

# A Framework for Online Segmentation and Classification of Modeled Actions Performed in the Context of Unmodeled Ones

Dimitrios Kosmopoulos, Konstantinos Papoutsakis, and Antonis Argyros

**Abstract**—In this paper, we propose a discriminative framework for online simultaneous segmentation and classification of modeled visual actions that can be performed in the context of other unknown actions. To this end, we employ Hough transform to vote in a 3D space for the begin point, the end point, and the label of the segmented part of the input stream. A support vector machine is used to model each class and to suggest putative labeled segments on the timeline. To identify the most plausible segments among the putative ones, we apply a dynamic programming algorithm, which maximizes the likelihood for label assignment in linear time. The performance of our method is evaluated on synthetic as well as on real data (Weizmann, TUM Kitchen, UTKAD, and Berkeley Multimodal Human Action databases). Extensive quantitative results obtained on a number of standard data sets demonstrate that the proposed approach is of comparable accuracy with the state-of-the-art approaches for online stream segmentation and classification when all performed actions are known, and performs considerably better in the presence of unmodeled actions.

**Index Terms**—Action recognition, action spotting, Hough transform.

## I. INTRODUCTION

IN THIS paper, we deal with the problem of online segmentation of visually observable actions, i.e., we have to provide action labels given the fact that the visual observations arrive streamwise in a sequential fashion and we need to decide on the label shortly after they are received, without having available the full sequence.

Manuscript received October 30, 2015; revised April 6, 2016; accepted July 6, 2016. Date of publication July 9, 2016; date of current version December 13, 2017. This work was supported in part by the European Commission FP7 Project HOBBIT UNDER Grant 288146, in part by the FP7 Project RoboHow.Cog under Grant 288533, and in part by the H2020 Project ACANTO under Grant 643644. This paper was recommended by Associate Editor C. Shan. (Corresponding author: Dimitrios Kosmopoulos.)

D. Kosmopoulos is with the Department of Cultural Heritage Management and New Technologies, University of Patras, Agrinio GR 30100, Greece, and also with the Institute of Computer Science, Foundation for Research and Technology–Hellas, Heraklion GR 70013, Greece (e-mail: dkosmo@upatras.gr).

K. Papoutsakis and A. Argyros are with the Institute of Computer Science, Foundation for Research and Technology–Hellas, Heraklion GR 70013, Greece, and also with the Computer Science Department, University of Crete, Heraklion GR 70013, Greece.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2016.2589678

The video segmentation has been traditionally treated separately from the classification step; however, these two problems are very tightly coupled and can be better handled by simultaneously considering the low-level cues and the high-level models representing the candidate classes (see [1]–[3]). Following that observation, generative models can build probabilistic models of actions and can give the posterior of assigning labels to observations. In case an unknown activity appears, the posterior probability for the given known classes will be low, a fact that facilitates the segmentation of a sequence of observations corresponding to an unknown action. However, generative models rely on simplifying statistical assumptions for computing the joint probability of the states and the observed features, whereas a more general discriminative model may better predict the conditional probability of the states for the given observed features. As a result, several researchers have investigated the use of discriminative models of actions such as conditional random fields [4], support vector machines (SVMs) [2], [5], or random forests [6], [7]. However, the discriminative models are not without problems, since they cannot easily handle novel/unknown actions. Such actions are not considered in training, so when they appear, they are classified as instances of the known classes.

In the more general formulation of the action segmentation and classification problem, we cannot exclude the possibility of previously unseen actions. In dynamic scenes, it is almost certain that at some point, we will come across some observations that will not be explainable by the existing action models. Learning all the possible unknown classes and assigning a specific class label or multiple class labels to represent those is not the best solution, since a single or set of related action models has to be really complex to cover the variety of possible observations, and most importantly, these observations are not known in advance. Therefore, as in certain application domains, we treat unknown actions essentially as irrelevant/don't care in the proposed framework.

In this paper, we seek to mitigate the aforementioned limitation of the discriminative methods, by employing a discriminative Hough transform. By collecting the votes generated by action primitives, we detect putative segments, i.e., the

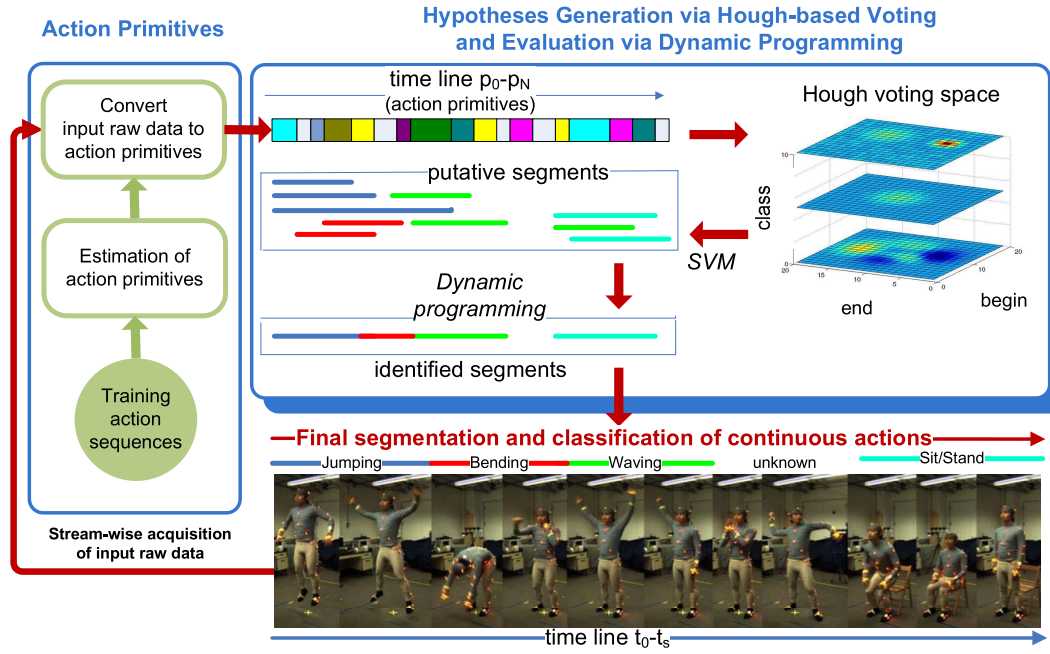


Fig. 1. Overview of the proposed framework. Top: action primitives in the considered time span vote in a 3D Hough voting space (begin–end class). The class-trained binary SVMs receive the votes and suggest the putative segments by assigning action labels. Dynamic programming is used to estimate the final solution, i.e., the labeling of the putative segments that maximize the likelihood. Bottom: visual observations of a test frame sequence of actions (the series of representative frames) arrive in a stream-wise sequential fashion and are further converted into action primitives, given the estimated set of those that are generated based on the training action sequences.

time span as well as the action type associated with each of them using an SVM. In a following step, we use the putative segments to assign labels to time instances so that the observations are best explained; to this end, we employ a dynamic programming algorithm. Fig. 1 gives an overview of the method.

More specifically, the innovations of the proposed approach are as follows:

- 1) a method to deal with unknown sequential patterns;
- 2) a generic voting scheme in a 3D Hough parameter space, which is defined by the start point, the end point, and the class-specific label in order to segment the observation stream in an online fashion;
- 3) a dynamic programming method for label assignment in linear time.

The rest of this paper is organized as follows. In the next section, we survey related work. In Section III, we describe the proposed framework, which includes the generation of hypotheses via voting and the evaluation via dynamic programming. Section IV describes the experimental results, Section V discusses the experiments and key features of the framework, and Section VI concludes this paper.

## II. RELATED WORK

Recently, the simultaneous segmentation and classification of visual or other time series has gained in popularity. *Generative models* have been used extensively. In [8], a Bayesian nonparametric approach was presented for speaker diarization that built on the hierarchical Dirichlet process hidden Markov model (HMM). Typical approaches that exploit

the hierarchical structure of time series to classify actions are the hierarchical HMMs [9] or the layered HMM [10]. The semi-Markov model, which explicitly captures the duration of events/actions, has also been employed [11], [12].

*Dynamic time warping* and its variations are also popularly [13] used a feature weighting variation for gesture recognition. Reyes *et al.* [14] proposed a framework for gesture recognition and spatiotemporal gesture segmentation using a stochastic dynamic time warping combined with a variation of the Viterbi algorithm. Another line of research is followed by methods that seek to exploit the *cooccurrence of tasks* (see [15], [16]). Our method does not currently exploit this information, but this solely depends on how we treat the overlapping tasks that we recognize. In this paper, we deal exclusively with the simpler case of actions that do not overlap with each other on the timeline. Recently, a great deal of work was done on *deep learning*, e.g., convolutional neural networks [17], [18] and restricted Boltzmann machines [19]. These methods can create a feature mapping in an unsupervised way and then they apply standard classification methods. Our method is agnostic to the employed features and could use these results.

In [20], a *discriminative framework* was proposed. The sequences were assigned to classes and segmented into subsequences using conditional random fields. However, the method requires the full sequence in advance and cannot operate in an online fashion. Similarly, conditional random fields were used in [21] and [22]. In [4], hierarchical layers of latent variables were used to model substructures within actions. In [1], a discriminative approach was introduced under a semi-Markov

model framework, and a Viterbi-like algorithm was devised to efficiently solve the induced optimization problem. In [2], a joint segmentation and classification scheme was presented and it sought to maximize the confidence of the segment assignment. To this end, a multiclass SVM was used and a dynamic programming approach was followed for efficient seeking of candidate segments and their evaluation. In [5], latent labels and state durations were optimized in a maximum margin approach. The results were very promising, but Tang *et al.* [5] relied on the assumption that the video sequences contain only instances of classes that were previously learned. These schemes have problems if segments belonging to previously unseen classes appear between the known ones because the dynamic programming scheme becomes inapplicable. A possible solution could be to model the content that does not belong to any of the known categories as a separate class; however, that approach would not handle properly the unknown sequences that might appear.

Of some relevance to our method is research relevant to *anomaly detection* in time series. In contrast to segmentation of time series, anomaly detection is the identification of unknown patterns, i.e., behaviors that deviate from normal, given the previously seen data. The work in [23] used the one-class SVM discriminative model to detect novel observations and outliers after transforming the time series data to a vector, which is the required input to the SVM. Such approaches can be used offline, where the whole sequence is known. Lee and Roberts [24] proposed an on-line (causal) novelty detection method capable of detecting both outliers and regime change points in sequential time-series data using a metric based on extreme value theory. This method is more related to change point detection methods used when a signal changes, e.g., in EEG analysis rather than to the classification of more complex patterns like actions or gestures. Our approach is different from anomaly detection and the approaches related to it, since our goal is not the detection of abnormal sequences; our primary goal is to segment some known sequential patterns, which could be occasionally interrupted by some other sequences. The later sequences in our experimental settings are not necessarily anomalous, and they can be just uninteresting, e.g., some random actions that cannot be modeled or should not be modeled because they do not correspond to known/interesting actions.

Another line of research is the definition of spatiotemporal approaches, which are followed by matching. In [25], local descriptors concatenated several histograms from a space–time grid defined on the patch and generalized the SIFT descriptor to space–time. A problem with that approach is the lack of structure capturing, due to the histogram representations. The work in [26] introduced a space-time descriptor based on visual space–time-oriented energy measurements, as well as techniques for matching. Such methods seem to work well for short-term actions, but do not have invariance to temporal variations, which are common in longer actions.

Related to our approach is *Hough transform*, which has been employed in some recent works for object detection, where each local part casts a weighted vote for the possible locations of the object center. In [27], the votes were based on

generative weights, while in [28]–[30], it was reported that this method produces false positives that can be partially reduced by introducing discriminative parameters. It was shown that the weights can be learned in a max-margin framework, which directly optimizes the classification performance.

Its resilience to noise and the fact that multiple objects can be present simultaneously make the Hough transform a very attractive option, which can be generalized to time series and therefore to gesture and action recognition. In [3], a new Hough-transform-based segmentation and classification method was proposed based on discriminative parameters, where the motion primitives voted for the center and the type of the action (2D voting). Then a convex optimization scheme was used to learn the Hough variables and thus how the primitives should distribute their votes on the timeline. The method was successfully applied to human action segmentation by relying on skeleton-based features.

We propose a discriminative Hough transform for time series analysis, where motion primitives are used instead of local descriptors, as opposed to [28]. We deal with concurrent segmentation and classification in time series, instead of object detection in images where the voting is different. We vote in a 3D space, which is defined by the time span and type of segment (begin point, end point, and class label) and therefore does not require learning how to vote as in [3]. Then we use dynamic programming for the final label assignment. Another interesting approach for the problem of action recognition using Hough was presented in [6], where the action segmentation was coupled to the action positioning problem for a single actor. By considering features such as optical flow, intensity, and position, a Hough forest was built and then used to cast votes in real scenarios. Compared with that work, we decouple the position estimation problem from the classification and segmentation problem, which reduces the dimensionality of the voting space. In [6], the actor was represented by a rectangle. Thus, it is unclear how such a coupled framework would generalize to more complex problems involving high-dimensional models (e.g., multiple actors, skeleton models, and region descriptors). The method presented in [31] also used the described a Hough-transform-based scheme to classify actions by comparing pose-based features derived from articulated 3D joint information and appearance-based and combined features for action recognition. The goal was to investigate whether pose estimation is useful for action recognition or if it is better to train a classifier only on low-level appearance features drawn from video data. The results showed that pose-based features can outperform appearance-based features. Hough transform was also used in [32]. However, unlike our work, which uses voting for actions, it used voting for pose estimation.

An early version of this paper has been presented in [33]. The current work extends [33] by updating the state-of-the-art section to include the latest developments, by proposing a new method for primitive selection, which improves our previous results, and by including two additional experiments with two standard data sets. Furthermore, we report a detailed quantitative comparison with other state-of-the-art methods.

### III. PROPOSED FRAMEWORK

#### A. Hypotheses Generation via Discriminative Voting

In the discriminative voting framework, we seek to simultaneously identify the following:

- 1) the instances of classes  $C$  as subsequences in time-series data;
- 2) the location  $\mathbf{x}$  of the class-specific subsequence, in other words the begin and the end time point of an action in the observations.

Let  $\mathbf{f}_t$  denote the feature vector observed at time instance  $t$  associated with an action primitive and let  $S(C, \mathbf{x})$  denote the score of class  $C$  at a location  $\mathbf{x}$  ( $(C, \mathbf{x})$  is a cell in a 3D voting space). The implicit model framework obtains the overall score  $S(C, \mathbf{x})$  by adding up the individual probabilities  $p(C, \mathbf{x}, \mathbf{f}_t)$  over all observations within a sliding window, that is

$$S(C, \mathbf{x}) = \sum_t p(C, \mathbf{x}, \mathbf{f}_t) = \sum_t p(\mathbf{f}_t) p(C, \mathbf{x} | \mathbf{f}_t). \quad (1)$$

We define  $M$  action primitives, which result from clustering of the visual observation vectors  $\mathbf{f}_t$ . Let  $P_t$  denote the action primitive in time  $t$ . By assuming a uniform prior over features and marginalizing over the action primitives, we get

$$\begin{aligned} S(C, \mathbf{x}) &= \sum_t p(C, \mathbf{x} | \mathbf{f}_t) \\ &= \sum_{i,t} p(P_t = i | \mathbf{f}_t) p(C, \mathbf{x} | P_t = i, \mathbf{f}_t). \end{aligned} \quad (2)$$

We observe that  $p(C, \mathbf{x} | P_t = i, \mathbf{f}_t)$  depends only on the matched primitive  $P_t$  and simplifies to  $p(C, \mathbf{x} | P_t = i)$ . Therefore, we obtain

$$\begin{aligned} S(C, \mathbf{x}) &= \sum_{i,t} p(P_t = i | \mathbf{f}_t) p(C, \mathbf{x} | P_t = i) \\ &= \sum_{i,t} p(P_t = i | \mathbf{f}_t) p(\mathbf{x} | C, P_t = i) p(C | P_t = i). \end{aligned} \quad (3)$$

The term  $p(P_t = i | \mathbf{f}_t)$  can be calculated by applying the Bayes rule assuming uniform distribution for  $\mathbf{f}_t$ :  $p(P_t = i | \mathbf{f}_t) \propto p(\mathbf{f}_t | P_t = i) p(P_t = i)$ . We use Gaussian mixture models (GMMs) to represent the distributions of the observation vectors and to express one primitive by one component of the GMM. The first factor can be simply obtained by evaluating the respective component of the GMM, while the latter is given by the associated prior.

Returning to (3), the term  $p(\mathbf{x} | C, P_t = i)$  gives the temporal distribution of the begin–end points  $\mathbf{x}$  for the given class  $C$  and with respect to the primitive  $P_t$ , i.e., what is the number of action primitives between  $P_t$  and the begin/end points. This can be learned from the training samples and is a simple bivariate discrete distribution. The third term is the weight of the primitive  $P_t$  emphasizing how confident are we that the primitive  $P_t$  at time  $t$  matches the class  $C$  as opposed to another class.

Our voting framework can be the basis for a discriminative voting scheme for time-series data. It is inspired by the framework presented in [28], which dealt with object detection. We can use maximum margin optimization if we observe

that the score  $S(C, \mathbf{x})$  is a linear function of  $p(C | P_t = i)$ . By considering (3), we obtain

$$\begin{aligned} S(C, \mathbf{x}) &= \sum_{i,t} p(P_t = i | \mathbf{f}_t) p(\mathbf{x} | C, P_t = i) p(C | P_t = i) \\ &\approx \sum_i p(C | P^i) \sum_t p(P_t = i | \mathbf{f}_t) p(\mathbf{x} | C, P_t = i) \\ &= \sum_i w_i \times a_i(\mathbf{x}, C) = W_c^T A(\mathbf{x}, C) \end{aligned} \quad (4)$$

where  $P^i$  is denoted by the  $i$ th primitive,  $A(\mathbf{x}, C) = [a_1 a_2, \dots, a_M]^T$  (hereafter mentioned as the activation vector), and  $a_i$  is given by

$$a_i(\mathbf{x}, C) = \sum_t p(\mathbf{x} | C, P_t = i) p(P_t = i | \mathbf{f}_t). \quad (5)$$

The weights  $W_c^T$  are class specific and they can be optimized in a discriminative fashion to facilitate labeling. For a given training sequence that is observed, we set the respective class-specific labels in the respective  $\mathbf{x}_i$ , i.e., at the bins that correspond to the correct begin/end points. The rest of the locations are defined to belong to an idle class. In other words, we define the ground-truth labels for all possible  $\mathbf{x}_i$  within a time window. For each of the  $\mathbf{x}_i$ , we find the activation vectors  $A(\mathbf{x}_i, C)$ , which are calculated using (5). Given the labels and the respective  $A(\mathbf{x}_i, C)$ , we calculate the weights  $W_c$  and the bias, which can be regarded as a threshold for the acceptable score. To this end, we learn multiple one-versus-all binary SVMs.

In the (4),  $p(C | P^i)$  does not depend on the temporal location  $t$  of  $P^i$ , which is not necessarily true. However, we do this approximation, which may give false positives, only for generating more hypotheses. The hypothesis evaluation will be treated in the next section, where the order is considered via the priors and transitions between primitives.

In *testing*, we vote in the 3D space using (4) and then we apply the SVMs in a sliding time window to get the putative segments. As may happen in many cases, the local maxima in the Hough parameter space may be the result of noise and thus may not correspond to a real segment. Therefore, an additional evaluation step is normally applied to eliminate some false positives using an HMM-like likelihood function, which is learned by training a standard HMM. An illustrative example of the proposed hypothesis generation process and the additional evaluation step is shown in Fig. 3(b) and (c) in the context of the proposed algorithmic steps presented in Fig. 1.

#### B. Hypothesis Evaluation via Dynamic Programming

The processing described in the previous section results in, say,  $G$  putative segments;  $G$  is many orders of magnitude smaller than the number resulting from exhaustive consideration of all possible combinations of classes and begin–end points. However, these  $G$  segments are typically overlapping and belong to different classes. Their possible combinations are  $O(G!)$ , which according to our experimental observations could still be a computational bottleneck even if we use evolutionary algorithms for fast estimation. We also observed

$$A(C_{t-1}, C_t) = \begin{cases} \sum_{i=1}^M \sum_{j=1}^M p_{tr}(P_{t-1}^i, P_t^j | C_{t-1}) \cdot p(\mathbf{f}_{t-1} | P_{t-1}^i, C_{t-1}) p(\mathbf{f}_t | P_t^j, C_t) & \text{if } C_t = C_{t-1} \\ \sum_{i=1}^M \sum_{j=1}^M p_e(P_{t-1}^i | C_{t-1}) p(\mathbf{f}_{t-1} | P_{t-1}^i, C_{t-1}) \cdot p_b(P_t^j | C_t) p(\mathbf{f}_t | P_t^j, C_t) & \text{if } C_t \neq C_{t-1}. \end{cases} \quad (8)$$

that in most of the cases, the discriminative framework proposes, among others, segments that are close to the ground truth. These observations motivated an approach that seeks to consider *only the proposed segments* to explain the sequence of measurement vectors on the timeline. If for parts of the timeline there are no putative segments, these parts remain unassigned and account for unknown observations.

We merge the putative segments that overlap and have the same label, but typically there are also overlaps between segments of different labels, which compete for the same time windows. Assuming only one label for each time slot, we propose a variation of the Viterbi algorithm for linear cost label assignment with regard to the number of input frames.

We define the likelihood  $\delta_t$ , which is calculated after the optimal labeling of time instances. The optimal sequence of labels for a time instance  $t = 1, \dots, T$ , which is covered by overlapping putative segments of different labels, is given by the path  $\psi_T = C_1, C_2, \dots, C_T$ , where  $C_t$  is the assigned label for the  $t$ th time instance taking values from a subset of  $1, \dots, L$ , by considering only the labels of the putative segments that cover the time segment. If the time segment is not covered by any putative segment, then it remains unlabeled. The initialization of the likelihood  $\delta_t$  for  $t = 1$  is then given by

$$\delta_1(C_1) = \sum_{i=1}^M p(\mathbf{f}_1 | P^i) \cdot p_b(P^i | C_1) \quad (6)$$

where  $p_b(P^i | C_1)$  denotes the prior probability that the action  $C_1$  begins with primitive  $P^i$ .

At time  $t$ , which accounts for the first  $t$  time instances, we get a recursive expression

$$\delta_t(C_t) = \max_{C_{t-1}} \{ \delta_{t-1}(C_{t-1}) \cdot A(C_{t-1}, C_t) \} \cdot \sum_{i=1}^M p(\mathbf{f}_t | P^i) p(P^i | C_t) \quad (7)$$

where the sum expresses the likelihood of  $p(\mathbf{f}_t | C_t)$ ,  $\delta_{t-1}(C_{t-1})$  is the recursive factor (typical for dynamic programming algorithms), and  $A(C_{t-1}, C_t)$  accounts for switching from action label  $C_{t-1}$  to  $C_t$ , further analyzed in Equation (8), as shown at the top of this page, treats separately the case of switching between different actions from  $t-1$  to  $t$  (handled in the second branch) from the case that two consecutive primitives belong to the same action (handled in the first branch). In the first case, the transition likelihood  $p_{tr}(P_{t-1}^i, P_t^j | C_{t-1})$  is involved, while the label remains the same ( $C_{t-1}$ ). In the times  $t-1$  and  $t$ , we observe the primitives  $P^i$  and  $P^j$ , respectively, as indicated

by the subscripts. There is no hard assignment of primitives, so all transitions are evaluated and weighted by the respective observation likelihoods.

The second branch accounts for switching between different labels, so the termination likelihood  $p_e$  is involved for label  $C_{t-1}$ . A new segment with a different label begins at  $t$ , so the begin likelihood  $p_b$  for the new class label  $C_t$  is involved. These are weighted by the respective observation likelihoods for the given classes  $C_t$  and  $C_{t-1}$  and the respective primitives. Again, all possible combinations are evaluated.

At this point, we should note that  $p_{tr}$ ,  $p_b$ , and  $p_e$  are learned by a standard EM-learning procedure for HMMs similar to [34]. We trained one HMM for each action.  $p_{tr}$  corresponds to the transition matrix, and  $p_b$  and  $p_e$  correspond to the priors for the first and last primitives (states) of the action, respectively.

In (7), only the rightmost factor needs to be evaluated for each class, while the left part is already available from the  $t-1$  step. The overall cost of the calculation is linear and is appropriate for online applications. To retrieve the sequence of assignments, we keep track of the argument that maximized (7) through the array  $\psi_t$ , which is given by

$$\psi_t(C_t) = \arg \max_{C_{t-1}=1..L} \{ \delta_{t-1}(C_{t-1}) \cdot A(C_{t-1}, C_t) \}. \quad (9)$$

An illustrative example of the hypothesis evaluation process is shown in Fig. 3(d) for the two experimental configurations that are provided, resulting in the final classification and segmentation of actions. The proposed dynamic programming algorithm differs from the typical Viterbi algorithm because the transition between labels on the timeline has to be treated differently in the case that the same segment continues [label remains the same—the first case in (8)] than the case that a new segment begins [label changes—the second case in (8)].

### C. Estimation of Primitives

A crucial part of the proposed methodology is the estimation of action primitives, which actually represent typical poses of the actors during the action or gesture. A highly discriminative primitive is one that is associated only with specific actions (labels), so that its presence is typical of only those specific actions. Typical estimation methods such as the widely used  $k$ -means or similar methods do not involve such specificity criteria, and therefore, their results may be suboptimal.

One way to define the primitives is to optimize an objective function that has a term that penalizes the appearance of primitives under multiple classes. Here we propose to solve the problem of primitives definition by penalizing primitives that appear in multiple classes and thus are less informative

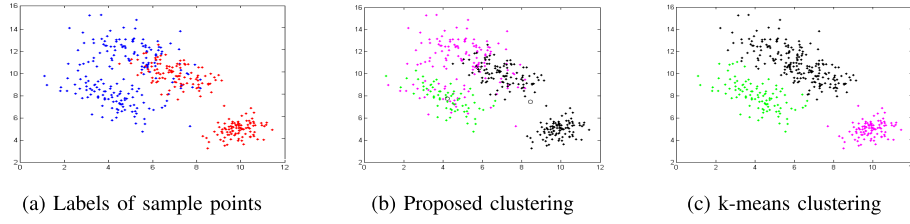


Fig. 2. (a) Demonstration of primitive estimation in comparison with  $k$ -means for  $k = 3$  and two classes. (b) Proposed method favors clusters that are more class specific. The black cluster is mainly composed of elements of the red class and the magenta and green clusters are mainly composed of members of the blue class. (c) In contrast, with  $k$ -means, the black cluster contains elements of both the red and blue classes.

for classification. To this end, we define the modified  $k$ -means algorithm as follows.

Let  $\mathbf{f}_t$ ,  $t = 1, \dots, T$  be the set of  $T$  observations. Let  $f_{tj}$  denote the  $j$ th feature of  $\mathbf{f}_t$ . Define for  $t = 1, \dots, T$  and  $m = 1, \dots, M$

$$u_{tm} = \begin{cases} 1, & \text{if the } t\text{-th observation belongs} \\ & \text{to the } m\text{-th cluster} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$v_{tl} = \begin{cases} 1, & \text{if the } t\text{-th observation belongs} \\ & \text{to the } l\text{-th class} \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Then for the matrices  $\mathbf{U} = [u_{tm}]$ ,  $\mathbf{V} = [v_{tl}]$  holds

$$u_{tm} \in \{0, 1\} \quad \text{and} \quad \sum_{m=1}^M u_{tm} = 1 \quad (12)$$

$$v_{tl} \in \{0, 1\} \quad \text{and} \quad \sum_{l=1}^L v_{tl} = 1. \quad (13)$$

The matrix  $\mathbf{V}$  is constant (defined by the label of each sample) and we seek to find the optimal matrix  $\mathbf{U}$ .

The centroid of the  $m$ th cluster is calculated by

$$c_{mj} = \frac{\sum_{t=1}^T u_{tm} f_{tj}}{\sum_{t=1}^T u_{tm}}. \quad (14)$$

We seek to find the optimal values for  $\mathbf{U}$  that minimize the objective function as follows:

$$\arg \min_{\mathbf{U}} [g(\mathbf{U}, \mathbf{F}, \mathbf{C}) - \lambda \cdot h(\mathbf{U}, \mathbf{V})] \quad (15)$$

where

$$g(\mathbf{U}, \mathbf{F}, \mathbf{C}) = \frac{1}{MT} \sum_{m=1}^M \sum_{t=1}^T u_{tm} \sum_{j=1}^d (f_{tj} - c_{mj})^2 \quad (16)$$

and

$$h(\mathbf{U}, \mathbf{V}) = \sum_{m=1}^M \sum_{l=1}^L \frac{\sum_{t=1}^T u_{tm} v_{tl}}{\sum_{t=1}^T u_{tm}} \cdot \log \left( \frac{\sum_{t=1}^T u_{tm} v_{tl}}{\sum_{t=1}^T u_{tm}} \right) \quad (17)$$

while  $\lambda$  weighs the contribution of the  $h$  factor (either user defined or obtained by cross validation),  $\mathbf{C}$  is the  $M \times d$  centroid matrix, and  $\mathbf{F}$  is the  $T \times d$  observation matrix. The first factor accounts for the within cluster variation, which

is a standard metric to minimize in the  $k$ -means framework (see [35]). The second factor represents the entropy of the distribution of primitives to different classes. If we define as  $p(P^m|l)$  the likelihood that a primitive with index  $m$  will be part of action with label  $l$ , then the entropy is given by

$$- \sum_{m=1}^M \sum_{l=1}^L p(P^m|l) \cdot \log(p(P^m|l)) \quad (18)$$

where

$$p(P^m|l) = \frac{\sum_{t=1}^T u_{tm} v_{tl}}{\sum_{t=1}^T u_{tm}}. \quad (19)$$

Clearly, the entropy is a measure that can quantify the primitive's specificity. It is maximized when  $p(P^m|l) = 1/L$  for all labels  $l$  (equal probability therefore minimal specificity), while it becomes zero for  $p(P^m|l_0) = 1$  and  $p(P^m|l) = 0$  for  $l \neq l_0$  (maximum specificity).

The optimization problem can be solved using one of the many available methods for discrete optimization such as a baseline genetic algorithm (GA) method [35] or some of the many discrete particle swarm optimization variations like those in [24] or [36]. The solution vector (chromosome) is of dimension equal to  $T$ , where each cell takes a value between 1 and  $M$ .

Fig. 2 presents a simple qualitative example in 2D, to give an intuition on how the proposed method clusters together elements of the same class, thus leading in most cases to more discriminative primitives. In the following section, we report the quantitative results about the effectiveness of the proposed technique and the gain toward the overall performance of our framework.

#### IV. EXPERIMENTAL RESULTS

To verify the validity of our framework, we have experimented with synthetic as well as various real data sets from the field of visual action recognition. For our comparisons, we implemented (using the CVX [37]) the optimization scheme that maximizes the confidence for segments similarly to [2] [hereafter denoted by MaxConfidence (MC)] and the scheme that maximizes the overall score similarly to [1] [hereafter denoted by MaxScore (MS)]; these are state-of-the-art methods that do online segmentation, like does our method.



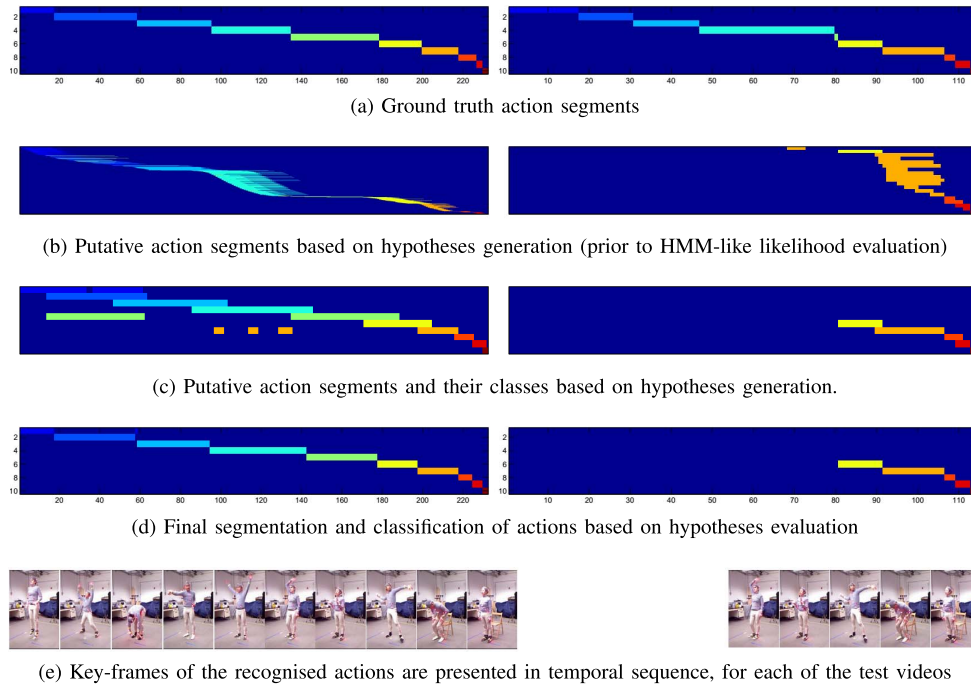


Fig. 3. Two sample results from the Berkeley data set (one example per column). (a)–(d) Actions 1–10 are illustrated as color-coded segments. The horizontal axis represents the timeline in frames. On the left column example, modeling and training of the proposed method is applied using all actions 1–10, performed by subjects 1–7. Testing is applied on an image sequence that contains each action once performed by subjects 8–12. The final result is illustrated in the left subfigure of (d), matching the ground truth in (a). On the right column example, training using only actions 6–10 for subjects 8–12 was performed (the actions 1–5 are unknown). A test sequence was compiled concatenating all available actions once, from subjects 1–7. The final labeling, shown in right subfigure of (d), demonstrates the segmentation and classification of the modeled actions in the context of the unmodeled actions 1–5, which are considered as unknown in that case, and thus no recognition results are present during the first five actions. (e) Corresponding key frames of the recognized actions are illustrated. (The figure is best viewed in color.)

#### A. Synthetic Data

We generated a data set of 2D data sequences. We created randomly ten HMM models, each composed of up to three different states, by random definition of means, covariances, priors, and state transitions. Then for each of them, we performed sampling and we produced 100 sequences of length between 450 and 750 each. These sequences were concatenated at random order to form bigger sequences consisting of one instance per class.

Given the data set, we investigated two different settings. To make our method comparable to the existing work, we initially made the assumption of a multiclass problem, where all the knowledge was given in advance to the system, i.e., no instances stemming from unknown latent classes appeared. This implies that a label from the known set of labels had to be assigned to every frame. We used 50% of the sequences for training and the rest for testing.

Fig. 4 presents per-class classification accuracies of our method, MC and MS, on a frame-by-frame basis. The size of the sliding window was defined by the maximum action length. Our method shows a similar or higher accuracy, i.e., 89.44% versus 87.65% and 81.71% of MC and MS, respectively. Our method is agnostic to the existence of instances stemming from unknown sequential patterns; therefore, small gaps falsely assigned to novel observations may appear. This is the largest source of error, i.e., segments that have actually larger duration are detected as shorter because the longer ones are sometimes not suggested as putative segments. On the contrary,

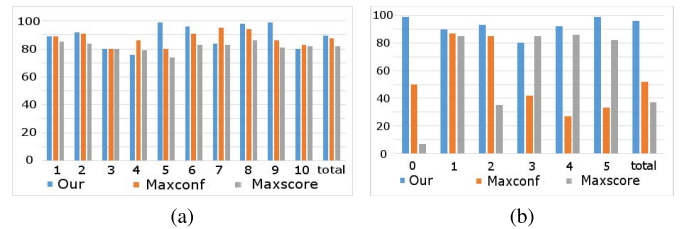


Fig. 4. Classification accuracy (%) per class for the synthetic experiment. Class 0 denotes the novel observations. (a) Training with full data set. (b) Training with subset.

MC and MS assume that all observations are known and that there are no gaps between the putative segments.

Next, we assumed that some of the observed actions resulted from unknown latent classes (hereafter denoted with the label 0). MC and MS are not able to classify instances of previously untrained actions; therefore, to make a fair comparison, we trained an HMM for each known action and we checked the likelihood of each segmented subsequence using the respective model. For low likelihoods, we classified the actions as unknown. We excluded the instances of five classes from training and we learned the rest.

Our method gave promising results, exhibiting accuracy 96.16% (assignment of instances from 6–10 to 0 were considered true). The best results for MC and MS were 52.42% and 37.34%, respectively, and were obtained using a length-normalized threshold of  $10^{-7}$  and  $M = 30$ . Their inferior

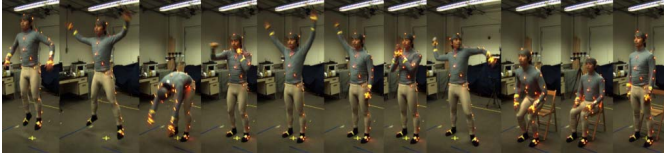


Fig. 5. Actions in MHAD. *Jumping, Jumping jacks, Bending, Punching, Waving two hands, Waving one hand, Clapping, Throwing, Sit down/stand up, Sit down, and Stand up.* The images were taken from [38].

performance can probably be attributed to the requirement for continuous labeling of the whole timeline, which inevitably couples together the actions and necessitates a suboptimal postprocessing step to detect unseen actions; in contrast, our method is optimized to classify known and detect unknown actions without that constraint.

### B. Berkeley Multimodal Human Action Database

The next experiment is related to visual recognition of actions, which involve the whole human body. To this end, we use the Berkeley Multimodal Human Action Database (MHAD) [38], which consists of temporally synchronized and geometrically calibrated data. The data set contains 11 actions performed by 13 subjects: *Jumping, Jumping jacks, Bending, Punching, Waving two hands, Waving one hand, Clapping, Throwing, Sit down/stand up, Sit down, and Stand up*, as illustrated in Fig. 5.

The data set contains about 660 action sequences, which correspond to about 82 min. In the original data set, the different actions were provided as segments. For the purpose of identifying actions in continuous data, we concatenated those videos. We did not consider the action *Sit down/stand up* as a separate action, but as the composition of the actions *Sit down* and *Stand up*; this approach is justified by the continuous recognition that we do. Therefore, we actually classified only ten different actions.

All available 3D skeleton joints were used to build our representation of human motion for each frame. We used the 3D orientation of each joint with respect to its ancestor in the kinematic skeletal chain based on quaternions. In addition, the 3D coordinates of each skeletal joint with respect to the hips joint were computed for each mocap record. Finally, the distance of the hips joint to the ground plane was also incorporated to the feature set for each frame. The last two subsets of our feature representation were normalized for each actor of the data set, given the total length of the skeletal chain. The feature vector had 220 dimensions 103 representing all the joint angles and 117 representing all the joint positions.

We trained with the first seven subjects and tested with the last five ones as in [38]. Each subject repeated the same actions four times. Fig. 6 gives the per-class accuracy. Our method gave the overall accuracy 92.17% compared with 76.22% of MC and 82.46% of MS when training with all classes. For reference purposes, we mention that the best results reported in [38] using the same skeletal data were 79.93% using kernel-SVM with the  $\chi^2$  kernel for classification. However, the setting was different, i.e., unlike ours, the classification

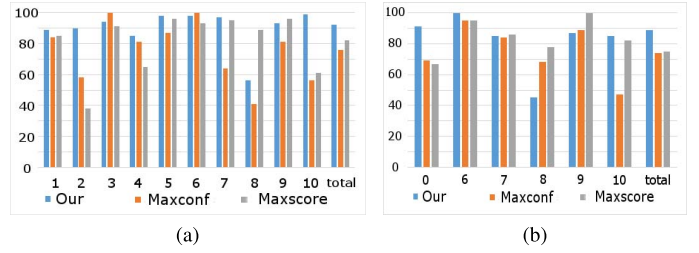


Fig. 6. Classification accuracy (%) per class for the Berkeley experiment. Class 0 denotes the novel observations. (a) Training with full data set. (b) Training with subset.

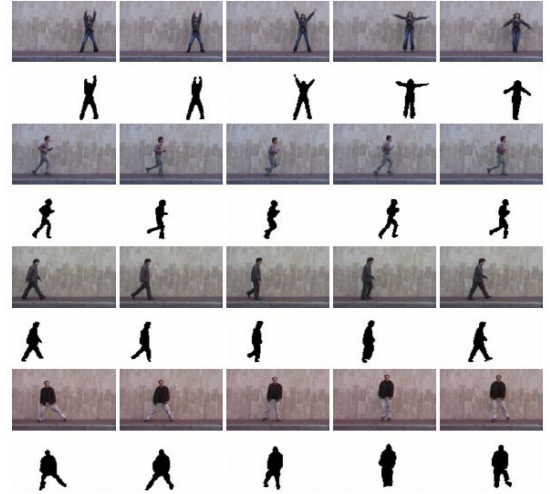


Fig. 7. Examples of video sequences and extracted silhouettes from the Weizmann database (from [39]) showing *Waving two hands, Running, Walking, and Galloping sideways*.

was performed on segmented videos and the *Sit down/stand up* action was treated separately.

We then examined the effect of unknown sequences, using the same postprocessing step as in the synthetic experiment to make the competing methods comparable to ours. We trained with classes 6–10 and then tested using the same data as in the previous experiment. The illustrative experimental results are demonstrated in Fig. 6. Our method had an overall accuracy of 89.04%, outperforming MC with 74.25% and MS with 75.14% (threshold  $10^{-5}$ ,  $M = 30$ ), which verifies the merit of the proposed method.

### C. Weizmann Data Set

We also used the classification database of the Weizmann data set [39]. Fox *et al.* [8] collected a database of 81 low-resolution ( $180 \times 144$ , 25 frames/s) video sequences showing nine different people, each performing nine natural actions such as *Running, Walking, Jumpingjack, Jumping forward on two legs, Jumping in place on two legs, Galloping sideways, Waving two hands, Waving one hand, and Bending* (see Fig. 7).

Based on the aligned binary foreground masks that are also available in the data set, we computed the Zernike moments up to order 20 [40]. We exploited only the even polynomials (excluding zero-based polynomials) and concatenated



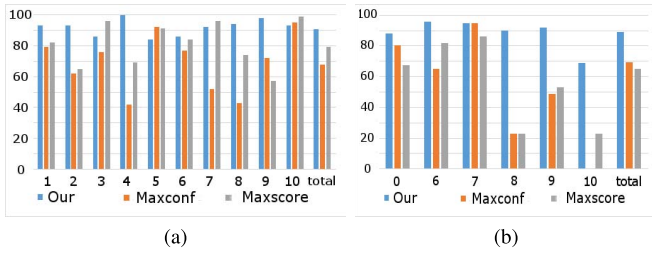


Fig. 8. Classification accuracy (%) per class for the Weizmann experiment. Class 0 denotes the novel observations. (a) Training with full data set. (b) Training with subset.



Fig. 9. TUM kitchen data set [41]. Example images of various actions performed by different subjects and viewpoints.

the resulting amplitude and phase values resulting in feature vectors of size 220. In the following, we applied a GMM-based clustering to build a dictionary of  $M = 30$  motion primitives that represent the notion of spatiotemporal keywords in our framework. As in [2], we concatenated each set of videos of a subject into a single longer sequence; consequently, we composed nine long videos of contiguous actions and used five of the videos for training and four for testing.

The results per class are illustrated in Fig. 8. We used a threshold of  $10^{-5}$  and  $M = 30$ . The training with all data yielded 90.82%, 67.64%, and 79.31% for our method, MC, and MS, respectively; the partial training with classes 1–5 yielded 89.23%, 69.34%, and 65.14% for the three methods, respectively. For reference purposes, we mention that in [2], 94% accuracy was reported for the MC, which is justified by the use of spatiotemporal features, while in our experiments, we used only spatial features.

#### D. TUM Kitchen Data Set

The next experiment was related to the TUM Kitchen Multimodal data set [41] (see Fig. 9). It consists of a collection of activity sequences recorded in a kitchen environment equipped with multiple complementary sensors, such as color cameras, RFID, motion capture, and magnetic data. The recorded actions regard naturally performed manipulation tasks as encountered in everyday activities of human life. Several instances of a table-setting task were performed by different subjects, involving the manipulation of objects and the environment (e.g., *Lowering an object*, *Releasing grasp of an object*, *Opening a drawer*, etc.). We treat each episode of the data set as sample for training or testing. An episode consists of a sequence of frames showing a single subject performing various daily manipulation tasks in a continuous manner. Our framework acquires as input the available motion capture data for each episode.

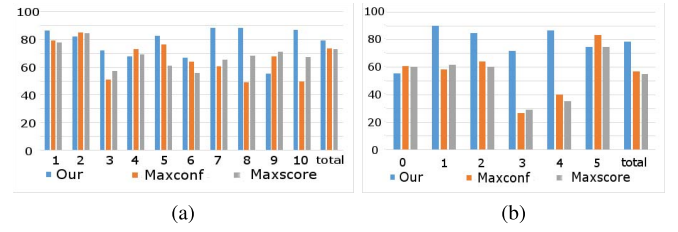


Fig. 10. Classification accuracy (%) per class for the TUM Kitchen experiment when training (a) with the full data set and (b) with a subset of the first five classes. Class 0 denotes the novel observations.

We processed the streams of skeletal (motion capture) data of five subjects performing actions from the data set, following the same experimental protocol as in [31], and thus we split the available video sequences into training and testing parts. More specifically, we used 0–2, 0–8, 0–4, 0–6, 0–10, 0–11, and 1–7 for testing and the remaining 12 episodes of the data set for training.

Moreover, we split the idle/carry class according to whether the subject was walking or standing, as also performed in [42]. We further employed the associated action labels for the left hand, resulting in ten different labels for the ground truth, namely, *Reach Up*, *Take Object*, *Lower Object*, *Release Grasp*, *Open Door*, *Close Door*, *Open Drawer*, *Close Drawer*, *Carrying/Walk*, and *Carrying/Still*.

We produced the same single articulation features based on the full set of skeletal joints provided per frame for the data set, as in [3], in order to maintain consistency and produce comparable quantitative results. The single articulation feature corresponded to the 3D position of one articulation that had been quantized independently for each articulation, using the  $k$ -means algorithm. The same value for  $k$  was considered for all articulations of the body model ( $k = 10$ ). We used  $M = 30$  primitives. Performances were measured by the total number of correct predictions over the total number of frames in testing videos, which is in accordance with the rest of the reported results in the literature.

The per-class results are given in Fig. 10. Our method gave the overall accuracy 79.32% compared with 73.35% and 72.99% of the MC and MS.

Furthermore, compared with the results in the related literature for reference, [3] gave 83.0% using 27 joints and more sophisticated features (poses, speeds, and tracklets) than we use. The same paper reported 77.60% accuracy using features similar to ours. The work in [31] reported 81.5% using pose-based features, however, solving the easier problem of classifying segmented actions.

In our next experiment, we learned only the first five action classes and we considered the rest as unknown similarly to the experiments with the previous data sets. The overall performance degraded to 78.47% due to the presence of additional unknown actions (classes 6–10 of the original data set). For the MC and MS methods, the accuracies were 56.85% and 55.19%, respectively (with a cutoff threshold of  $10^{-10}$ ).

#### E. UTKAD Data Set

Finally, we conducted experiments on the University of Texas Kinect-Action Dataset (UTKAD) [43] (see Fig. 11).



Fig. 11. Examples of actions from videos of the ten activities in the UTKAD [43].

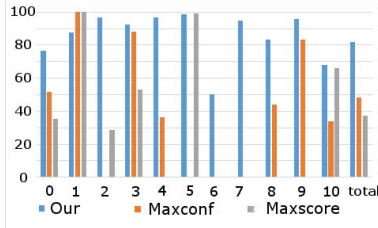


Fig. 12. Classification accuracy (%) per class for the UTKAD experiment. Class 0 denotes the unknown observations.

It is composed of ten indoor activities in a human-machine interaction setting, performed by ten different individuals in varied views: nine males and one female. Each subject performed the set of actions twice, in a continuous manner per sequence. In total, it consists of 200 action samples in 20 video sequences, captured by a single stationary Kinect. The set of actions include *Walk*, *Sit down*, *Stand up*, *Pick up*, *Carry*, *Throw*, *Push*, *Pull*, *Wave*, and *Clap hands*.

As in TUM data set, we processed the streams of skeletal data of the available action sequences following the same experimental protocol as in [3] and [43]. Thus, performances were measured using leave one out cross validation. Moreover, we utilized the single articulation features for single frames, used also in [3] achieving the highest scores, if only joint position data are considered.

Various unlabeled motions performed by the subjects were also interleaved to known actions in the sequences. We considered that configuration as ideal for our methodology, as we aspire to maximize the performance of simultaneous segmentation and classification of actions in the presence of unknown ones. Thus, there was no need to exclude from training any classes, as we did with the previous data sets.

The per-class results are given in Fig. 12. Our method gave the overall accuracy 81.65% compared with 74.8% in [3] (best overall performance in this paper using single articulation features) and 90.9% in [43]. However, the latter methodology and results refer to the task of action classification to already segmented data.

The MC and MS methods were also compared with ours, in combination with an HMM, to label the data as unknown if the normalized log-likelihood of the segment was lower than an experimentally optimized threshold ( $10^{-4.2}$  for MC and  $10^{-4.1}$  for MS). We used  $M = 30$ . The comparison per class is given in Fig. 11. Clearly, the comparison verifies the results

of the previous experiments with our method giving overall superior results 81.65% compared with 48.50% and 37.41% of MC and MS, respectively.

#### F. Computational Cost

The cost of the proposed method can be analyzed as follows. We initially group the observation vectors to primitives, eventually after low-pass filtering, depending on their distance from the associated primitive centers obtained from  $k$ -means. Consecutive observations that are close to the same center count as a single primitive. For each time window, we sum up the votes of all primitives, which has a linear cost to the number of primitives. Assuming that  $n$  primitives fit in the window, and given that the starting primitive of the segment is always the first window primitive, the primitive that ends the segment must be selected from the  $n - 1$  remaining ones. Each of the  $n - 1$  starting-ending primitive combinations represents an  $\mathbf{x}$  and is evaluated against  $C$  action classes, so that the  $A(\mathbf{x}, C)$  is calculated according to (4). For each  $A(\mathbf{x}, C)$ , the classification result is calculated using an SVM, which gives the putative segments. Therefore, the cost for each window is  $\sim O((n - 1) \times C)$  or  $\sim O(n \times C)$ . After the putative segments are calculated, the dynamic programming algorithm verifies the hypotheses with cost that is linear to the total number of primitives, i.e., linear to the number of observations.

MS and MC start from the current observation and scan a time window forward in time to find, within that window, the end point of the segment that gives the best score and the best confidence, respectively, using an SVM as well. A dynamic programming algorithm is used to proceed to the following segments. The cost for each window is  $\sim O(n \times C)$ , where  $n$  is the size of the window. Additional postprocessing steps are necessary to locate previously unseen actions, as we will mention in the following section. The total cost is linear to the number of frames.

MS and MC may use simpler segment representations, e.g., average of pseudoprobabilities of observations with respect to feature centers (after  $k$ -means). However, the gain in efficiency is not significant and the overall cost is still linear to the amount of observations.

The method [3] has a higher cost per window that is  $\sim O(n \times C \times (\text{number of frames}))$ . This is a consequence of the used voting function, which requires voting per frame. On the other hand, our method has to maintain a 3D voting space instead of a 2D space used in [3] and we have to execute an additional linear cost dynamic programming algorithm.

#### V. DISCUSSION

We have applied our method on synthetically generated, as well as on natural visual action streams using four different publicly available data sets (Berkeley MHAD, Weizmann, TUM Kitchen, and UTKAD), to show the merit of the method on different data modalities. The synthetic data set was the simplest to classify with only two dimensions. In Weizmann, we used the moments of the foreground regions, while the Berkeley MHAD, TUM Kitchen and UTKAD used higher dimensional skeletal joints with or without mapping using per-joint clustering.

TABLE I

OUR METHOD AND RESULTS COMPARED WITH PUBLISHED RESULTS ON SYNTHETIC, MHAD, TUM, UTKAD, AND WEIZMANN DATA SETS, ASSUMING TRAINING ON ALL ACTIONS. THE FEATURES USED ARE THE SAME AS IN OUR METHOD UNLESS OTHERWISE STATED. WE ALSO NOTE THE METHODS THAT WE IMPLEMENTED OURSELVES

Algorithm	synthetic	MHAD	TUM (27 Joints)	UTKAD	Weizmann
[31] HF segmented actions pose-based features	N/A	N/A	81.50%	N/A	N/A
[3] DOHT pose+ speed+tracklets based features	N/A	N/A	83.00%	N/A	N/A
[3] DOHT	N/A	N/A	77.60%	74.80%	N/A
[2] MC our implementation	87.65%	76.22%	73.35%	48.50%	69.64%
[2] MC segmented actions spatiotemporal features	N/A	N/A	N/A	N/A	94%
[1] MS our implementation	81.71%	82.46	72.99%	37.41%	79.31%
[43] HOJ3D segmented actions	N/A	N/A	N/A	90.90%	N/A
[38] segmented videos	N/A	79.93%	N/A	N/A	N/A
Ours	89.44%	92.17%	79.32%	81.65%	90.82%

In general, the performance of our method was comparable or better than the competing discriminative methods when a closed world was assumed. Furthermore, when the actions of interest were interrupted by previously unseen actions, our method was still able to classify them and detect the unknown ones in most of the cases. MS and MC gave inferior performance mainly due to the fact that they enforce continuity between actions. The inevitable threshold-based postprocessing proved to be relatively ineffective. Despite our exhaustive search for a threshold value that would give consistent results, these methods were unable to accurately recognize the instances of the novel classes.

To assess the overall performance of our algorithm given the state-of-the-art approaches, we summarize the experimental results for the synthetic, MHAD, Weizmann, TUM Kitchen, and UTKAD in Table I for the given closed world assumption. The results are not always comparable due to the different employed features and due to the fact that many of these works aim to label videos containing single actions, which is much simpler. However, our method demonstrated a similar or better performance for all the cases that the same experimental protocol and the same features were used. We need to note here that these results are with our method set to recognize one additional class (unknown) and that we did not execute any postprocessing step to assign the instances classified as unknown to any of the known classes.

In comparison with [3], which uses probabilistic voting functions, we use a parameter space of higher dimension (three instead of two), but we alleviate that by avoiding the optimization of the voting function. This strategy works well in practice since several hypotheses (putative segments) seem to be concentrated around the true positive segment (see Fig. 3). The quantitative results seem to be advantageous for our method in two different data sets (TUM and UTKAD); however, the synergies between the two approaches could be further investigated in the future.

TABLE II

SUMMARIZATION OF EXPERIMENTS ON THE SYNTHETIC, MHAD, TUM, UTKAD, AND WEIZMANN DATA SETS ASSUMING THAT SOME ACTIONS ARE UNKNOWN

Algorithm	synthetic	MHAD	TUM (27 Joints)	UTKAD	Weizmann
[2] MC (our implementation)	52.42%	74.25%	69.34%	56.85%	48.50%
[1] MS (our implementation)	37.34%	75.14%	65.14%	55.19%	37.41%
Ours	96.16%	89.04%	89.23%	78.47%	81.65%

TABLE III

OVERALL CLASSIFICATION ACCURACY OF OUR METHOD FOR THE MHAD, TUM, UTKAD, AND WEIZMANN DATA SETS. WE COMPARE THE RESULTS OF THE GA-BASED PRIMITIVE DEFINITION AS DEFINED IN SECTION III-C WITH THOSE OF THE STANDARD  $k$ -MEANS-BASED DEFINITION

Method	MHAD	TUM	UTKAD	Weizmann
GA-based primitive definition	92.17%	79.32%	81.65%	90.82%
$k$ -means based primitive definition	87.12%	69.15%	65.83%	64.90%

A key aspect of our method was the appropriate selection of action primitives, which have to be defined in such a way that will be able to differentiate between different classes. The objective function proposed in (15) was employed in a genetic framework and gave discriminative action primitives. That improved accuracy compared with a typical  $k$ -means algorithm, which uses random initialization. In Table III, we present the results using the primitives obtained using this method in comparison with some random  $k$ -means initialization to show the improvement. A baseline GA was used with mutation function that replaced only one solution element with probability 0.9 and a crossover function that combined two solution vectors using a random split with probability 0.1 (other discrete optimization methods could also have been employed). Degenerate cases of very simple actions consisting of single action primitives needed special treatment, as the solution is obvious and it makes not much sense to use the proposed voting framework for recognizing and segmenting them. In our experiments, we faced this issue when we used relatively small values for  $M$ ; however, the complexity of the activities justified the use of higher values for  $M$ . At this point, we should note that the results obtained in Table III using the  $k$ -means method are not strictly comparable to other methods in Table I using the  $k$ -means as well, such as those in [3], due to the fact that the features are different.

Probably, the most important aspect of the proposed method is our choice to train binary SVMs in the Hough parameter space for learning each available action. Such a treatment allows for the separation between instances of known and unknown classes. The same instance can be concurrently negative for all known classes, thus indicating an observation belonging to a novel class. An alternative treatment would employ a multiclass discriminative classifier in combination with a generative model. This is how we used the segments returned by MS and MC, which resulted from a multiclass maximum margin optimization; subsequently, and due to the

TABLE IV  
CLASSIFICATION ACCURACY ON GESTURE CLIPS  
(TAKEN FROM THE CHALEARN DATA SET [44])

Method	vieniqui	cheduepalle	combinato
HMM	70%	20%	100%
HMM on putative segments	90%	90%	100%

fact that the typical Viterbi algorithm is not applicable in the presence of unknown actions, we input those segments into an HMM-based generative model to identify known-unknown actions using a threshold. As presented in Table II, which summarizes our experiments with unknown classes, our method consistently outperformed MS and MC by a quite large margin, no matter what the threshold value was.

Our method could also be beneficial in the context of classification of whole clips, especially when these clips contain irrelevant actions. This could be done by identifying the relevant actions as putative segments and by excluding the rest. We did a preliminary experiment to verify this insight using a small part of the Chalearn Gesture data set [44], using as feature the Euclidean distances between the skeletal joints. We used 15 clips of the gestures vieniqui, cheduepalle, and combinato to train an HMM classifier and ten different sample clips per gesture for testing. The test clips were containing parts of other gestures as well as the idle state, which had been excluded from the training set. By finding the timeline covered by the putative segments, we were able to segment out a lot of the irrelevant unknown actions. The classification results before and after the rejection of these unknown actions are given in Table IV. Segmenting out the unknown parts improved the results, which shows that our method could be potentially useful in the context of whole clip classification. That is a direction that we wish to pursue further in the future.

## VI. CONCLUSION

We presented a framework for online simultaneous segmentation and classification of sequential data interrupted by unknown actions. The method uses action primitives, which are associated with actions. Each motion primitive votes in a 3D Hough space for begin-end and type of action, and then the votes in the 3D space are evaluated per possible action using a binary SVM. The final results are obtained using a dynamic programming framework.

The proposed framework gave an accuracy that was similar or better compared with those of other discriminative methods for online segmentation and classification in the case that all actions are known. Furthermore, it was able to recognize actions belonging to previously modeled classes in the context of other unknown activities. To our knowledge, this is the first discriminative method for online simultaneous segmentation and classification that has been demonstrated to have this property.

In the future, we plan to test our method on big data, including segmented clips, which involve learning representations for large numbers of classes and to employ more elaborate features, e.g., from deep learning or derived from improved

dense trajectories. Alternative learning methods like random forests could be also evaluated in the future and compared in terms of speed and accuracy with our current discriminative model.

## REFERENCES

- [1] Q. Shi, L. Wang, L. Cheng, and A. Smola, "Discriminative human action segmentation and recognition using semi-Markov model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
- [2] M. Hoai, Z.-Z. Lan, and F. De la Torre, "Joint segmentation and classification of human actions in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3265–3272.
- [3] A. Chan-Hon-Tong, C. Achard, and L. Lucat, "Simultaneous segmentation and classification of human actions in video streams using deeply optimized Hough transform," *Pattern Recognit.*, vol. 44, no. 12, pp. 3807–3818, 2014.
- [4] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
- [5] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1250–1257.
- [6] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2061–2068.
- [7] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2188–2202, Nov. 2011.
- [8] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "A sticky HDP-HMM with application to speaker diarization," *Ann. Appl. Statist.*, vol. 5, no. 2A, pp. 1020–1056, 2011.
- [9] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, Jul. 1998.
- [10] N. Oliver, A. Garg, and E. Horvitz, "Layered representations for learning and inferring office activity from multiple sensory channels," *Comput. Vis. Image Understand.*, vol. 96, no. 2, pp. 163–180, Nov. 2004.
- [11] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-Markov model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 838–845.
- [12] P. Natarajan and R. Nevatia, "Coupled hidden semi Markov models for activity recognition," in *Proc. IEEE Workshop Motion Video Comput. (WMVC)*, Feb. 2007, p. 10.
- [13] M. Reyes, G. Domínguez, and S. Escalera, "Feature weighting in dynamic time warping for gesture recognition in depth data," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 1182–1188.
- [14] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1685–1699, Sep. 2009.
- [15] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware activity recognition and anomaly detection in video," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 1, pp. 91–101, Feb. 2013.
- [16] H. Izadinia and M. Shah, "Recognizing complex events using large margin joint low-level event model," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 430–444.
- [17] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
- [18] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng, "Tiled convolutional neural networks," in *Proc. NIPS*, 2010, pp. 1279–1287.
- [19] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [20] S. P. Chatzis, D. I. Kosmopoulos, and P. Doliotis, "A conditional random field-based model for joint sequence segmentation and classification," *Pattern Recognit.*, vol. 46, pp. 1569–1578, Jun. 2013.
- [21] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Conditional models for contextual human motion recognition," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 210–220, 2006.

- [22] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848–1852, Oct. 2007.
- [23] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 3, Jul. 2003, pp. 1741–1745.
- [24] H.-J. Lee and S. J. Roberts, "On-line novelty detection using the Kalman filter and extreme value theory," in *Proc. 19th IEEE ICPR*, Dec. 2008, pp. 1–4.
- [25] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [26] K. G. Derpanis, M. Sizintsev, K. Cannons, and R. P. Wildes, "Efficient action spotting based on a spacetime oriented structure representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1990–1997.
- [27] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. ECCV Workshop Statist. Learn. Comput. Vis.*, 2004, pp. 17–32.
- [28] S. Maji and J. Malik, "Object detection using a max-margin Hough transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 1038–1045.
- [29] Y. Zhang and T. Chen, "Implicit shape kernel for discriminative learning of the Hough transform detector," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 1–11.
- [30] P. Wohlhart, S. Schulter, M. Köstinger, P. Roth, and H. Bischof, "Discriminative Hough forests for object detection," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 40.1–40.11.
- [31] A. Yao, J. Gall, G. Fanelli, and L. V. Gool, "Does human action recognition benefit from pose estimation?" in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 67.1–67.11. [Online]. Available: <http://dx.doi.org/10.5244/C.25.67>
- [32] T.-H. Yu, T.-K. Kim, and R. Cipolla, "Unconstrained monocular 3D human pose estimation by action detection and cross-modality regression forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3642–3649.
- [33] D. Kosmopoulos, K. Papoutsakis, and A. Argyros, "Online segmentation and classification of modeled actions performed in the context of unmodeled ones," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.
- [34] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Readings in Speech Recognition*, A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1990, pp. 267–296.
- [35] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 3, pp. 433–439, Jun. 1999.
- [36] A. H. Kashan and B. Karimi, "A discrete particle swarm optimization algorithm for scheduling parallel machines," *Comput. Ind. Eng.*, vol. 56, no. 1, pp. 216–223, 2009.
- [37] M. C. Grant and S. P. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (Lecture Notes in Control and Information Sciences), V. D. Blondel, S. P. Boyd, and H. Kimura, Eds. London, U.K.: Springer, 2008, pp. 95–110.
- [38] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley MHAD: A comprehensive multimodal human action database," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2013, pp. 53–60.
- [39] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
- [40] R. Mukundan and K. R. Ramakrishnan, *Moment Functions in Image Analysis: Theory and Applications*. New York, NY, USA: World Scientific, 1998.
- [41] M. Tenorth, J. Bandouch, and M. Beetz, "The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep./Oct. 2009, pp. 1089–1096.
- [42] J. Gall, A. Yao, and L. Van Gool, "2D action recognition serves 3D human pose estimation," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 6313, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010, pp. 425–438.
- [43] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2012, pp. 20–27.
- [44] S. Escalera *et al.*, "ChaLearn looking at dataset and results," in *ECCV Workshops, Part I (LNCS)*, vol. 8925, L. Agapito *et al.* Eds. Springer, 2015, pp. 459–473, doi: 10.1007/978-3-319-16178-532.



**Dimitrios Kosmopoulos** received the B.Eng. degree in electrical and computer engineering and the Ph.D. degree from the National Technical University of Athens, Athens, Greece, in 1997 and 2002, respectively.

He was with the Computational Biomedicine Imaging and Modeling Laboratory, Computer Science Department, Rutgers University, Brunswick, NJ, USA, and the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, USA. He was with the Institute of Computer Science, Foundation for Research and Technology–Hellas, Heraklion, Greece, and the Department of Informatics Engineering, Technical Educational Institute of Crete, Heraklion, Greece. He was also with the Department of Electrical and Computer Engineering, National Technical University of Athens. He was a Researcher and Developer for various companies and institutions. He has been a Research Scientist with the Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Centre of Scientific Research Demokritos, Aghia Paraskevi, Greece. He is currently an Assistant Professor with the Department of Cultural Heritage and New Technologies, University of Patras, Agrinio, Greece. He has authored over 90 papers in peer-reviewed scientific journals and conferences.



**Konstantinos Papoutsakis** received the Diploma degree in computer engineering and informatics from the University of Patras, Patras, Greece, and the M.Sc. degree in computer science from the University of Crete, Heraklion, Greece, where he is currently pursuing the Ph.D. degree.

He was a Research Assistant with the Institute of Computer Science, Foundation for Research and Technology–Hellas, Heraklion, Greece, under the supervision of Prof. A. Argyros. His current research interests include computer vision, machine learning, and robotics with an emphasis on visual object tracking, human motion analysis, gesture and action recognition, and human robot interaction.



**Antonis Argyros** is currently a Professor of Computer Science with the Computer Science Department, University of Crete, Heraklion, Greece, and a Researcher with the Institute of Computer Science, Foundation for Research and Technology–Hellas, Heraklion, Greece. He is also interested in the applications of computer vision in the fields of robotics and smart environments. In these areas, he has authored over 140 papers in scientific journals and refereed conference proceedings. His current research interests include computer vision with an

emphasis on 2D/3D tracking, human gesture and posture recognition, and 3D reconstruction and omnidirectional vision.

Prof. Argyros is a member of the Strategy Task Group of the European Consortium for Informatics and Mathematics. He served as a General Co-Chair of ECCV 2010, an Area Chair of ECCV 2016, a Co-Organizer of HANDS 2015, and an Associate Editor of the IEEE ICRA 2016 and the IEEE IROS 2016. He also serves as an Area Editor of the *Computer Vision and Image Understanding Journal* and is a member of the Editorial Boards of the *IET Image Processing* and the *IEEE ROBOTICS AND AUTOMATION LETTERS*.