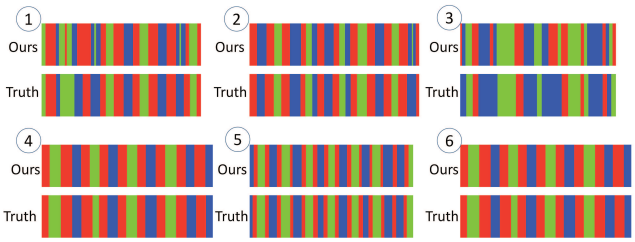


Figure 6. Automatic segmentation-recognition versus human-labeled ground truth. The segments are color coded; red, green, and blue correspond to waggle, right-turn, and left-turn, respectively. This figure is best seen in color.

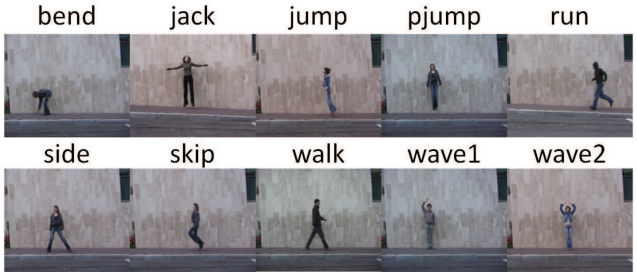


Figure 7. Typical frames from the Weizmann dataset.

people, each performing 10 actions: bend, jumping-jack (or shortly jack), jump-forward-on-two-legs (jump), jump-in-place-on-two-legs (pjump), run, gallop-sideways (side), skip, walk, wave-one-hand (wave1), and wave-two-hands (wave2). Figure 7 displays several typical frames extracted from the dataset. Each video sequence in this dataset only consists of a single action.

To evaluate the segmentation and recognition performance of our method, we performed experiments on longer video sequences which were created by concatenating existing single-action sequences. Specifically, we created 9 long sequences, each composed of 10 videos for 10 different actions (each original video samples was used only once). Following [9], we extracted binary masks (Figure 8.b) and computed Euclidean distance transform (Figure 8.c) for frame-level features. We built a dictionary of temporal words with 100 clusters using $k$-means. As in the experiment for honeybee dataset, we measured the leave-one-out segmentation and recognition performance. Table 2 shows the confusion matrix for segmentation and recognition of 10 actions. Our method yielded the average accuracy of 87.7%, aggregated over 9 sequences and 20 runs. Gorelick et al. [9] reported the recognition result of 97.8%. Unfortunately, their result and ours are not directly comparable. Their method required pre-segmented video sequences and only measured the recognition performance. The variant of our method, MaxScoreSeg [21], that performed temporal segmentation by maximizing the total SVM scores (Eq. 4) obtained the average accuracy of 69.7%. This relatively low accuracy is due to the mismatch


Figure 8. Frame-level features for Weizmann dataset. a) original frame, b) binary mask, c) Euclidean distance transform for frame-level features.

between the segmentation criterion and the training objective, as explained in Section 3.2.

| | bend | jack | jump | pjump | run | side | skip | walk | wave1 | wave2 |
|---|---|---|---|---|---|---|---|---|---|---|
| bend | .85 | .08 | .05 | .01 | .00 | .01 | .00 | .00 | .00 | .00 |
| jack | .00 | .93 | .00 | .00 | .04 | .00 | .01 | .00 | .01 | .01 |
| jump | .00 | .01 | .88 | .06 | .04 | .00 | .00 | .00 | .00 | .01 |
| pjump | .00 | .01 | .04 | .85 | .02 | .00 | .00 | .00 | .08 | .00 |
| run | .00 | .00 | .03 | .00 | .93 | .00 | .00 | .01 | .03 | .00 |
| side | .00 | .03 | .00 | .03 | .00 | .90 | .00 | .01 | .00 | .03 |
| skip | .00 | .00 | .02 | .00 | .05 | .00 | .77 | .03 | .00 | .13 |
| walk | .00 | .00 | .08 | .00 | .00 | .00 | .00 | .88 | .00 | .04 |
| wave1 | .00 | .00 | .00 | .00 | .01 | .00 | .03 | .00 | .93 | .03 |
| wave2 | .00 | .02 | .02 | .00 | .00 | .00 | .08 | .02 | .01 | .85 |

Table 2. Performance on Weizmann dataset, confusion matrix for segmentation and recognition of 10 different actions at frame level. The number at row R and column C is the proportion of R class which is classified as C class. For example, 3% of the **wave1** frames is misclassified as **wave2** class. The average accuracy is 87.7%.

To evaluate the performance of the proposed method in the presence of the null class, background clutter with large variability, we repeated the experiment considering the last five classes of actions (side, skip, walk, wave1, and wave2) as the null class. Table 3 shows the confusion matrix for five actions and the null class. Our method yielded the average accuracy of 93.3%, compared with 77.9% of MaxScoreSeg. Figure 9 displays side-by-side comparison of the prediction result and the human-labeled ground truth. Except for several cases, the majority of error occurs at the boundaries between actions. Error at the boundaries does not necessarily indicate the flaw of our method as human labels are often imperfect [20].

### 4.3. Hollywood dataset

Hollywood dataset contains video samples of human action from 32 movies. Each sample is labeled with one of eight action classes: AnswerPhone, HugPerson, Kiss, SitDown, SitUp, GetOutCar, HandShake, and StandUp. The

| | bend | jack | jump | pjump | run | Null |
|---|---|---|---|---|---|---|
| bend | .96 | .01 | .01 | .00 | .00 | .01 |
| jack | .00 | .97 | .00 | .01 | .00 | .02 |
| jump | .00 | .00 | .88 | .06 | .04 | .02 |
| pjump | .00 | .00 | .01 | .98 | .00 | .01 |
| run | .00 | .00 | .01 | .00 | .91 | .08 |
| Null | .01 | .03 | .00 | .03 | .03 | .90 |

Table 3. Weizmann dataset with the null class. Confusion matrix for segmentation and recognition of five different actions: bend, jack, jump, pjump, and run. The null class is the combination of all other classes. The average accuracy is 93.3%.

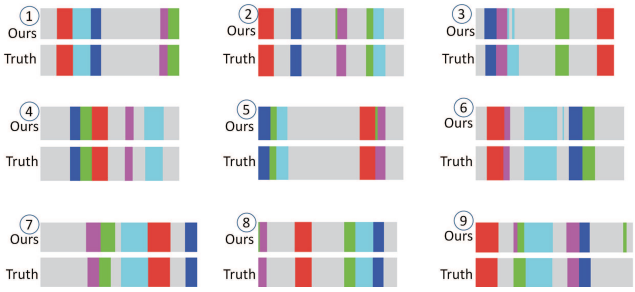


Figure 9. Automatic segmentation-recognition versus human-labeled ground truth. The segments are color coded; red, cyan, magenta, blue, green, and gray correspond to bend, jack, jump, pjump, run, and null classes, respectively. This figure is best seen in color.


Figure 10. Typical frames from the Hollywood dataset.

dataset is divided into two disjoint subsets; the training set contains video clips from 12 movies and the testing set contains the remaining clips. The total number of video samples in the training and testing sets are 219 and 211, respectively. Here we selected the first four classes as actions to be recognized, and the others were considered as parts of the null class.

Following [12], we detected space-time interest points and described them using histogram of oriented (spatial) gradients (HOG). Features belong to the same frame were combined together. A dictionary of temporal words with 100 clusters was constructed using $k$-means quantization. To evaluate the joint segmentation and recognition performance, we created 30 long testing sequences by concatenating eight randomly selected original video samples. The evaluation criterion was based on frame-level accuracy as described at the beginning of Section 4. Our method achieved the average accuracy of 42.24% (averaged over 30 sequences, repeated with 50 runs). As a reference, Laptev et al. [12] reported the average recognition result of 27% on this dataset with the same HOG features. Unfortunately, their result and ours are not directly comparable since their method required pre-segmented video sequences and only measured the recognition performance. Furthermore, the number of action classes in two experiments are different.

| | AP | HP | KS | SD | Null |
|---|---|---|---|---|---|
| AP | .35 | .14 | .13 | .22 | .16 |
| HP | .08 | .34 | .20 | .17 | .22 |
| KS | .08 | .10 | .51 | .11 | .21 |
| SD | .09 | .06 | .14 | .45 | .27 |
| Null | .11 | .07 | .17 | .19 | .47 |

Table 4. Hollywood dataset—confusion matrix for AnswerPhone (AP), HugPerson (HP), Kiss (KS), SitDown (SD), and the null class (all other actions). The average accuracy is 42.24%.

## 5. Conclusions

We proposed a novel approach for simultaneous temporal segmentation and action recognition from video. The recognition model was trained discriminatively using multi-class SVM, while segmentation inference was done efficiently with dynamic programming. This supervised framework provides a systematic and mathematically elegant algorithm for time series segmentation and action recognition. Experimental validation on standard datasets showed the competitiveness of our approach against state-of-the-art methods.

Though the proposed method yielded encouraging results on standard datasets, its requirement of fully labeled data for training inevitably limits its applicability to small training set with few actions. A possible direction for future work is to develop an unsupervised method for joint segmentation and modeling.

## Acknowledgements

This work was partially supported by the National Science Foundation under Grant CPS-0931999. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References


[1] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):288–303, 2010. 3266

[2] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. In *International Conference on Machine Learning*, 2003. 3268

[3] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, 2000. 3267

[4] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. 3266, 3267

[5] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, 2000. 3266

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCV Workshop on Visual Surveillance & Performance Evaluation of Tracking and Surveillance*, 2005. 3266

[7] O. Duchenne, I. Laptev, J. Sivic, F. R. Bach, and J. Ponce. Automatic annotation of human actions in video. In *International Conference on Computer Vision*, 2009. 3266

[8] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *Neural Information Processing Systems*, 2009. 3265, 3267, 3269

[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007. 3268, 3270

[10] Z. Harchaoui, F. Bach, and E. Moulines. Kernel change-point analysis. In *Neural Information Processing Systems*, 2009. 3266

[11] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *International Conference on Computer Vision*, 2007. 3266

[12] I. Laptev, M. Marsza, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition*, 2008. 3265, 3266, 3268, 3271

[13] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *Computer Vision and Pattern Recognition*, 2007. 3267

[14] M. H. Nguyen, T. Simon, F. De la Torre, and J. Cohn. Action unit detection with segment-based SVMs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 3266

[15] M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of International Conference on Computer Vision*, 2009. 3266

[16] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision*, 2010. 3265

[17] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008. 3266, 3267

[18] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *International Journal of Computer Vision*, 77(1–3):103–124, 2008. 3265, 3267, 3268, 3269

[19] E. Pogalin, A. Smeulders, and A. Thean. Visual quasi-periodicity. In *Computer Vision and Pattern Recognition*, 2008. 3266

[20] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *European Conference on Computer Vision*, 2010. 3265, 3266, 3268, 3270

[21] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. In *Computer Vision and Pattern Recognition*, 2008. 3267, 3269, 3270

[22] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In *International Conference on Computer Vision*, 2005. 3267

[23] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *European Conference on Computer Vision*, 2008. 3266

[24] P. Turaga, R. Chellappa, and V. Subrahmanian. Machine recognition of human activities: a survey. *Circuits and Sustems for Video Technology*, 18(11):1473–1488, 2008. 3266

[25] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *International Conference on Machine Learning*, 2007. 3266

[26] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1530–1535, 2006. 3266

[27] F. Zhou, F. De la Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Conference on Automatic Face and Gestures Recognition*, 2008. 3267