Modeling Transition Patterns Between Events for Temporal Human Action Segmentation and Classification

Yelin Kim[1], Jixu Chen[2], Ming-Ching Chang[2], Xin Wang[3],
Emily Mower Provost[1], and Siwei Lyu[3]
[1] Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, USA
[2] Computer Vision Lab, GE Global Research Center, Niskayuna, USA
[3] Computer Science Department, University at Albany, SUNY, Albany, USA

***Abstract*— We propose a temporal segmentation and classification method that accounts for transition patterns between events of interest. We apply this method to automatically detect salient human action events from videos. A discriminative classifier (e.g., Support Vector Machine) is used to recognize human action events and an efficient dynamic programming algorithm is used to jointly determine the starting and ending temporal segments of recognized human actions. The key difference from previous work is that we introduce the modeling of two kinds of event transition information, namely *event transition segments*, which capture the occurrence patterns between two consecutive events of interest, and *event transition probabilities*, which model the transition probability between the two events. Experimental results show that our approach significantly improves the segmentation and recognition performance for the two datasets we tested, in which distinctive transition patterns between events exist.**

I. Introduction

The pervasive installations of large camera networks and widely availability of digital video cameras have created a gigantic volume of video data that need to be processed and analyzed to retrieve useful information. As many videos involve human activities and behaviors, a central task and main challenge in video analytics is to effectively and efficiently extract complex and highly varying human-centric events. A general purpose event recognition system entails two essential steps: the localization of temporal segments in a video containing salient events (*when something happened*) and the classification of localized events into relevant categories (*what happened*). The extracted events can be piped for further analysis, such as indexing and retrieval of video collections in multimedia applications and suspicious behavior recognition in video surveillance.

Most update-to-date video event analysis methods treat event localization and classification as separate problems (e.g. [13], [15]). It has been noticed that these two problems are interrelated and can mutually bootstrap each other [3], [9]. Better event localization improves subsequent classification performance, while reliable event classification can be used as a guide for more precise localization. Based on this intuition, recent efforts have emerged in unifying both the localization and classification problems. These methods fall into two main categories: (i) *generative* approaches based on dynamic Bayesian models, such as the hidden Markov model (HMM) [1] and switching linear dynamical systems (SLDS) [16]; and (ii) *discriminative* approaches, which use maximum margin classifiers as in [2], [3], [9].

Conventional event models used in most existing methods only consider monolithic or persistent events. For example, action recognition focuses on the identification of action states such as walking or standing with arms folded. These methods ignore the regular transition patterns often exist between events of interest. To illustrate, consider a person with his/her arms down in a resting position who starts to raise his/her arm to touch his/her nose. A transition segment or event in which the arm moves upward governs the change between gesture states. Although a naive detection of such transition might be difficult (following the generative or discriminative approaches), the consecutive motion flow in between the transitions is indeed unique and recognizable. Explicitly incorporating *transition patterns* into the recognition framework will provide more reliable cues to localize and recognize persistent events.

In this paper, we propose a new method that jointly analyzes video events with precise temporal localization and classification, by modeling arbitrary transition patterns between events. It improves event recognition rates by leveraging the clearly identified event boundaries. Our method combines two approaches together by explicit modeling of *event transition segments*: (i) large margin discriminative learning of distinct event patterns (also introduced in [3], [9]) and (ii) generative event-level transition probability models. The event location and classification can be found by an efficient dynamic programming (DP) inference. Our framework is general to any time series data that have transition patterns between events and is applicable to problems outside video analytics. For human action recognition in particular, the use of transition patterns can greatly improve performance. Since even the same action (e.g. touching face) can be highly varying in both spatial and temporal domains, their transition patterns are more important for robust systems. Explicit consideration of transition patterns increases robustness and can provide critical information for decision making [17], [20], [25].

We focus on the application of video-based human action recognition. Specifically, we extract per-frame human pose estimation cues (i.e. body joint coordinates) [19] as a time series signal. We compute variable-length segment-level features using statistical functionals and linear regression

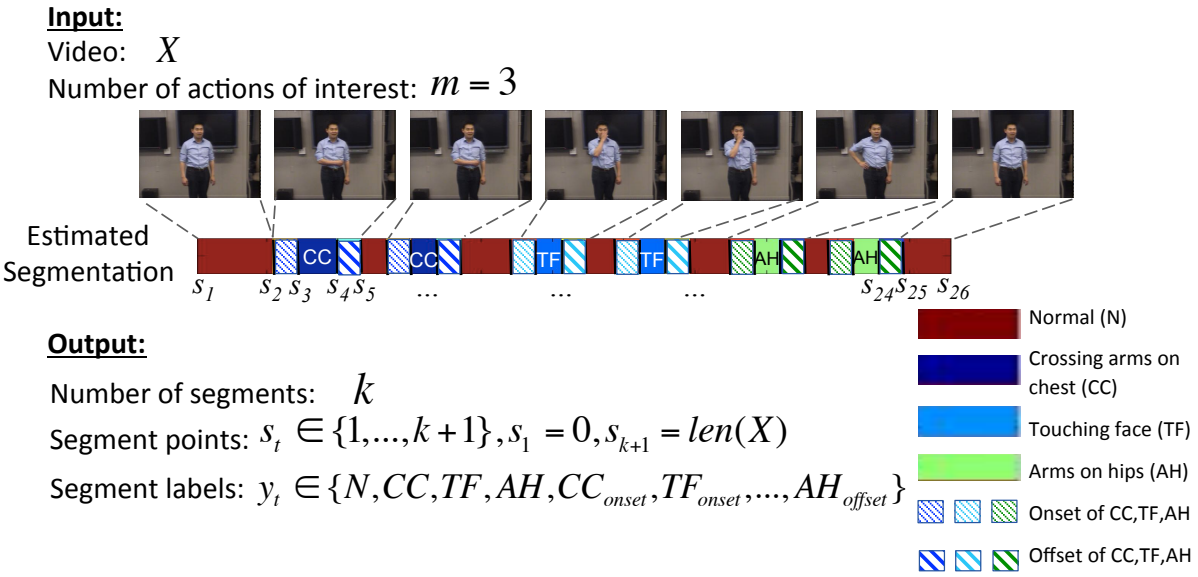

Fig. 1: Overview of the proposed video event localization and classification framework, where the event types are, e.g., *Crossing arms on Chest* (CC), *Touching Face* (TF), *Arms on Hip* (AH), and *Neutral* (N) (Section IV-A). The temporal onset and offset transitions between these events are optimally solved by efficient dynamic programming.

coefficients (slope) of the frame-level features for each segment. In the supervised training phase, we use labeled intervals of video events and their corresponding event types to train a discriminative model. This model is used in the testing phase, in which for a given test video, we infer the best segmentation start and end points with corresponding event labels, by searching for the highest pattern matching score and transition probability using efficient dynamic programming. Figure 1 provides an overview of our framework.

Our method has demonstrated significantly improved classification and localization performance on a newly collected video dataset and a public CMU-MAD [10] benchmark dataset, in comparison to a state-of-art work [9].

II. Related Work

Human action recognition is an active research area in computer vision [18], [22], [26].

Video segmentation. Segmentation of videos into salient events is an important task in video analysis that facilitates the retrieval, indexing, annotation, and representation of video data [12]. Traditionally it entails shot boundary detection, i.e., the complete segmentation of a video into continuously imaged temporal segments [5].

Video event recognition. A recent research trend in temporal segmentation is based on salient events of interest, rather than continuously recorded images, e.g., [9], [14], [21], [28]. Tang et al. studied Hidden Markov Model (HMM)-based models to learn the temporal structure of complex events in Internet videos [21]. They utilized a variable-duration HMM to model the durations and transitions of an event segment of interest, where the model is trained in a discriminative, max-margin fashion. They achieved competitive accuracies on activity recognition and event detection tasks. However, their work differs from ours in that a video clip with a single event label is analyzed, instead of a video sequence with multiple events. Hoai et al. [9], Cheng et al. [3], and Zhou et al. [28] studied the temporal segmentation of human action videos that contain multiple action events. Hoai et al. jointly localized and classified action events using a max-margin classifier and DP, which is most relevant to our work [9]. The main difference is that our approach benefits from the inclusion of transition events (i.e. events between two salient events of interest). The introduction of event transitions and the probabilistic modeling, and an efficient implementation are the key novelties of our work. Cheng et al. demonstrated the importance of temporal dependencies between events in joint segmentation and classification tasks [3], by applying the Sequence Memorizer [27]. The main difference of our work is that our system identifies events at the individual frame level, whereas the work of Cheng et al. represents a video using visual words of fixed-length sub-sequences. Zhou et al. studied unsupervised temporal clustering of human motion using the kernel k-means algorithm with the generalized dynamic time alignment kernel [28]. Our work differs from [28] in that we utilize the event-level transition information, to capture longer-range temporal information of human motions.

Generative and discriminative event modeling. Transition events have been handled using generative models (e.g., transition matrix in HMM) [8] and modeled as individual transition events in specific domains, for example the onset and offset states in facial Action Unit recognition. Galata et al. used variable-length Markov models that temporally segmented human activities into atomic behavior components [8]. Valstar et al. presented a hybrid SVM/HMM system to segment a facial action into temporal phases (e.g. onset, offset, peak, and neutral states), with a noticeable performance gain [23], [24]. They used a sigmoid function

operating on the SVM outputs as an emission probability for HMMs (instead of traditional Gaussian mixture models, since SVMs discriminate extremely well). Several studies have demonstrated the efficacy of using transition information for temporal segmentation of videos [21].

Event transition in facial movements. Studies in facial Action Units (AU) detection have demonstrated the utility of event transition information [7], [11], [23], [23]. AUs are anatomical facial muscle actions based on the Facial Action Coding System (FACS), where 9 upper face AUs and 18 lower face AUs are defined [24]. The set of AU's can be categorized by their transition states into *onset* (muscles contracting and expression becoming stronger), *peak* (with consistently strong expression), and *offset* (muscles relaxing back to neutral appearance) phases. The order of the phases are often "*neutral-onset-peak-offset-neutral*", whereas spontaneous facial expressions with multiple peaks and other ordering are also possible [4], [23]. Koelstra et al. introduced a combination of discriminative frame-based GentleBoost ensemble learners and used a dynamic generative HMM to detect AU and its temporal segments [11]. The 'cascade of tasks' of Ding et al. combines outputs of different tasks (frame, segment, and transition detection) linearly for the final AU event detection [7]. The combination parameters are learned by cross-validation, and independent onset and offset detectors were trained using a linear SVM for transition detection.

To our best knowledge, the use of transitions in discriminative learning has not been extensively exploited for event recognition, in particular for the purpose of joint localization and classification of complex video events.

III. Proposed Method

Our method can be applied to general tasks of segmenting human actions with transition patterns. Our proposed algorithm (Equation 2) is generic to model arbitrary transitions between actions, and transitions between actions and neutral states (e.g., standing person with hands down). Any transition event model can be applied based on the transition characteristics that reflect the nature of the problem or the dataset. However, neutral states between events are prevalent in the datasets we performed experiments on, and thus it is important to model them effectively in our chosen transition event model. We describe our event transition model with segment transition probabilities in Section III-A. We then describe our generic method for event finding, localization and classification: the training of a multi-class SVM using the peak and transition segments (Section III-B.1) and the inference and labeling of each putative temporal segments using the SVM and dynamic programming (Section III-B.2).

A. transition Event Model

Event Peak and Transition Segments. Any transition event model can be used to describe the temporal characteristics present between events of interest. Since the two datasets we tested have prevalent neutral states between events, we explicitly models four types of segments in this paper:

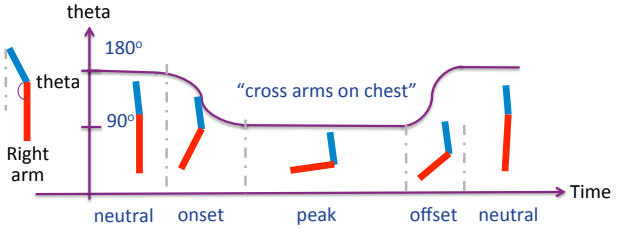

Fig. 2: transition event model example: the *neutral-onset-peak-offset-neutral* model of *cross arms on chest*. For visualization purpose, the joint angle θ between the upper and lower arms is shown as a cue to segment out the "cross arms" and "arm-down" events.

neutral, peak, onset, and offset. *Neutral* segments describe no significant visual cues of any event of interest. *Peak* segments describe salient and consistent visual cues of an event of interest. Both the definitions of neutral and peak can be application dependent (see Section IV). For each event type, we define two types of event transition segments based on the neutral and peak segments: *Onset* transition segments describe the transition from neutral to peak events, and *Offset* transition segments describe the transition from peak back to neutral.

In many video event analytic applications, segments of no particular utility or interest can be modeled as neutral events. Visual cues of onset transitions of the same peak event share commonalities (and the same for offset transitions). Thus a repeating sequence of "*neutral-onset-peak-offset-neutral*" can be found in many event types of interest. For instance, Figure 2 shows an example of neutral, onset, offset, and peak segments for the action event corresponding to "crossing arms on chest." We assume a simpler event model that does not consider direct transitions between events without going through the neutral event. This assumption effectively reduces the modeling of rarely occurred transitions, as supported by our experimental results.

Segment-level Transition Probability. We model the temporal patterns between neutral, peak, onset, and offset segments using a transition probability matrix. Following the *neutral-onset-peak-offset-neutral* observation from the training dataset, the transition probability from peak to offset, offset to normal, and onset to peak can be equally assigned to a default value based on the frequencies of event transitions. For the transition from neutral states, we model two cases: (i) the changing to one of the m types of possible events is modeled with a transition probability P, or (ii) the event remains unchanged, which is modeled with a *self-transition* probability γ. In this work, γ was chosen as 0.5 to maximize the randomness of repeating the same events.

B. SVM-based Event Localization and Classification

The input and output notations of our proposed system are described in Figure 1. We first train a multi (M)-class SVM using event peak and transition segments (vs. neutral segments). In testing, for a given video X without any segmentation information, we automatically find the optimal number of segments k, the temporal start and end points of

each segment $s_t, t \in 1, ..., k+1$, where $s_1 = 0$ and $s_{k+1} = len(X)$ the length of X, and segment labels $y_t, t \in 1, ..., k$. Our method keeps track of the highest sum of SVM scores and the log transition probability of all segments.

1) Training Segment-SVM with Max Margin Optimization: We learn discriminative patterns of each peak and transition segments using a multi-class SVM [6] similar to [9]. For each video sequence in the training data X^i, where $i \in \{1, 2, ..., n\}$, with known segments $t \in \{1, 2, ..., k_i\}$, where k_i is the number of segments of the i-th video sequence, we solve the following SVM and learn weights w_j for inference:

$$\min_{w_j, \xi_t^i \geq 0} \frac{1}{2M} \sum_{j=1}^{M} ||w_j||^2 + C \sum_{i=1}^{n} \sum_{t=1}^{k_i} \xi_t^i, \quad (1)$$
$$\text{s.t.} (w_{y_t^i} - w_y)^T \varphi(X^i_{(s_t^i, s^i_{(t+1)})}) \geq 1 - \xi_t^i, \forall i, t, y \neq y_t^i,$$

where $\varphi(X^i_{(s_t^i, s^i_{(t+1)})})$ is the segment-level feature of the segment $X^i_{(s_t^i, s^i_{(t+1)})}$, consisting of frames from s_t^i to $s^i_{(t+1)}$. We describe the segment-level feature mapping in detail in Section IV.

2) Efficient Inference with Dynamic Programming: **Transition-based Segmentation.** For each test video sequence X with unknown segment points and labels, we segment and classify the sequence using the following optimization function that maximizes the sum of the total SVM scores and the log transition probability between consecutive segment pairs:

$$\max_{k, s_t, y_t} \sum_{t=1}^{k} w_{y_t}^T \varphi(x_t) + (1+\gamma) \log P(y_t | y_{t-1}), \text{ s.t.} \quad (2)$$
$$l_{min} \leq s_{t+1} - s_t \leq l_{max}, \forall t,$$
$$s_1 = 0, s_{k+1} = len(X),$$

The intuition is to maximize the sum of segment-specific scores for each segmentation configuration, i.e. determine the number of total segments k, segment points s_t, and segment labels y_t, where $t \in \{1, 2, ..., k+1\}$, as well as the probability of transition from one segment to another. l_{min} and l_{max} are the minimum and maximum length of segments in the training data.

The relationship between temporally adjacent segments $(1+\gamma) \log P(y_t | y_{t-1})$ is calculated based on our prior transition probabilities described in Section III-A. Our novelty compared to Hoai et al. [9] is the $\log P(y_t | y_{t-1})$ term that explicitly considers event transitions in the optimization framework. Our work also differs from [9] in that non-maxima suppression based segmentation is performed (instead of a maximum SVM score based segmentation). Hoai et al. chooses the optimal segmentation that maximizes the difference of SVM scores between the best and the second best class, by filtering using the Hinge loss. We take a different approach by seeking the optimal segmentation that maximizes the sum of both (i) the SVM score of the segment class and (ii) the transition probability between consecutive segments.

Inference using DP. To solve Eq.(2) efficiently, we formulate the following function f to determine the best segmentation for the truncated time series $X_{(0,u]}$,

$$f(u, y_k) = \max_{k, s_t, y_t} \sum_{t=1}^{k} w_{y_t}^T \varphi(x_t) + (1+\gamma) \log P(y_t | y_{t-1}), \quad (3)$$

where k is the number of segments for the truncated $X_{(0,u]}$. u can be considered as the increasing "front" of the dynamic programming (DP) formulation. Since the transition probability depends on the last segment's label y_k of the truncated time series $X_{(0,u]}$, each f value depends on u as well as y_k. Therefore, for every tuple $u \in (0, len(X))$, $l \in [l_{min}, l_{max}]$ and class $y \in \{1, 2, ..., M\}$, we calculate $\eta(u, l, y) = w_y^T \varphi(X_{(u-l,u]})$ for inference, where η is the SVM score of the segment $X_{(u-l,u]}$. Dynamic programming computes $\max_{y_k} f(len(X), y_k)$ efficiently using Equation 4. Algorithm 1 lists the pseudo code, where w is a learned weight vector, *testX* and $len(X)$ are test video sequence and the number of frames of it, m_{tr} and std_tr are mean and standard deviation of each feature dimension in the training data for z-standardization, nCl is the number of classes, and *transMat* is a transition matrix to calculate f.

$$f(u, y_k) = \max_{l, y_{k-1}} f(u-l, y_{k-1}) + \eta(u, l, y_k)$$
$$+ (1+\gamma) \log P(y_k | y_{k-1}) \quad (4)$$

Algorithm 1: DP with transition Event Model

Data: learned weight vector w, test video X, m_{tr}, std_{tr}, l_{min}, l_{max}, number of classes nCl
Result: f, $bestL$, $bestY_{k-1}$
for *each frame* u = l_{min} *:* $len(X)$ **do**
  for *each last segment label* y_k = *1:*nCl **do**
    for l = l_{min}*:*$min(l_{max}, u-1)$ **do**
      Calculate $\eta(u, l, y) = w_y^T \varphi(X_{(u-l,u]})$, where $\varphi(X_{(u-l,u]})$ is z-standardized using m_{tr} and std_tr.
    end
    for *each second last segment label* y_{k-1} = *1:*nCl **do**
      $f_{temp}(l, y_{k-1}) =$
      $f(u-l, y_{k-1}) + \eta(u, l, y_k) + \log P(y_k | y_{k-1})$
    end
    find y^*_{k-1}, l^* that maximizes $f_{temp}(u, y_k)$.
    $f(u, y_k) = f_{temp}(l^*, y^*_{k-1})$ $bestL(u, y_k) = l^*$
    $bestY_{k-1}(u, y_k) = y^*_{k-1}$
  end
end
Use $f, bestL, bestY_{k-1}$ for back-tracking

The complexity of our algorithm is $O(M^2(l_{max} - l_{min} + 1)(len(X) - l_{min} + 1))$.

IV. Experiments

We evaluate our method for joint segmentation and classification of video events on two datasets: (i) the Smartroom Dataset we collected for real-life suspicious behavior recognition and (ii) the public CMU-MAD human action dataset

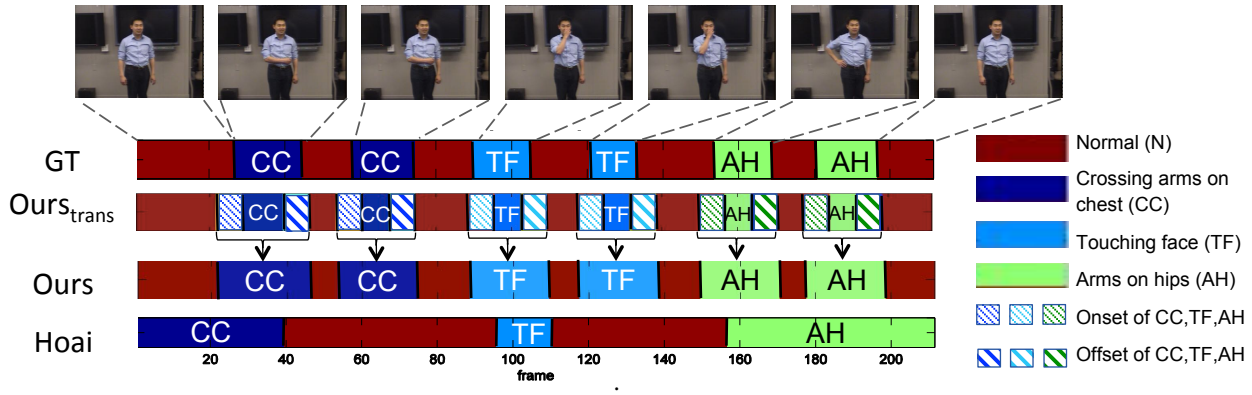

Fig. 3: Evaluation results from our Smartroom (Clean) Dataset (video 1). The four rows of illustrations depict ground truth (first row), result of our method with transition segments (second row), result of our method with combined transition (onset and offset) segments into a single action segment to match the comparison of Hoai et al. (third row), and SVM+DP method output presented by Hoai et al. [9] (bottom row), respectively.

[10]. Both of the datasets contain large variability in human poses and actions.

We compare the performance of our algorithm to the SVM-DP algorithm of Hoai et al. [9]. For a fair comparison to the SVM-DP algorithm of Hoai et al., which does not consider the transition segments, we calculate the recognition rate after transferring the estimated M action classes with transition segments, where $M = \{m \text{ peak events}\} + \{1 \text{ neutral event}\} + \{m \text{ offset events}\} + \{m \text{ onset events}\}$, to m peak action classes, as shown in Figure 3. We combine the detected onset, offset, and peak segments of each action into one action. For instance in our Smartroom Dataset, after we finish back-tracking and get 10-class labels for each detected segment, we combine onset, offset, and peak segments into one action segment to match the 4-class ground-truth labels.

We report the performance of both algorithms in terms of frame-level and event-level recognition rates. (i) Frame-level recognition rate measures the ratio of frames that are correctly classified. We compute frame-level precision ('Prec'), recall ('Rec'), and F-measure ('F-mea'). The accuracy is calculated as $(TP+TN)/(TP+TN+FP+FN)$, where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. (ii) The measure of event-level recognition rate is suggested in [10] to reflect the ratio of event segments that are correctly identified, by counting the number of correct frames that overlaps with 50% of a segment. We evaluate event-level precision, recall, and F-measure. Event-level precision ($prec$) computes the ratio between the number of correctly detected events and the number of detected events and event-level recall (rec) computes the ratio between the number of correctly detected events and the number of ground truth events. Event-level F-measure computes the balanced F-score using $2 * \frac{prec*rec}{prec+rec}$. In our datasets where there is at most 9 ground truth events, our event-level recognition rate is highly sensitive compared to frame-level recognition rates.

A. *Smartroom Dataset*

We create a new Smartroom Dataset with volunteers performing a series of upper body actions, where the challenge is that both the temporal durations of events and the number of events are unknowns. The dataset contains six subjects performing a mix of the following actions in 8 videos: Crossing arms on chest (CC), Touching face (TF), Arms on hip (AH), and Normal (N). Each action is repeated two to three times in each video. Normal action represents the case of hands down in a resting position. The average length of the videos is 47.8 seconds. Each of the {CC, TF, AH} actions was enacted sequentially following the "neutral-onset-peak-offset-neutral" pattern for the right arm, left arm, and both arms. The enacted events share a large extent of variations in terms of temporal durations and spatial locations.

We use the MODEC algorithm [19] to estimate per-frame body pose cues to serve as action features, and we employ a Kalman filter to produce a smooth pose time series. The pose estimation from the image is converted into body joint angles as shown in Figure 5. The performance of MODEC pose estimation varies for different clothing and illumination conditions. We evaluate the robustness of event recognition upon such variability in the input data. We divide the Smartroom dataset into two subsets and evaluate our system for each subset: (i) the ones with more accurate pose estimation ("Clean"), (ii) the remaining with large pose estimation noise due to appearance and clothing variations ("Noisy"). Comparisons of the MODEC pose estimations on the two subsets are shown in Figure 4. The Smartroom (Clean) dataset contains three videos, and the Smartroom (Noisy) dataset contains five videos.

Two types of segment-level features φ are extracted for each video segment: (1) the first and second-order statistics (mean and standard deviation) of the frame-level features, and (2) the linear regression coefficient (slope) across frames within each segment, which captures the dynamics of the changes of the frames within the segment. We perform z-standardization to normalize the segment-level features as

(a) (b)

Fig. 4: Pose estimation comparison between the Smartroom (a) Clean and (b) Noisy datasets for *Crossing arms on chest* (top), *Touching face* (center), and *Putting arms on hip* (bottom) actions. The performance of the MODEC algorithm [19] varies for different clothing and illumination conditions. The Smartroom (Clean) dataset shows more accurate pose estimation than the Smartroom (Noisy) dataset.

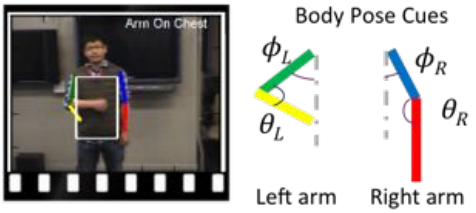

Fig. 5: Estimated body pose cues of our Smartroom Dataset utilized for frame-level features. We estimate the four joint angles at the shoulders (between torso and upper arms: ϕ_L, ϕ_R) and the elbows (θ_L, θ_R).

follows: we first find the mean m_i and standard deviation st_i of each feature dimension i in the training data and normalize the training data (z-standardization) using the two statistics. Then, during the inference, we use the same mean m_i and standard deviation st_i of each feature dimension to normalize the test segments in the Dynamic Programming steps.

For ground truth segment configurations, two human annotators labeled both (1) the start and end timing of peak segment, and (2) the action label of the three pre-defined actions. We add three frames prior-to and post-to each peak boundary, and define non-overlapping onset, peak, offset, and neutral segments. The onset and offset segments are always chosen to be 7 frames in length.

We perform leave-one-video-out cross validation, and take a subset (left-hand movements) of a video as a test sequence. We train our model using the remaining videos. Figure 3 shows the segmentation result comparison between the ground truth *(top)*, our algorithm *(center)*, and the algorithm presented by Hoai et al. ([9], "Hoai SVM+DP") *(bottom)*. Both methods determine the start and end points, as well as the label of each action event. Our method significantly outperforms the method of Hoai et al. in terms of both frame and event-level recognition rates.

Tables I and II show the comparisons between our algorithm and the algorithm of Hoai et al. for the Smartroom (clean) and Smartroom (noisy) datasets, respectively. For the Smartroom (clean) dataset, our algorithm has a frame-level precision of 83.84%, recall of 80.41%, and an F-measure of 81.95%. All of the frame-level recognition rates are higher than the SVM-DP method of Hoai et al. by 27.65%, 19.91%, 23.79% (relative improvements) in terms of precision, recall, and F-measure, respectively. Also, event-level precision, recall, and F-measure of our algorithm are 86.67% 89.63% 88.07%, respectively, 15.55%, 22.22%, and 19.75% higher than the method of Hoai et al. Our algorithm also demonstrates improvements in performance even when the pose estimation was noisy. For the Smartroom (noisy) dataset, our algorithm shows a frame-level precision of 44.41% , recall of 40.38%, and F-measure of 41.33%; relative improvement of 20.02%, 26.78%, and 24.07%, compared to the method of Hoai et al. The event-level recognition rates are also significantly improved when using our algorithm. The event-level precision of our system is 25.36 %, recall was 54.45%, and F-measure was 33.51%. These are 11.03%, 43.24%, and 21.76% relative improvement over the method of Hoai et al [9]. This demonstrates that with a presence of clear transitions between actions, our algorithm can robustly segment and classify each salient action.

B. CMU-MAD Dataset

We test our method on the CMU-MAD dataset [10], which contains 35 human actions of 20 subjects recorded using a Microsoft Kinect sensor. Similar to the Smartroom Dataset, we use the joint angles of elbows and shoulders as frame-level features, and utilize the same segment-level features φ mapping as in the Smartroom Dataset, i.e. mean, standard deviation, and linear regression slope. The start and end time of each action are provided in this dataset. However, the timings can not be directly used in our *neutral-onset-peak-offset-neutral* model, since the action between the start and end time contain all of the *neutral, onset, peak, offset,* and *neutral* events. Due to the specific labeling scheme of this dataset, it is reasonable to separate each labeled action segment into three sub-sequences: [0-33.3%] for onset, [33.3-66.6%] for peak, and [66.6-100%] for offset. We focus on the evaluation of 9 actions that contain meaningful transitions and exclude actions such as running (where the action peak as well as onset/offset transitions are not clearly defined). These selected 9 actions are: left/right arm waving, left/right arm pointing to the ceiling, crossing arms on the chest, basketball shooting, both arms pointing to both sides and left/right side.

We perform 5-fold cross validation over the 20 subjects

	Frame-level						Event-level					
	Prec		Rec		F-mea		Prec		Rec		F-mea	
Method	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Ours	83.84	7.45	80.41	12.18	81.95	9.52	86.67	11.55	89.63	10.02	88.07	10.54
Hoai	56.19	5.32	60.50	7.98	58.15	5.74	71.11	7.70	67.41	12.24	68.32	3.86
Diff	**27.65**		**19.91**		**23.79**		**15.55**		**22.22**		**19.75**	

TABLE I: *Recognition rate (%) of Smartroom (Clean) Dataset using our proposed algorithm and the Hoai et al. [9] at the frame and event level (see text). The last row ("Diff") shows the relative improvement of using our algorithm over the algorithm of Hoai et al.*

	Frame-level						Event-level					
	Prec		Rec		F-mea		Prec		Rec		F-mea	
Method	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Ours	44.41	18.85	40.38	18.20	41.33	17.09	25.36	16.36	54.45	15.91	33.51	17.93
Hoai	24.39	11.54	13.60	6.88	17.26	8.33	14.33	14.93	11.20	6.81	11.75	10.56
Diff	**20.02**		**26.78**		**24.07**		**11.03**		**43.24**		**21.76**	

TABLE II: *Recognition rate (%) of Smartroom (Noisy) Dataset using our proposed algorithm and the Hoai et al. [9] at the frame and event level (see text). The last row ("Diff") shows the relative improvement of using our algorithm over the algorithm of Hoai et al.*

	Frame-level						Event-level					
	Prec		Rec		F-mea		Prec		Rec		F-mea	
Method	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Ours	85.00	8.82	71.41	7.25	77.41	7.01	74.40	15.02	85.02	12.17	78.83	12.95
Hoai	73.79	9.62	70.57	9.96	71.87	8.70	73.45	15.84	83.88	13.06	77.85	14.23
Diff	**11.21**		**0.84**		**5.54**		**0.95**		**1.14**		**0.98**	

TABLE III: *Recognition rate (%) comparison on the CMU-MAD dataset using our proposed algorithm ("Ours") and the Hoai et al. ("Hoai", [9]) at the frame and event level (see text). The last row ("Diff") shows the relative improvement.*

and measure the event-level performance as suggested in [10]. Each fold contains videos of 4 subjects, each with 2 video sequences, in total 8 video sequences. We train our model using segments of the four folds and test our model for the held out. Due to the computational cost, we use DP over sliding windows of 500 frames (about 25% length of a video sequence) along the test time series as in [10], to solve for the optimal segment configuration that maximizes the sum of the SVM scores and the event transition probability.

Figure 6 shows the results of our algorithm (center) and Hoai's SVM+DP method (bottom), along with the ground truth segmentation (top). Table III summarizes the results. All of our frame-level recognition measures are higher than the SVM-DP method of Hoai et al. [9]. For event-level accuracy, our event-level precision (74.40%), recall (85.02%), and F-measure (78.3%) are higher than the SVM-DP method, by 0.95%, 1.14%, and 0.98%, respectively.

Our method improves the frame-level recognition rates compared to the previous work of Hoai et al. [9], achieving 85.00% (precision), 71.41% (recall), and 77.41% (F-measure), corresponding to relative improvement of 11.21%, 0.84%, and 5.54%, respectively. We achieve an event-level precision of 74.40%, recall of 85.02%, and F-measure of 78.83%, and all of these are slightly higher than that of Hoai et al. by 0.95%, 1.14%, and 0.98%, respectively. The improvement in both frame and event-level recognition rates using our algorithm over the previous method of Hoai et al. [9] demonstrates that for actions of interest with distinguishable transition patterns, our algorithm can localize and classify the action segments more effectively.

Regarding the difference between the Smartroom and CMU-MAD dataset results in performance gain, we raise two major points: (i) the transition segments were not explicitly labeled for the CMU-MAD dataset, therefore the segments were estimated during training. Since the major advantage of our method is a better modeling of the transition states, the improvement on CMU-MAD dataset is marginal. This also explains a greater performance gain in the frame-level compared to the event-level accuracy. In comparison, our Smartroom dataset includes clearer labeling in event transitions; hence the performance improves significantly due to better transition modeling. (ii) The visual features for the Smartroom dataset (i.e., pose estimation features from RGB cameras without depth information) are more difficult to estimate and thus are noisier than those of the CMU-MAD dataset (i.e., 3D pose estimation features using Kinect sensor). Therefore, a better transition model as a prior results in a greater performance gain on the Smartroom dataset, where the input features are noisier in nature.

V. Conclusions

In this work, we describe a new method combining discriminative large margin classification with generative modeling, where the explicit modeling of event transition segments improves the state-of-art performance on the joint

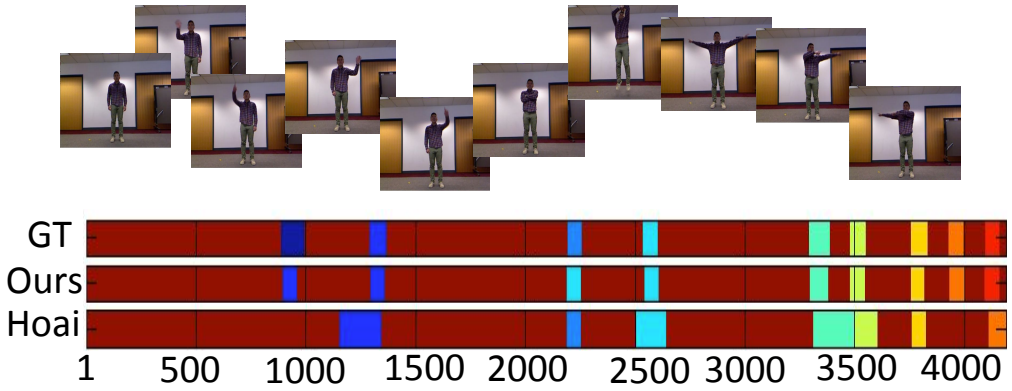

Fig. 6: An example result (subject 20, sequence 20) from the CMU-MAD evaluation. Ground truth (top), our method (center), and SVM+DP presented by Hoai et al. [9] (bottom). Best viewed in color. The image is from the CMU-MAD dataset [10].

localization and classification of video events. Our experimental results on two benchmark datasets shows promising recognition rates. An important future work we plan to pursue is the consideration of event transition probability with discriminative learning in finding an effective solution to model the full relationships between events.

Nevertheless, there is still room for improvement in the current work. In particular, though this work demonstrates that the modeling of onset and offset of event transitions can boost the localization and classification of video events, while effective solution to properly model the full relationships between pairwise events are yet to be explored. In future work, we will study automatic methods that can learn the transition probabilities of the full set of pairwise event transitions.

VI. Acknowledgement

The work is funded by Morpho Detection LLC. The authors gratefully acknowledge the support.

References

[1] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE PAMI*, 22(8):844–851, 2000.

[2] A. Chan-Hon-Tong, C. Achard, and L. Lucat. Deeply optimized hough transform: Application to action segmentation. In *ICIAP*, pages 51–60. Springer, 2013.

[3] Y. Cheng, Q. Fan, S. Pankanti, and A. Choudhary. Temporal sequence modeling for video event detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2235–2242. IEEE, 2014.

[4] J. F. Cohn and K. L. Schmidt. The timing of facial motion in posed and spontaneous smiles. *IJWMIP*, 2(02):121–132, 2004.

[5] C. Cotsaces, N. Nikolaidis, and I. Pitas. Video shot detection and condensed representation. a review. *Signal Processing Magazine, IEEE*, 23(2):28–37, 2006.

[6] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2-3):201–233, 2002.

[7] X. Ding, W.-S. Chu, F. D. L. Torre, J. F. Cohn, and Q. Wang. Facial action unit event detection by cascade of tasks. In *IEEE ICCV*, pages 2400–2407. IEEE, 2013.

[8] A. Galata, N. Johnson, and D. Hogg. Learning variable-length markov models of behavior. *CVIU*, 81(3):398–413, 2001.

[9] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *IEEE CVPR*, pages 3265–3272. IEEE, 2011.

[10] D. Huang, Y. Wang, S. Yao, and F. De la Torre. Sequential max-margin event detectors. In *ECCV*, 2014.

[11] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *IEEE PAMI*, 32(11):1940–1954, 2010.

[12] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE CVPR*, pages 1–8. IEEE, 2008.

[14] M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *IEEE ICCV*, pages 1925–1932. IEEE, 2009.

[15] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, pages 392–405. Springer, 2010.

[16] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert. Learning and inferring motion patterns using parametric segmental switching linear dynamic systems. *IJCV*, 77(1-3):103–124, 2008.

[17] A. Pentland and A. Liu. Modeling and prediction of human behavior. *Neural computation*, 11(1):229–242, 1999.

[18] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[19] B. Sapp and B. Taskar. Multimodal decomposable models for human pose estimation. In *IEEE CVPR*, pages 3674–3681, 2013.

[20] G. Schöner, H. Haken, and J. Kelso. A stochastic theory of phase transitions in human hand movement. *Biological cybernetics*, 53(4):247–257, 1986.

[21] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *IEEE CVPR*, pages 1250–1257. IEEE, 2012.

[22] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[23] M. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(1):28–43, Feb 2012.

[24] M. F. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *Human–Computer Interaction*, pages 118–127. Springer, 2007.

[25] W. H. Warren. The dynamics of perception and action. *Psychological review*, 113(2):358, 2006.

[26] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 115(2):224–241, 2011.

[27] F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y. W. Teh. A stochastic memoizer for sequence data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1129–1136. ACM, 2009.

[28] F. Zhou, F. De la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE PAMI*, 35(3):582–596, 2013.