Hankelet-based Dynamical Systems Modeling for 3D Action Recognition

Liliana Lo Presti[1], Marco La Cascia

DICGIM - University of Palermo,
V.le delle Scienze, Ed. 6
90128 Palermo, ITALY

Stan Sclaroff

Computer Science Department - Boston University,
111 Cummington Mall,
02215 Boston, MA, USA

Octavia Camps

Dept. of Electrical and Computer Eng., Northeastern University,
360 Huntington Ave.,
02115 Boston, MA, USA

Abstract

This paper proposes to model an action as the output of a sequence of atomic Linear Time Invariant (LTI) systems. The sequence of LTI systems generating the action is modeled as a Markov chain, where a Hidden Markov Model (HMM) is used to model the transition from one atomic LTI system to another. In turn, the LTI systems are represented in terms of their Hankel matrices. For classification purposes, the parameters of a set of HMMs (one for each action class) are learned via a discriminative approach. This work proposes a novel method to learn the atomic LTI systems from training data, and analyzes in detail the action representation in terms of a sequence of Hankel matrices. Extensive evaluation of the proposed approach on two publicly available datasets demonstrates that the proposed method attains state-of-the-art accuracy in ac-

Email addresses: liliana.lopresti@unipa.it (Liliana Lo Presti), marco.lacascia@unipa.it (Marco La Cascia), sclaroff@bu.edu (Stan Sclaroff), camps@coe.neu.edu (Octavia Camps)

[1] Corresponding author

tion classification from the 3D locations of body joints (skeleton).

Keywords: Hidden Markov Model, Hankel Matrix, Linear Time Invariant System, Discriminative Learning, Action

1. Introduction

In recent years, a large portion of the research in computer vision has focused on the problem of action recognition and modeling. Detection, recognition and analysis of actions is of great interest in several application domains such as surveillance [1], [2], [3], [4], human-computer interaction [5], assistive technologies [6], sign language [7], [8], [9], and, more recently, computational behavioral science [10], [11] and consumer behavior analysis [12].

The wide diffusion of cheap depth cameras, and the seminal work by Shotton, et al. [13] for estimating the locations of the joints of a human body from depth maps, have given new stimulus to the research in 3D action classification both by quickening the development of novel applications, and by providing a setting to test new ideas and frameworks. Therefore, very recently, we have seen a proliferation of works introducing novel body pose representations for action recognition given depth maps and/or skeleton data [14], [15], [16], [17], [18], [19], [20].

In this paper, we propose to represent an action as a series of movements to exploit their temporal structure while discriminating among different action classes. As an example, consider the action of *handshaking* which can be modeled by the following ordered sequence of movements: *moving the whole body to approach the other person, raising the arm, and shaking the hand.* Furthermore, each of these movements can be represented as a sequence of observations (for example a sequence of body poses) which are characterized by their own dynamics. Therefore, an action can be represented in terms of a “sequence of simpler dynamics”.

This reasoning leads to the idea that an action should be modeled by a hierarchical dynamical model, such as a mixture of Hidden Markov Models (HMMs) [21], coordinated mixture of factor analyzers [22] or switching models [23]. However, the burden of learning the model parameters and the size of the required training set may limit the applicability of these methods.

Here, we propose to approximate the abovementioned complex hierarchical dynamical model by adopting a simpler representation for the movements. In particular, we focus our attention on the switching of the dynamics across time. For this purpose, we represent movements using body motion templates. A body motion template may be either an ordered set of trajectories (i.e. trajectories of body parts such as hands, arms, legs, head, torso) or a sequence of frame descriptors (based on bag-of-words, oriented flow, dense trajectories, etc.) within a temporal window. For simplicity, in the remainder of the paper we will assume that a body motion template is an ordered set of trajectories of 3D body joints within a temporal window. However, our framework may be used with other feature representations as long as they have an ordering relation.

Fig. 1 illustrates the basic idea of our approach. An action is a temporal series of body motion templates (movements). Each body motion template is a series of raw observations in a temporal window (eventually of varying duration) which is characterized by a specific dynamic. Thus, we aim at decomposing an action into sub-trajectories that are modeled as the outputs of a sequence of atomic linear time invariant (LTI) systems, using an HMM to model the transitions from one atomic LTI system to another. Furthermore, each body motion template is described by means of a truncated Hankel matrix (Hankelet) [24], which embeds the parameters of the LTI system [25]. In summary, an action is modeled by an HMM where the observations are Hankel matrices, computed in a sliding window, and where each hidden state represents an LTI system for which only a Hankelet is known. Finally, for classification purposes, we train a set of HMMs (one for each action class) using a discriminative approach.

Fig. 2 contrasts traditional HMM representations against our approach. Instead of learning the parameters of a switching HMM (Fig. 2(a)), we consider a probabilistic switching LTI system (Fig. 2(b)) where an HMM is used over the Hankel matrices of the systems, avoiding the need of performing any system identification (Fig. 2(c)).

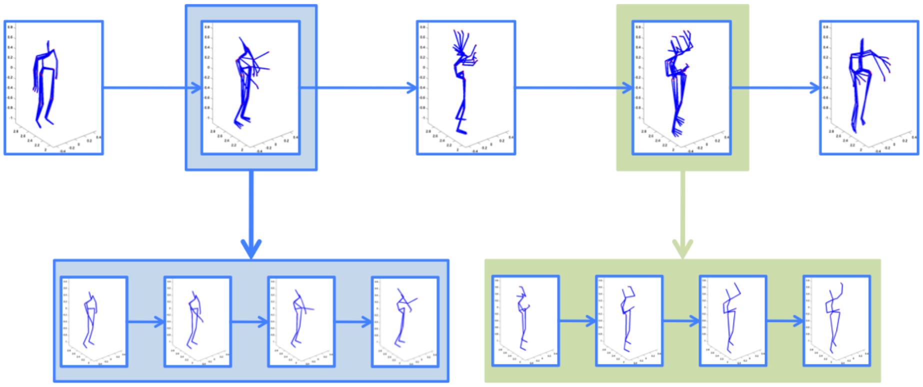
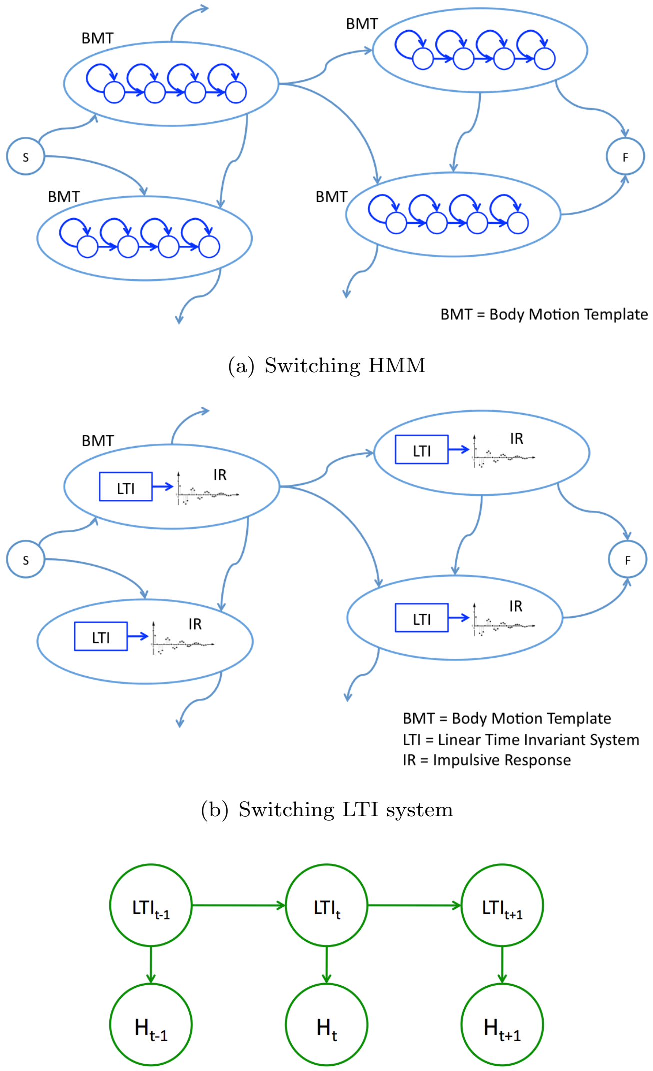


Figure 1: "Tennis serve" action from the MSRA-3D dataset. An action is a sequence of body motion templates (movements). Each body motion template is, in the case illustrated in this figure, a sequence of 3D Joints Trajectories characterized by their dynamics. The figure shows a sequence of only 5 body motion templates (sub-sampled from the original sequence) and expands only two of them for clarity.

The results presented here are an extension of our preliminary work [26]. In particular, in this paper:

- we account for the learning of the atomic LTI systems via a discriminative method that encourages correct predictions of the HMMs;
- we provide a deeper description of our discriminative learning approach in relation to former models;
- we present an extensive validation of our Hankelet-based action representation for different parameter settings.

The paper is organized as follows: in Section 2 we review previous work on action recognition and modeling, and on discriminative learning of HMM parameters; in Section 3 we present our Hankelet-based action representation; in Section 4 we explain our action model and describe our classification and LTI inference methods; in Section 5 we describe the discriminative learning of the model parameters and of the atomic LTI systems; in Section 6 we present experimental results on publicly available datasets and analyses of the performance



(a) Switching HMM

(b) Switching LTI system

(c) HMM of latent LTI systems

Figure 2: Fig. 2(a) shows an ideal model where the action is modeled as a switching HMM (similar to [21]). In Fig. 2(b), the action is modeled as a switching LTI system. This way we can represent a sub-trajectory by its Hankel matrix, and the action is modeled by a simple HMM (Fig. 2(c)) where the latent variables LTI represent atomic LTI systems, while the observed variables H are Hankel matrices.

of our technique for varying settings and parameters of the Hankelet-based representation. Finally, in Section 7 we present conclusions and outline future research directions.

2. Related Work

The literature about action recognition and time series modeling is very extensive. Here, we focus on three main aspects of the methods at the state-of-the-art: action representation, especially for 3D data, modeling of time-varying dynamics, and discriminative learning of parameters. We refer the reader to the following surveys for more general discussions on these topics: [27], [28], [29], [30], [31].

2.1. Action Representation

Most approaches for human action recognition in still images and RGB video [29] attempt to extract features that may be correlated with the human body pose (human body pose represents the configuration of body parts including head, arms, and legs). Descriptors such as Histogram of Oriented Gradients (HOG) [32], 3D-SIFT [33], Local Binary Pattern (LBP) [34] have been widely used in the literature. Often, a bag-of-words approach is used to compute a histogram of visual words based on a dictionary of local features [35].

Good motion representations can help to discriminate among actions during recognition. Several techniques have tried to combine body pose representation with motion information. Recently, Spatio-Temporal Interest Points (STIPs) [36] and Dense Trajectories (DT) [37, 25], jointly with Motion Boundary (MB) [38], have proved to increase accuracy of action recognition in video sequences.

Since the introduction of depth cameras and the work by Shotton, et al. [13] for estimating the body part locations in depth maps, several researchers have focused on the problem of recognizing actions from depth maps and/or 3D skeletons of the body.

A depth map stores the distance of each point in the scene to the camera. This allows reasoning about body surfaces and shapes across time. Li et al. [39]

proposed to use an action graph where each node is a bag of 3D points that encodes the body pose. In Wang et al. [19], a 3D action sequence is treated as a 4D shape and a Random Occupancy Patterns (ROP) feature is extracted. Sparse coding is used to encode only the features that contain information useful for classification purposes. In Vieira et al. [40], space and time axes are divided in cells, and space-time occupancy patterns are computed to represent depth sequences. Oreifej et al. [16] describe the depth sequence as histograms of oriented surface normals (HON4D) captured in the 4D volume, based on depth and spatial coordinates.

The main difficulty of working directly with 3D skeleton data arises from inaccuracy or failures of the skeleton estimation method. Moreover, "*one of the biggest challenges of using pose-based features is that semantically similar motions may not necessarily be numerically similar*" [41]. Most of the research using only 3D skeleton data tries to extract features to represent the correlation among the locations of the joints. In [15], the body pose is represented by concatenating the distances between all the possible pairs of the joints in the current frame, the distances between the joints in the current frame and the ones in the previous frame, the distances between the joints in the current frame and in a neutral pose (computed by averaging the initial skeletons of all the actions). Principal component analysis (PCA) is applied for dimensionality reduction providing a descriptor called EigenJoints.

In Xia et al. [14], a histogram of the locations of 11 manually selected 3D skeleton joints is computed to get a compact body pose representation that is invariant to the use of left and right limbs (Histogram of 3D Joints (HOJ3D)). Linear Discriminant Analysis (LDA) is used to project the histograms and compute the K discrete states of the HMM classifier. In [42], each action is represented by spatio-temporal motion trajectories of the joints. Trajectories are represented as curves in the Riemannian manifold of open curve shape space, and a dynamic programming-based elastic distance is used to compare them. Classification is performed by KNN on the Riemannian manifold.

Due to the difficulty of achieving high accuracy with just 3D skeleton data [20], other approaches combine skeleton data with other sources of information (depth maps or RGB video). In Wang et al. [20], depth data and the estimated 3D locations of the joints are used to compute the local occupancy pattern (LOP) feature. The Fourier temporal pyramid is used to capture the temporal structure of the action. Data mining techniques are used to discover the most discriminative actionlets and a multiple kernels learning approach is used to weight the actionlets. Sung et al. [43] combine HOG on RGB and depth data, hand positions, body pose and motion features from skeleton data. Then, a two-layer maximum-entropy Markov model is adopted for classification. In [44] the authors fuse skeleton information and STIPS within the random forest framework to perform feature selection and action classification.

In this paper, we only use the 3D locations of the joints in skeleton data. We adopt a Hankelet-based representation [24] to describe body motion in a sliding window, and a set of HMMs to perform action classification.

2.2. Modeling of Time Series

Our approach is related to both linear parameter varying model identification [45] and switched system identification [46]. In linear parameter varying models, the parameters of each autoregressive model may change over time based on a scheduling variable. Our method may be considered as a discretization of linear parameter varying models; we model the switching of the LTI systems as a Markov process and, instead of estimating the scheduling variable, we infer the atomic LTI system that may have generated the given observations. In this sense, our method is more similar to piecewise models and Markovian jump linear models [46], [47], [48] where there is a stochastic process that regulates the switching from one LTI system to another. Unlike previous methods [47], [48], our goal is not that of segmenting the sequence as outputs of different LTI systems; instead, we parse the sequence with a sliding window of fixed duration, and model probabilistically the switching among the atomic LTI systems to capture the temporal structure of the whole action.

To associate output measurements with a generating LTI system, we could apply system identification techniques to estimate the parameters of the LTI system, as in [49]. However, trajectories produced by a dynamical system can be also represented through a Hankel matrix. The Hankel matrix embeds the observability matrix of the LTI system, and it is invariant to affine-transformations of the trajectory points [24]. Hankel matrices have been successfully used in previous works on action recognition [24], tracking [50], compressive sensing [51], face emotion recognition [52], [53] and dynamic textures [49]. In [24], a bag of dynamical models is used for action recognition in RGB video. The method extracts dense trajectories to represent body motion. The truncated Hankel matrices (Hankelets) associated with the detected trajectories are used to learn a dictionary of Hankelets. A histogram of Hankelets is computed to represent each action instance and train a SVM. In such a method, the temporal structure of the action and the switching among LTI systems are not considered.

In our approach we use the space of Hankel matrices as an intermediate space where it is possible to compare the Hankelet-based representations of both body motion templates and atomic LTI systems. We model the transitions between LTI systems means of an HMM. In this sense, there is an interesting connection between our work and the work of [54]. In [54], each video sequence is associated with a dynamical model. Then, a metric is learned in order to optimally classify these dynamical models. In contrast, we represent a video as a sequence of dynamical models and learn the parameters of an HMM that may regulate this sequence of atomic models.

2.3. Discriminative Learning of Parameters

Among the various statistical learning approaches, we can distinguish between two categories: generative and discriminative learning methods [55]. The former schema aims at estimating the parameters of the probability distribution of the data based on the maximum likelihood or the maximum a posteriori principles. In contrast, discriminative approaches attempt to optimize a scoring function on the available training samples and the classifier's output. Such a scoring function is defined based on criteria that are directly linked with the classification purpose, such as conditional maximum likelihood or maximum mutual information [56], [57], empirical risk minimization [58] and large margin estimation [59].

One of the most used discriminative models for time series is the Conditional Random Fields (CRF) [60] model, which has proven to be successful for labeling tasks but, in general, requires a fully annotated training set. Therefore, Quattoni et al. [61] proposed Hidden Conditional Random Fields (HCRF) where the states of a temporal sequence are hidden but depend on the class. In the HCRF model, all classes share the same state space. The class label is inferred by considering the compatibility of the appearance features with the hidden states, the compatibility of the couple of states in the chain, and the compatibility of the inferred states with the class labels. Learning of the parameters for a chain-based HCRF requires inference of the hidden states, which can be solved via belief propagation.

The main difficulties in discriminative training of dynamical models arise when hidden variables are present. In general, hidden variables make the objective function non-convex and increase the complexity of the optimization problem [61]. By borrowing ideas from Latent variable Support Vector Machine (LSVM) [62] and structural SVM [63], Wang and Mori [64] developed Max-Margin HCRF (MMHCRF) for human action recognition in RGB video sequences. Using this approach, learning of the parameters requires a two step procedure: in the first step, given the currently estimated parameters, inference of the hidden states is performed by linear programming for each sample in the training set; in the second step, given the sequences of hidden states, multi-class SVM is learned in the dual space by solving a quadratic programming problem. The optimization is carried out only on the correct labeling and on the most violated constraints.

HMMs have been widely used for modeling time series. In particular, they have been widely used for action recognition [14], [65], [66], [67], [68], [69]. The standard generative learning scheme for HMMs is the well-known Baum-Welch algorithm [70], which is based on expectation-maximization to alternate between inference of hidden states and maximum likelihood estimation of the parameters. Due to the success of discriminative learning approaches, several researchers [71], [72], [73], [74] have attempted to learn the parameters of Markov networks in a discriminative way, especially in speech recognition.

In this paper, we train a set of HMMs (one for each action class) using a discriminative training approach. We define a loss function that measures the misclassification error as a "distance" between the correct labeling and the most competitive but wrong class label. During optimization, the method tries to enforce a "large margin" among the HMMs' decision boundaries.

3. Hankelet-based Representation

We represent an action as a sequence of body motion templates, where a template is defined as a set of feature trajectories in a temporal window of τ frames. In particular, we consider a temporal sequence $[y_o, \ldots, y_\tau]$, where y_t is a vector of the concatenated 3D locations of body joints in the skeleton at time t. As LTI systems are universal approximators [47], the temporal sequence can be regarded as the output of an LTI system of unknown parameters.

The state and measurement equations of LTI systems are linear, where the matrices A and C are constant over time, and $w_k \sim N(0, Q)$ is uncorrelated zero mean Gaussian measurement noise:

$$\begin{aligned} x_{k+1} &= A \cdot x_k + w_k; \\ y_k &= C \cdot x_k. \end{aligned} \tag{1}$$

In these equations, $x_k \in R^u$ is the u-dimensional hidden state of the LTI system, while $y_k \in R^v$ is the v-dimensional measurement.

We can describe the trajectory produced by a dynamical system through its Hankel matrix. Given a sequence of output measurements $[y_o, \ldots, y_\tau]$ from (1), its associated block-Hankel matrix is

$$\widetilde{H} = \begin{bmatrix} y_0, & y_1, & y_2, & \dots, & y_m \\ y_1, & y_2, & y_3, & \dots, & y_{m+1} \\ \dots & \dots & \dots & \dots & \dots \\ y_n, & y_{n+1}, & y_{n+2}, & \dots, & y_\tau \end{bmatrix}, \tag{2}$$

where n is the maximal order of the system, τ is the temporal length of the sequence, and it holds that $\tau = n + m - 1$.

The Hankel matrix embeds the observability matrix Γ of the system, since $\widetilde{H} = \Gamma \cdot X$, where X is the sequence of hidden states of the LTI system, and Γ is defined as follows:

$$\Gamma = \begin{bmatrix} C \\ CA \\ \dots \\ CA^n \end{bmatrix}.$$

Therefore, $\widetilde{H}$ provides information about the dynamics of the temporal sequence.

To compute the Hankelet, we first center the sequence by subtracting its average, then we build the Hankel matrix and normalize it as in [24]:

$$H = \frac{\widetilde{H}}{\sqrt{||\widetilde{H} \cdot \widetilde{H}^T||_F}}. \tag{3}$$

Based on the above definition, we represent an action sample as a sequence of Hankelets computed in a sliding window of duration τ.

To compare two Hankelets H_p and H_q, we adopt the same dissimilarity score introduced in [24] and defined as follows:

$$d(H_p, H_q) = \quad 2 - ||H_p \cdot H_p^T + H_q \cdot H_q^T||_F. \tag{4}$$

We highlight that Eq. 4 is a surrogate used to estimate the subspace angle between the spaces spanned by the columns of the Hankel matrices and it does not define a distance. Nonetheless, this score conveys the degree to which two

Hankelets may correspond to the same dynamical system [24], and therefore we assume this score represents the degree to which two trajectories have been produced by the same LTI system.

4. Hankelet-based HMM

As shown in Fig. 2(c), our switching dynamical system model reduces to an HMM where the hidden variable LTI represents the atomic linear time invariant system that has generated the current trajectory. Each observed trajectory is represented by the corresponding Hankelet H. For each class, we learn a set of M atomic LTI systems (as will be described in Sec. 5.1). Each atomic LTI system is modeled by means of the Hankelet S of an exemplar output sequence that the system has produced. Therefore, for each class c, the set of atomic LTI systems is represented by a set of Hankelets $\{S^{c,i}\}_{i=1}^{M}$.

The probability that a given sequence of measurements is produced by an LTI system is modeled by the following exponential distribution:

$$p(H|S^{c,i}, \lambda^{c,i}) = \lambda^{c,i} \cdot e^{-\lambda^{c,i} \cdot d(H,S^{c,i})} \tag{5}$$

where H is the Hankelet corresponding to the given sequence of measurements, $S^{c,i}$ is the Hankelet used for representing the i-th atomic LTI system of the c-th class model, $d(H, S^{c,i})$ is the dissimilarity score in Eq. (4), $\lambda^{c,i}$ is the parameter of the exponential distribution that corresponds to the i-th state of the c-th model and has to be learnt from training data.

The switching process that generates an action is assumed to be a Markovian process and is modeled by an HMM. The HMM of the c-th action class is characterized by a stochastic transition matrix T^c such that $T^c(i,j) = P(S_t = S^{c,j}|S_{t-1} = S^{c,i})$. We also define a prior probability π^c such that $\pi^c(i) = P(S_0 = S^{c,i})$ is the probability that the measurement in the first temporal window ($t = 0$) has been generated by the i-th atomic LTI model.

The joint probability of a sequence of N observed Hankelets $\mathbf{H} = \{H_t\}_{t=0}^{t=N}$ and the sequence of generating LTI systems represented by their corresponding

```
Input  : ${H_t}_{t=0}^{t=W}$ test sequence;
         ${\Lambda^c}_{c=1}^{c=N}$, ${T^c}_{c=1}^{c=N}$, ${\pi^c}_{c=1}^{c=N}$ parameters of the HMMs;
         ${S^{c,i}}_{c=1,i=1}^{;c=N,i=M}$ state space
Output: $c^*$ predicted label

%% For each class $c$
for $c \leftarrow 1$ to $N$ do
    %% For each state $i$
    for $i \leftarrow 1$ to $M$ do
        %% For each Hankelet $H_t$
        for $t \leftarrow 1$ to $W$ do
            $D^c(i,t) \leftarrow d(S^{c,i}, H_t)$ (eq. 4)
        end
    end
    %% Compute Negative Log-Likelihood of c-th model
    NLL($c$) ←applyViterbi($D^c, \Lambda^c, T^c, \pi^c$)
end

%% Classify test sequence
$c^* \leftarrow$ argmin($NLL$) (eq. 7)
```

Algorithm 1: Inference of Action-Class

Hankelets $\mathbf{S} = \{S_t\}_{t=0}^{t=N}$ is:

$$p(\mathbf{H}, \mathbf{S}|T^c, \pi^c, \Lambda^c) = P(S_0) \cdot p(H_0|S_0, \lambda^{c,S_0}) \cdot \prod_{t=1}^{t=N} (p(H_t|S_t, \lambda^{c,S_t}) \cdot P(S_t|S_{t-1})) =$$

$$= \pi^c(S_0) \cdot p(H_0|S_0, \lambda^{c,S_0}) \cdot \prod_{t=1}^{t=N} (p(H_t|S_t, \lambda^{c,S_t}) \cdot T^c(S_{t-1}, S_t)) \quad (6)$$

where, with a little abuse of the notation, we are using the states as indices, and $\Lambda^c = \{\lambda^{c,s}\}$ is the set of parameters $\lambda^{c,s}$ associated with each state s of class c.

```
Input  : $\{\lambda_s\}_{s=1}^{M}$, $T$, $\pi$ param. of the $c$-th HMM;
         $D$ score matrix of $W$ observed Hankelets to $M$ states
Output: $NLL$ negative log-likelihood;
         $\{S_t\}_{t=0}^{t=W}$ inferred sequence of states

%% For each state $i$
for $i \leftarrow 1$ to $M$ do
    %% For each observed Hankelet $H_t$, compute log-likelihood
    for $t \leftarrow 1$ to $W$ do
        $\log P_{H|S}(i,t) \leftarrow -\lambda_i \cdot D(i,t) + \log \lambda_i$ (log. of eq. 5)
    end

    %% Initialize joint log-likelihood
    $\log P_{H,S}(i,1) \leftarrow \log P_{H|S}(i,1) + \log \pi(i)$
end

%% Do dynamic programming: for each time step $t$
for $t \leftarrow 2$ to $W$ do
    %% For each state $i$
    for $i \leftarrow 1$ to $M$ do
        %%Compute joint log-likelihood and the best previous step
        $\log P_{H,S}(i,t) \leftarrow \max_h \{\log P_{H,S}(h,t-1) + \log T(h,t)\} + \log P_{H|S}(i,t)$
        bestState$(i,t) \leftarrow \operatorname{argmax}_h \{\log P_{H,S}(h,t-1) + \log T(h,t)\}$
    end
end

%% Infer the best state-path
$NLL \leftarrow -\max_h \{\log P_{H,S}(h,W)\}$
Compute inferred sequence of states $\{S_t\}_{t=0}^{t=W}$ by back-tracking
```

Algorithm 2: applyViterbi (Decoding of the observed Hankelet sequence)

4.1. Inference and Classification

Given an action model described by parameters $\{\Lambda^c, T^c, \pi^c\}$, where c is the label of the action to which the model refers, the inference of the sequence of LTI-systems is performed via the Viterbi algorithm [75]. This well-known algorithm is based on Dynamic Programming and attempts to maximize the log-likelihood of the joint probability of the states and the observations sequentially.

The inference of which label should be assigned to a sequence of Hankelets $\mathbf{H} = \{H_t\}$ is performed via maximum likelihood. In practice, the predicted label c^* is computed solving:

$$c^* = \min_c \{-\max_{\mathbf{S}} \log p(\mathbf{H}, \mathbf{S}|T^c, \pi^c, \Lambda^c)\}. \tag{7}$$

Then, the label of the model providing the highest likelihood is assigned to the sequence of observations. $\mathbf{S} = \{S_t\}$ is the best sequence of hidden states inferred by the Viterbi algorithm.

Algorithm 1 shows how the classification of an input Hankelet sequence is performed. The Viterbi algorithm is applied N times, once for each action class.

For completeness, we report in Algorithm 2 the Viterbi algorithm for inferring the sequence of LTI systems that generated the observed sequence and the negative log-likelihood.

5. Discriminative Training of HMMs

Our discriminative learning procedure learns the parameters of all the HMMs simultaneously by encouraging correct predictions and penalizing the incorrect ones based on the values of the negative log-likelihood provided by the models.

Based on Eq. 6, the negative log-likelihood provided by the HMM of class c for a sequence of Hankelets may be written as:

$$\begin{gathered} g^c(\mathbf{H}) = -\log(p(\mathbf{H}, \mathbf{S}|T^c, \pi^c, \Lambda^c)) = \\ -\textstyle\sum_{t=0}^{t=N} \log(p(H_t|S_t, , \lambda^{c,S_t})) - \sum_{t=1}^{t=N} \log(T^c(S_{t-1}, S_t)) - \log(\pi^c(S_0)) \end{gathered} \tag{8}$$

where $\mathbf{S} = \{S_t\}$ is computed by the Viterbi algorithm as

$$\mathbf{S} = arg \max_{\mathbf{S}} p(\mathbf{H}, \mathbf{S} | T^c, \pi^c, \Lambda^c). \tag{9}$$

The decision about which label to assign to $\mathbf{H}$ is taken by maximum likelihood as detailed in Eq. 7. Therefore, if the sample belongs to the k-th class, to obtain a correct prediction we need to have:

$$g^k(\mathbf{H}) < \min_{j \neq k} g^j(\mathbf{H}) \tag{10}$$

where superscripts indicate the class to which the negative log-likelihood refers. The difference $\delta g(\mathbf{H}, k, j) = g^k(\mathbf{H}) - \min_{j \neq k} g^j(\mathbf{H})$ represents the distance between the correct model and the most competitive but incorrect one, and can be interpreted as the margin of the classifier. Minimizing δg over the whole training set corresponds to increasing the inter-class distances of the classifier. In case of a correct prediction, then $\delta g(\mathbf{H}, k, j) < 0$. We can enforce a large negative log-likelihood difference by requiring that $\delta g(\mathbf{H}, k, j) < -m$ where m is a positive constant value (we set m to 1).

For this purpose, in our optimization problem, we adopt a hinge loss function defined as:

$$loss(\mathbf{H}) = \max\{0, g^k(\mathbf{H}) - \min_{j \neq k} g^j(\mathbf{H}) + m\}. \tag{11}$$

Whenever this loss is greater than 0, the prediction is incorrect or the achieved margin is not large enough. In both cases, an update of the model parameters is required.

In our formulation, the optimization involves hidden variables, which makes the problem non-convex as in other related frameworks such as [61], [64]. In the HCRF [61], classification is based on the conditional likelihood $P(y|\mathbf{H})$ where y is the class-label. This conditional likelihood is computed by summing $P(\mathbf{S}, y|\mathbf{H})$ over all the possible state paths. In our discriminative HMMs, the classification is based exclusively on the optimal state path $\mathbf{S}$ and on the maximum joint probability $P(\mathbf{S}, \mathbf{H}|y)$. The main difficulty arises in jointly estimating the optimal state path and learning the parameters. In this sense, we take an approach that is analogous to the approach taken in [62] and [64] and, at each iteration, we adopt a coordinate descent approach. In practice, we optimize our objective function in two steps:

1. Holding the parameters of the model fixed, we infer the optimal path $\mathbf{S}$ for each sample and for each class, and we store the paths of the correct model and the most competitive but incorrect one;
2. Holding the sequences of hidden variables (paths), we correct the model parameters by minimizing the loss function.

In our formulation we only select the most competitive but incorrect model by taking the minimum of the negative log-likelihood (NLL) over all the incorrect models. Our optimization is based on the joint probability $P(\mathbf{S}, \mathbf{H}|y)$. As an alternative, we might have marginalized over all the possible paths S (which means performing inference via the forward algorithm rather than decoding via the Viterbi algorithm) but this choice would have yielded complex expressions for the gradients of the loss function. In contrast, our choices of classifying a sequence based on the NLL of the optimal paths of each HMM, and of considering the joint probability $P(\mathbf{S}, \mathbf{H}|y)$ result in a straightforward and convenient computation of the gradients of the cumulative loss.

Whilst our model is not a structural SVM, our adopted strategy has an analog in structural learning [63] where the optimization is carried out on the set of most violated constraints. In our method, only the sequences for which the models do not achieve a large margin contribute to the parameter refinement.

5.1. Learning of atomic LTI systems

The state space, that is the set of atomic LTI systems, is initialized by computing K clusters per class via K-medoids.

In our preliminary work [26], the atomic LTI systems were not updated during the training procedure. When inferring the state-paths given the correct class-label for all the sequences, we obtain a partition of the Hankelets into clusters (state-clusters), each one associated with a state. We empirically found that updating the state space with the medoids of these clusters does not improve the final classification accuracy [26]. Indeed, this strategy does not ensure that the cumulative loss function decreases, and there is no guarantee that the optimization procedure will converge towards a local minima.

In this paper, we retrain the state space and select the atomic LTI systems in such a way to encourage correct predictions of the classifier. We consider the Hankelets of the correctly classified sequences within a state-cluster as candidate representations of the atomic LTI systems. Ideally, we would like to find a joint set of M states for each class that decreases the loss on the training set. However, this problem has an exponential complexity. Therefore, we apply a greedy strategy where we re-estimate one state at a time. For each state $S^{c,i}$ and for each candidate Hankelet $C^{c,j}$, we compute the cumulative loss on the whole training set on the state space $C^{c,j} \cup \{S^{c,k}\}_{k \neq i}$. If this loss is lower than the loss computed on the initial state space, we accept the change; otherwise, we reject the candidate Hankelet. To further reduce the computational complexity of the method, we only consider P random candidates at each step, chosen within the state-cluster.

5.2. Implementation Details

In contrast to the HCRF model, in the HMM the parameters of the model are subjected to constraints. In particular, for each class c, the prior parameters π^c must be positive and sum to 1; the transition probability parameters T^c must be positive and must form a stochastic matrix, which means that they must sum to 1 per row (each row of T^c represents $P(S_t|S_{t-1})$ for a given S_{t-1}). Moreover, considering the emission probability defined by Eq. 5, namely an exponential probability density function, the parameters Λ^c must be positive.

We neglect the constraints on Λ^c (see [76], page 6), and enforce that these parameters assume a value greater than 0.

As for the priors, we make the following assumption:

$$\pi^c(S) = \frac{\widetilde{\pi}^c(S)}{\sum_S \widetilde{\pi}^c(S)}. \tag{12}$$

When computing the log-likelihood we consider:

$$\log(\pi^c(S)) = \log(\widetilde{\pi}^c(S)) - \log(\sum_S \widetilde{\pi}^c(S)). \tag{13}$$

While $\pi^c(S)$ must be positive and sum to 1, we do not have any constraints on $\log(\widetilde{\pi}^c(S))$. Therefore, we redefine the variables in our optimization problem as follows:

$$\beta_S^c = \log(\widetilde{\pi}^c(S)), \tag{14}$$

so that

$$\log(\pi^c(S)) = \beta_S^c - \log(\sum_S e^{\beta_S^c}). \tag{15}$$

Similar considerations hold also for the parameters $T^c(S, S')$. Therefore, we define the following variable

$$\alpha_{S,S'}^c = \log(\widetilde{T}^c(S, S')), \tag{16}$$

such that

$$\log(T^c(S, S')) = \alpha_{S,S'}^c - \log(\sum_S e^{\alpha_{S,S'}^c}). \tag{17}$$

With these variable re-definitions, the original constrained optimization problem becomes an unconstrained one.

Our cumulative loss over all the training samples is a non-convex function. We use gradient descent to minimize the objective function by adopting a quasi-Newton strategy with limited-memory BFGS updates. We adopt a block-coordinate descent approach such that the optimization is carried out on the parameters $\{\Lambda^c\}_c$, $\{T^c\}_c$ and the prior $\{\pi^c\}_c$ in turn.

Algorithm 3 shows the pseudo-code for our training procedure. After initializing all the models with the same parameters (uniform distributions for T^c and π^c and 1 for λ^c), the method iteratively estimates the best set of atomic LTI systems by means of the function est_states() (see Sec. 5.1), and minimizes the objective function $f(\cdot)$ on each block of variables. The function check_convergence() checks if some convergence criterion is met (no significant changes in the estimated parameters).

Algorithm 4 summarizes the main steps to evaluate the cumulative loss function over the training set. For each sample, it computes the negative log-likelihood of the correct model and of the most likely but incorrect model. This

```
Input  : ${\mathbf{Y_i}}_{i=1}^{W}$: training set of Hankelet sequences;
         labels: action-class for each training sequence;
Output: $\{\Lambda^c\}_{c=1}^N$, $\{T^c\}_{c=1}^N$, $\{\pi^c\}_{c=1}^N$ parameters of the HMMs;
         $\{S^{c,i}\}_{c=1,i=1}^{N,M}$ state space

%% Parameter initialization for each class-model
for $c \leftarrow 1$ to $N$ do
    $\lambda^c \leftarrow$ all-ones vector of dimension $M$
    $T^c \leftarrow M$ x $M$ stochastic matrix with uniform distribution on each row
    $\pi^c \leftarrow$ uniform distribution over the $M$ states
end

iter $\leftarrow 1$
converged $\leftarrow false$

while $iter < Max\_Iter$ & $!converged$ do
    %% Re-train States as described in Sec. 5.1
    $\{S^{c,i}\}_{c=1,i=1}^{N,M}$ =est_states($\{\Lambda^c\}_{c=1}^N, \{T^c\}_{c=1}^N, \{\pi^c\}_{c=1}^N, \{\mathbf{Y_i}\}_{i=1}^W$,labels)

    %% Optimize parameters
    $[\{\Lambda^c\}_{c=1}^N, \{T^c\}_{c=1}^N, \{\pi^c\}_{c=1}^N] \leftarrow$
    argmin $f(\{\mathbf{Y_i}\}_{i=1}^W$, labels, $\{\Lambda^c\}_{c=1}^N, \{T^c\}_{c=1}^N, \{\pi^c\}_{c=1}^N, \{S^{c,i}\}_{c=1,i=1}^{N,M})$

    converged $\leftarrow$ check_convergence($\{\Lambda^c\}_{c=1}^N, \{T^c\}_{c=1}^N, \{\pi^c\}_{c=1}^N$)
    iter $\leftarrow$ iter + 1
end
```

Algorithm 3: Discriminative learning of the Parameters

is achieved by applying the Viterbi algorithm. If the loss is positive, then the models have produced a wrong prediction or the achieved margin is not large enough; therefore, the gradients are accumulated and returned to the L-BFGS algorithm to update the parameters.

```
Input  : $\{\mathbf{Y_i}\}_{i=1}^{W}$: training set of Hankelet sequences;
         labels: action-classes for each training sequence;
         $\{S^{c,i}\}_{c=1,i=1}^{N,M}$ state space;
         $\{\Lambda^c\}_{c=1}^{N}$, $\{T^c\}_{c=1}^{N}$, $\{\pi^c\}_{c=1}^{N}$ parameters of the HMMs;
Output: Cum_loss: loss over all the samples in the dataset;
         Grad: gradients

%% Accumulate loss for all the sequences in the training set (Eq. 11)
$m \leftarrow 1$
Cum_loss $\leftarrow 0$
for $i \leftarrow 1$ to $W$ do
    %% Compute loss for the i-th training sequence
    for $c \leftarrow 1$ to $N$ do
        $\mathrm{D}^c \leftarrow$ dissimilarity score matrix between $\mathbf{Y_i}$ and $\{S^{c,i}\}_{i=1}^{M}$
    end
    $k \leftarrow$ labels(i)
    $[g^k, z^k] \leftarrow$ applyViterbi($\mathrm{D}^k, \Lambda^k, T^k, \pi^k$) (Eq. 8)
    $[g^c, z^c] \leftarrow \min_{c \neq k}$ applyViterbi($\mathrm{D}^c, \Lambda^c, T^c, \pi^c$) (Eq. 8)
    loss $\leftarrow \max(0, g^k - g^c + m)$ (Eq. 11)
    Cum_loss $\leftarrow$ Cum_loss + loss

    %% If the sequence is misclassified, accumulate gradients. The
    %% optimization algorithm will use the gradients to update
    %% the parameters
    if loss> 0 then
        Accumulate gradients Grad for the parameters along the inferred
        paths $z^k$ and $z^c$ for classes $k$ and $c$ respectively
    end
end
```

Algorithm 4: f() : Objective Function to Minimize

6. Experimental Results

We evaluated our method on two publicly available datasets: MSRA-3D [20] and UCF [77]. In all our experiments, we initialize the set of states at random and we report both the overall classification accuracy and the average per-class classification accuracy values over 10 runs. The overall accuracy is computed as percentage of correctly classified test samples. The per-class accuracy values are computed as the percentage of correctly classified test samples within each class.

In our experiments we only used the body skeletons without performing any pre-processing of the data; such data are corrupted by various levels of noise/failures of the skeleton estimation method. This affects, in general, the recognition accuracy.

Each dataset has its own recommended evaluation protocol and data splitting, which we briefly describe below.

MSRA-3D: The MSRA-3D dataset [2] provides both skeleton and depth data (taken at about 15 fps). The dataset provides the skeleton (20 joints) for 20 actions, performed between 2 to 3 times by each of 10 subjects. During data collection, the subjects were facing the camera. The actions cover various movements of arms, legs, torso and their combinations. If an action is performed by a single arm or leg, the subjects were advised to use their right arm or leg.

We use the 3D coordinates from 557 sequences. The range of the lengths of the sequences is $[13, 76]$, with an average duration of 39.6 ± 10. We use the same experimental setting reported on the authors' website; hence, the splitting of the data in training and test set is as follows: subjects 1, 3, 5, 7, and 9 for training, the others for test.

The MSRA-3D dataset includes some sequences where the skeleton data is noisy, corrupted, and/or missing. Fig. 3 shows some frames from such sequences, together with the corresponding class labels. In the last image of *Pick*

[2] `http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/`

up and Throw, no skeleton has been detected likely due to a failure of the skeleton estimation method.

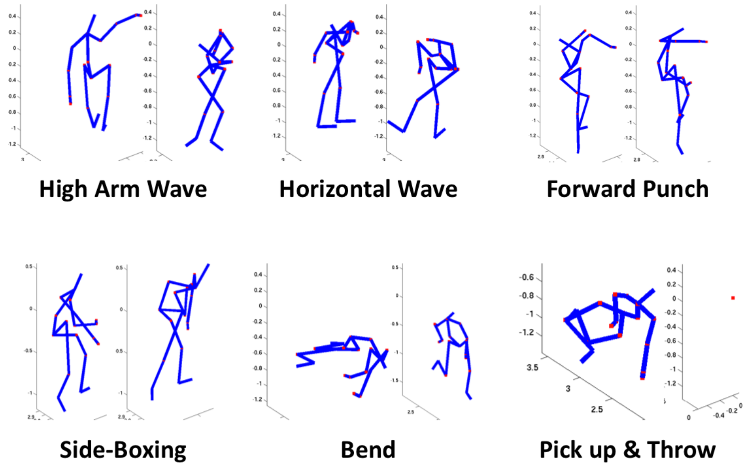

High Arm Wave
Horizontal Wave
Forward Punch
Side-Boxing
Bend
Pick up & Throw

Figure 3: Noisy and/or corrupted skeletons from the MSRA-3D dataset.

This dataset has been extensively adopted in the literature. We have found two main evaluation protocols used with this dataset**: MSRA-3D Protocol 1 and MSRA-3D Protocol 2**. We report experimental results for both of these evaluation protocols.

- **MSRA-3D Protocol 1.** The first protocol consists of testing over the whole set of 20 classes: *high arm wave (HAW), horizontal arm wave (HoW), hammer (H), hand catch (HCa), forward punch (FP), high throw (HT), draw x (DX), draw tick (DT), draw circle (DC), hand clap (HCl), two hand wave (2HW), side-boxing (SB), bend (B), forward kick (FK), side kick (SK), jogging (J), tennis swing (TSw), tennis serve (TSe), golf swing (GS), pickup and throw (PT).*

- **MSRA-3D Protocol 2.** The second evaluation protocol splits the actions into 3 overlapping subsets of 8 classes each. The first action set (AS1) includes the actions *horizontal arm wave, hammer, forward punch, high throw, hand clap, bend, tennis serve, pickup and throw.* The second action set (AS2) includes *high arm wave, hand catch, draw x, draw tick, draw circle, two hand wave, forward kick, side boxing.* The third action set (AS3) includes *high throw, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup and throw.* The AS1 and AS2 sets group actions that require similar movements, while the AS3 set groups more complex actions.

UCF: The UCF dataset [3] provides only the skeleton data. The framerate for data acquisition is unknown. The dataset provides the skeleton (15 joints) data for 16 actions performed 5 times by 16 individuals. There are 1280 action samples in total with a temporal duration that ranges in $[27, 229]$, with an average length of 66 ± 34 frames. Each action starts and ends from/to a resting pose. We used the 3D coordinates of the locations of the body joints. The actions in this dataset are: *balance, climbladder, climbup, duck, hop, kick, leap, punch, run, stepback, stepfront, stepleft, stepright, twistleft, twistright, vault.* We adopted the suggested evaluation protocol, which is a 4-fold cross evaluation[77].

6.1. Baselines and Parameter Initialization

The parameters of our discriminative HMM (DHMM) classifier were initialized as follows: the parameters λ were set to 1, the transition probabilities and the prior probabilities were initialized to uniform distributions. Therefore, all the HMMs are the same at the beginning of the training. In contrast to our earlier work [26], in this paper we train HMMs with a different state space. This largely reduces the number of parameters and speeds-up the training procedure. The atomic LTI systems (state space) were computed by K-medoids, and the initial medoids were selected at random from the training data.

We compare our method with two baselines: a standard HMM classifier (SHMM) where a set of HMMs (one for each action class) has been trained using the Baum-Welch algorithm, and an HCRF. For SHMM and DHMM, the state space was initialized in the same way.

[3] `http://www.cs.ucf.edu/~smasood/research.html`

The standard HCRF formulation was modified in such a way that each hidden state refers to an atomic LTI system. In practice, the input for the HCRF is the sequence of dissimilarity scores between the observed Hankelet and the states. The total number of states for HCRF is 8 times the number of classes.

While we can employ the same parameter initialization of our method for the SHMM, we have been forced to initialize the HCRF parameters randomly. Indeed, as the classes share the state parameters, a uniform initialization does not allow the model to differentiate one class from another.

We report accuracy in classification with our Hankelet-based action representation for all the three models. We report the accuracy of our method with (DHMM-SL) and without (DHMM) our atomic LTI systems learning procedure.

Furthermore, to gain more insights about our discriminative learning approach, we present accuracy in classification for SHMM and DHMM when the input of the methods is directly the skeleton (that is, the concatenation of the 3D locations of the body joints on a frame-per-frame basis). In this case, we adopt the Euclidean distance to compare skeletons, and initialization of the states is performed by K-means.

The number of states for each SHMM and DHMM has been set empirically to 8. In Sec. 6.2 and 6.4, we performed the experiments with square block-Hankelets, by empirically setting the order n of the Hankel matrix to 4. Hence, the sliding window is of 7 frames. In Sec. 6.5, we test the SHMM with different values of order and window length.

6.2. Results on MSRA-3D – Protocol 1

We conducted a cross-subject evaluation on the MSRA-3D Action dataset, and present the average accuracy values in classification in Table 1.

The first part of the table shows accuracy in classification for methods at the state-of-the-art. The second part of the table shows the accuracy achieved by SHMM and our DHMM using the 3D locations of the joints. The results show that our discriminative learning approach improves the accuracy by ap-

Methods (on 3D data)	Accuracy
Most Informative Joints + SVM [78]*	33.33
RNN [79]**	42.5
Log. Reg. [77]	65.7
Lie-Group + DTW + SVM [80]	89.48
BoW + SVM (freq. pattern) [81]	90.22
LTBSVM [18]****	91.21
SHMM(Joints)	55.60
DHMM(Joints)	63
HCRF (Hankelets)	55.7
SHMM (Hankelets)	85.60
DHMM (Hankelets)	88.64
DHMM-SL (Hankelets)	89.23

Table 1: Average Accuracy (in %) on the MSRA-3D action dataset – Protocol 1. Our results are averaged on 10 runs. * Results on 17 classes. ** Results reported in [77]. *** It excludes 13 more corrupted sequences. ****Different splitting of the subjects. Our method achieves accuracy similar to that of [80] and [81], which use a comparable evaluation setting. Our DHMM attains superior performance with respect to SHMM on equal terms of feature representation.

proximately 13% with respect to the standard learning approach. The lower part of the table shows the accuracy attained by HCRF, SHMM, DHMM and DHMM-SL using our Hankelet-based action representation. As the table shows, the state learning approach slightly increases the overall accuracy of the DHMM (approximately 0.5%). In comparison to SHMM, the accuracy of our DHMM improves upon it by 3.5% and, when adopting our state learning approach, by 4.25%.

Overall, by employing our Hankelet-based representation, our method can attain state-of-the-art accuracy. In particular, our method attains an accuracy similar to that obtained by [80] and [81] on equal terms of data splitting. Our method does not need any data pre-processing to compute the Hankelet; in

Methods (multi-stream)	Accuracy
Action Graph [39]	74.7
DMM-HOG [82]	85.52
HON4D [16]	85.8
Rnd Occupancy Patterns (ROP) [19]	86.5
Actionlet Ensemble (LOP) [20]	87.21
HON4D + D_{disc} [16]	88.89
JAS [83]	94.84
Random Forest [44]	94.3

Table 2: Accuracy on the MSRA-3D action dataset of methods that use other data than just the skeletons.

contrast, [80] needs to account for biometric differences in the estimated skeletons by performing data normalization and skeleton registration, and applies Dynamic Time Warping (DTW) to a reference sequence to account for varying lengths of the sequences. While we learn a model for each action class, [81] learns one-vs-one SVM (i.e. 190 SVMs on the MSRA-3D dataset) and adopts a voting scheme to classify the actions.

For completeness, we also report in Table 2 the results for methods at the state-of-the-art that use hybrid descriptors extracted from skeleton data and RGB video and/or depth maps. As these methods also use RGB/Depth data, they are not directly comparable with the ones in Table 1, which use instead only the 3D joint positions. We may note that still our method achieves competitive accuracy values.

Fig. 5 shows the confusion matrix for all the classes. This matrix was obtained by averaging the classification results over 10 runs. Therefore, the classes for which we report 100% accuracy were consistently and correctly classified in all the runs.

The figure shows that most of the confusion is between some pairs of action classes. In particular, there is confusion between *high arm wave* (HAW) and *horizontal arm wave* (HoW), *hand catch* (HCa) and *high throw* (HT), *forward* *punch* (FP) and *high throw* (HT), *forward punch* (FP) and *tennis serve* (TSe), *high throw* (HT) and *tennis serve* (TSe), *side kick* (SK) and *forward kick* (FK). Comparing these pairs of classes, it is possible to note that these actions involve the same joints and may have similar dynamics and, therefore, similar Hankelets. We stress that our Hankelet-based representation can capture the dynamics of the body joints, but not their relative positions, which can instead help in these cases. To solve such ambiguities, it might be possible to combine our Hankelet-based representation with other pose representations. From a modeling point of view, this might be implemented by a Cartesian product of discrete state variables (to jointly represent dynamics and poses) at the cost of a higher computational complexity.

Some confusion is present between *hand catch* (HCa) and *draw x* (DX), and between *high arm wave* (HAW) and *draw x* (DX). This is probably due to the variability among subjects with which DX has been performed.

As for the pair of actions *bend* (B) and *pickup and throw* (PT), we have visually inspected the data and noted that: 1) the action B is a sub-action for PT; 2) due to skeleton tracking failures, in several PT sequences, the skeleton is not reliable or is totally missing. As a result, PT sequences are severely corrupted and several of them reduce to a *bend* action. This is also shown in Fig. 4: the first and the second lines of skeletons refer to the actions PT and B respectively. As the figure shows, the action B may look like a sub-action of PT since the two action classes are characterized by similar movements of the body. The third and fourth lines show two sequences of the same classes that are strongly corrupted by noise and failures of the skeleton tracker. Out of the 27 samples in the PT class, 6 present missing skeletons, with about 50% of missing on average. Of the 21 samples without missing, about 8 sequences have more than 3 severely corrupted skeletons (see Fig. 3 for examples of corrupted skeletons).

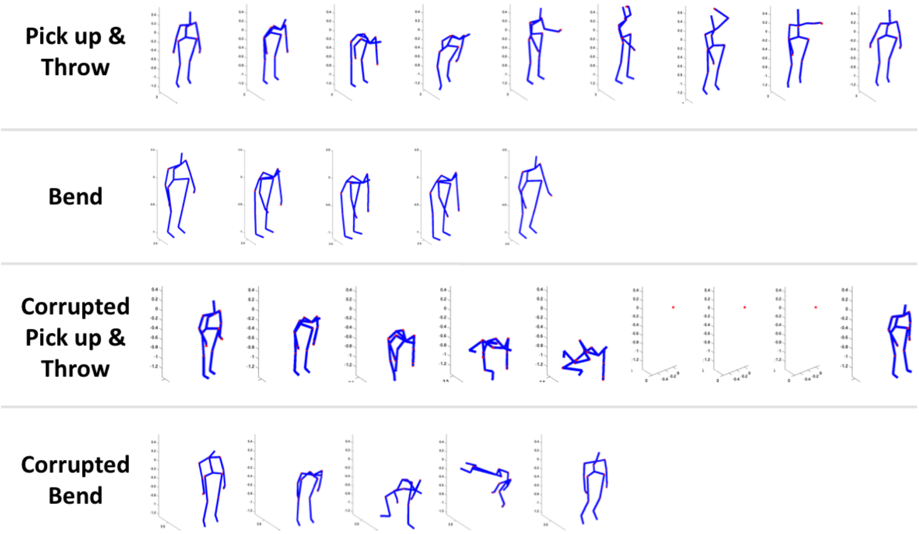


Figure 4: Corrupted skeletons from the *bend* and *pick up & throw* action classes in the MSRA-3D dataset.

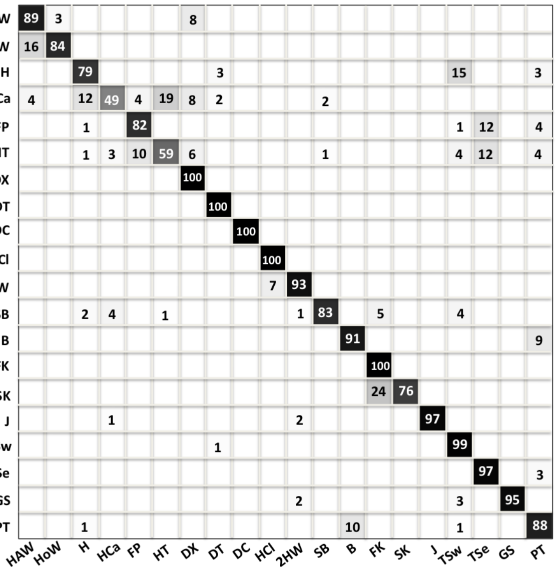
6.3. Results on MSRA-3D – Protocol 2

We also conducted a cross-subject analysis on the three subsets AS1, AS2 and AS3 whose results are reported in Table 3.

Looking at the accuracy values of SHMM (Joints) and DHMM (Joints), which are trained directly on the 3D locations of the joints, our discriminative learning approach yields an improvement of the accuracy of about 5.2%, 4.64% and 19,78%, respectively on the sets AS1, AS2 and AS3. On average, the accuracy on the 3 sets increases by 9.89%.

When using the Hankelet-based representation, our discriminative learning approach yields an increase in accuracy of about 2.3%, on average, for the 3 sets. When the state learning approach is also adopted, the average accuracy increases by 3.89%.

Overall, our method attains accuracy similar to that in [42] and [80]. We note that [42] adopts a KNN classifier with a dynamic programming based distance, which means that classification is performed by comparing against all the sequences in the training set. Our classification is based on the Viterbi



	HAW	HoW	H	HCa	FP	HT	DX	DT	DC	HCl	2HW	SB	B	FK	SK	J	TSw	TSe	GS	PT
HAW	89	3					8													
HoW	16	84																		
H			79					3									15			3
HCa	4		12	49	4	19	8	2				2								
FP			1		82												1	12		4
HT			1	3	10	59	6					1					4	12		4
DX							100													
DT								100												
DC									100											
HCl										100										
2HW										7	93									
SB			2	4		1					1	83		5			4			
B													91							9
FK														100						
SK														24	76					
J				1							2					97				
TSw								1									99			
TSe																		97		3
GS											2						3		95	
PT			1										10				1			88

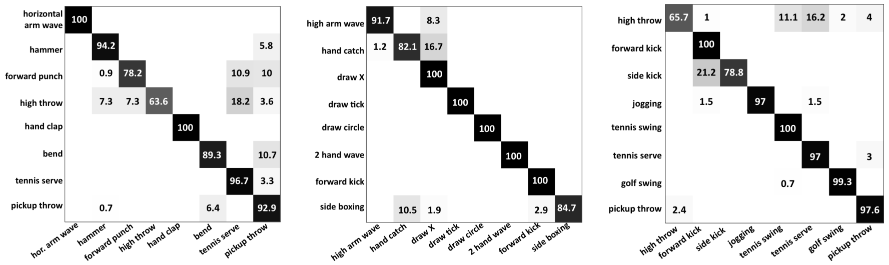
Figure 5: Confusion matrix of the 20 classes of the MSRA-3D dataset in cross-subjects evaluation (results averaged on 10 runs). The actions in the dataset are: *high arm wave* (HAW), *horizontal arm wave* (HoW), *hammer* (H), *hand catch* (HCa), *forward punch* (FP), *high throw* (HT), *draw x* (DX), *draw tick* (DT), *draw circle* (DC), *hand clap* (HCl), *two hand wave* (2HW), *side-boxing* (SB), *bend* (B), *forward kick* (FK), *side kick* (SK), *jogging* (J), *tennis swing* (TSw), *tennis serve* (TSe), *golf swing* (GS), *pickup and throw* (PT).

algorithm, which is applied once for each class. Therefore, our classification procedure has a lower computational complexity with respect to [42].

Fig. 6 shows the confusion matrices for the three subsets, AS1, AS2, and AS3. These matrices are consistent with the one obtained when testing on all the classes. In AS1, most of the confusion is again between the classes *forward punch* and *tennis serve*, *forward punch* and *high throw*, *high throw* and *tennis serve*, *bend and pickup* and *throw*. This time, *horizontal arm wave* reaches 100% accuracy (on all the 10 runs). Indeed, the AS1 subset does not contain the class *high arm wave*, which was confusing the classifier in the experiment with all 20 classes.

Methods (on 3D data)	AS1	AS2	AS3	Average
Joint-Histogram + HMM [14]	87.98	85.48	63.46	78.97
EigenJoints-NN [15]	74.50	76.10	96.40	82.33
Skeletal Quads + SVM [17]	88.39	86.61	94.59	89.86
KNN-Riemannian [42]	90.1	90.6	97.6	92.77
Multi-level HDP-HMM [84]**	81.2	78.1	90.6	83.3
Lie-Group + DTW + SVM [80]	95.29	83.87	98.22	92.46
Cov3DJ + SVM [85]*	88.04	89.29	94.29	90.53
SHMM(Joints)	68.57	65.89	67.39	67.28
DHMM(Joints)	72.14	68.95	80.72	73.94
HCRF (Hankelets)	58.5	60.79	65.36	61.55
SHMM (Hankelets)	86.76	88.75	92.79	89.43
DHMM (Hankelets)	88.62	94.18	91.70	91.5
DHMM-SL (Hankelets)	90.29	95.15	93.29	92.91

Table 3: Accuracy on the MSRA-3D action dataset – Protocol 2. *It excludes 13 noisy sequences. **Different splitting of the data. On average on the 3 subsets, our method achieves accuracy similar to that of [42] and [80]. On equal terms of feature representation and on average, our DHMM attains superior performance than the SHMM.



	hor. arm wave	hammer	forward punch	high throw	hand clap	bend	tennis serve	pickup throw
horizontal arm wave	100							
hammer		94.2						5.8
forward punch		0.9	78.2				10.9	10
high throw		7.3	7.3	63.6			18.2	3.6
hand clap					100			
bend						89.3		10.7
tennis serve							96.7	3.3
pickup throw		0.7				6.4		92.9

(a) MSRA-3D – AS1

	high arm wave	hand catch	draw X	draw tick	draw circle	2 hand wave	forward kick	side boxing
high arm wave	91.7		8.3					
hand catch	1.2	82.1	16.7					
draw X			100					
draw tick				100				
draw circle					100			
2 hand wave						100		
forward kick							100	
side boxing		10.5	1.9				2.9	84.7

(b) MSRA-3D – AS2

	high throw	forward kick	side kick	jogging	tennis swing	tennis serve	golf swing	pickup throw
high throw	65.7	1			11.1	16.2	2	4
forward kick		100						
side kick		21.2	78.8					
jogging		1.5		97		1.5		
tennis swing					100			
tennis serve						97		3
golf swing					0.7		99.3	
pickup throw	2.4							97.6

(c) MSRA-3D – AS3

Figure 6: Confusion matrices of the 3 subsets of 8 classes AS1, AS2 and AS3 of the MSRA-3D dataset in cross-subjects evaluation (results averaged on 10 runs).

6.4. Results on UCF

We performed the experiments on the UCF dataset using 4-fold cross-validation. We split the dataset into 4 sets and used 3/4 of the data to train the models and the remaining 1/4 to test them. The average accuracy of the 4 splits on 10 runs is reported in Table 4.

When comparing SHMM with DHMM on the 3D locations of the body joints, we get an increase of the accuracy of about 5.8%, indicating that on these features our discriminative learning approach outperforms the standard learning approach for HMMs. On the Hankelets, our method attains state-of-the-art performance. On these features, the discriminative learning improves the accuracy of the SHMM by about 0.8% and, jointly with the state learning procedure, the increase of the accuracy is near 1.22%.

In contrast to [18], which performs data normalization to account for cross-subjects biometric differences and noise, along with dimensionality reduction, our method does not require any pre-processing of the data.

Looking at the average confusion matrix in Fig. 7, we observe that most of the confusion is between pairs of classes *step back*/*step front*, *twist left*/*twist right* and *step left*/*step right*. These results are consistent with the ones obtained for the MSRA-3D dataset and highlight once more the limitation of the Hankelets in discriminating between action classes that share similar dynamics but involve movement in different directions.

6.5. Analysis of the Hankelet-based Representation

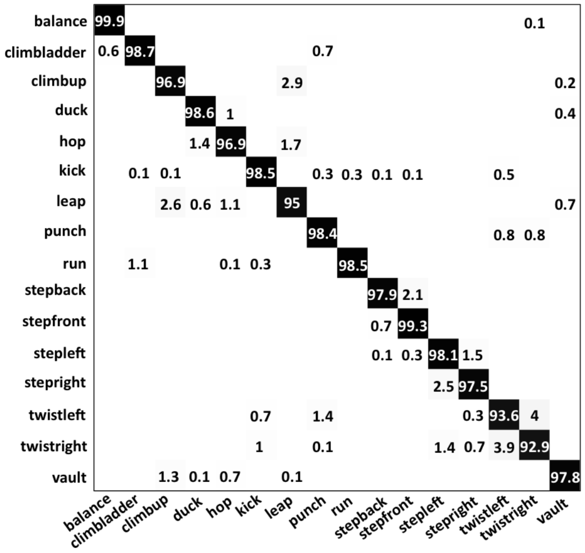
We have conducted experiments to study the effect of varying the parameters of the Hankelets on the recognition accuracy. Considering the time required for training a model and the fact that we have performed 1100 experiments, we restricted the analysis to the SHMM but we believe similar considerations also hold for our DHMM. All the results are reported in Fig. 8. In the horizontal axes of each plot we report the order and, within brackets, the number of frames used to compute the Hankel matrix. We recall that, as explained in Sec. 3, the number of frames τ used to compute a Hankel matrix is $\tau = n + m - 1$.

Methods	Accuracy
Log. Reg [77]	95.94
CRF [77]	94.29
BoW + SVM (distance) [77]	94.06
LTBSVM [18]*	97.91
SHMM(Joints)	56.80
DHMM(Joints)	60.12
HCRF (Hankelets)	53.31
SHMM (Hankelets)	96,48
DHMM (Hankelets)	97.27
DHMM-SL (Hankelets)	97.66

Table 4: Accuracy on all the 16 classes of the UCF dataset. * 70% of data used for training, 30% used in test. On equal terms of data splitting, our method achieves superior accuracy than [77]. Accuracy of [18] is not fully comparable due to a different splitting proportion of the dataset.

The first experiment measures the recognition accuracy while varying the order n of the square block-Hankel matrix in the range $[2, 10]$. As shown in Fig. 8(a), the recognition accuracy on the MSRA-3D dataset and on its subsets AS1 and AS3 increases for $n < 5$ and then starts to decrease. For the subset AS3, the accuracy decreases for $n > 6$. As for the UCF dataset, the accuracy always increases and tends to be consistently higher than 98% for $n > 6$, which means using more than 11 frames. The decrease of performance for higher orders in the MSRA-3D dataset may be a side-effect of the sliding window approach: by using longer sliding window, the switching of LTI systems can affect more windows, and this can make modeling the action more difficult.

Therefore, we have run another experiment where we set the order n to 2 and use longer temporal windows to compute the Hankel matrix. This means that the Hankel matrices are now rectangular, with more columns m than block rows n. The results of this experiment are shown in Fig. 8(b). With this setting,



	balance	climbladder	climbup	duck	hop	kick	leap	punch	run	stepback	stepfront	stepleft	stepright	twistleft	twistright	vault
balance	99.9														0.1	
climbladder	0.6	98.7						0.7								
climbup			96.9				2.9									0.2
duck				98.6	1											0.4
hop				1.4	96.9		1.7									
kick		0.1	0.1			98.5		0.3	0.3	0.1	0.1			0.5		
leap			2.6	0.6	1.1		95									0.7
punch								98.4						0.8	0.8	
run		1.1			0.1	0.3			98.5							
stepback										97.9	2.1					
stepfront										0.7	99.3					
stepleft										0.1	0.3	98.1	1.5			
stepright												2.5	97.5			
twistleft						0.7		1.4					0.3	93.6	4	
twistright						1		0.1				1.4	0.7	3.9	92.9	
vault			1.3	0.1	0.7		0.1									97.8

Figure 7: Confusion matrix of the 16 classes of the UCF dataset in 4-fold cross-validation (results averaged on 10 runs).

we may note a decrease in the performance for a number of frames higher than 9. On the UCF dataset the average accuracy is higher than 98% when more than 11 frames are used, which is consistent with the former experiment. This experiment suggests that the duration of the sliding window is a crucial parameter for the success of our approach and, considering the different impact of the duration on the subsets AS1, AS2 and AS3, it may be dependent on the action class. Looking at the results, we also note that, when using 19 frames to build a Hankel matrix of order 2, the performance on the subsets AS1 and AS2 decreases more than when using 19 frames to build a Hankel matrix of order 10. Therefore, the order is also an important parameter.

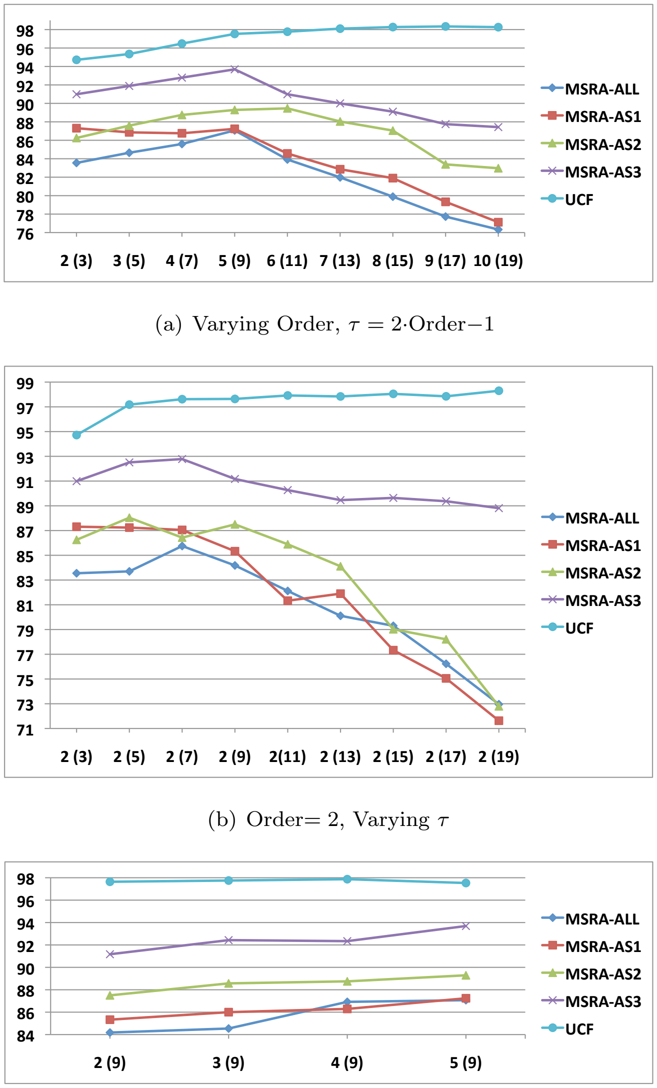
To gain more insights on the role of the order with respect to the duration of the sliding window, we have performed a further experiment, which we show in Fig. 8(c). In this case, we set the duration of the sliding window to 9 and vary the order in the range $[2, 5]$. The figure shows that increasing the order does increase the overall accuracy, even if this increment is of about $1 - 2\%$

on the MSRA-3D dataset. This might suggest that increased accuracy can be attained with Hankelet of higher order. However, at least 11 frames are required to build a Hankel matrix of order 6 and, as shown in Fig. 8(a) and 8(b), on the MSRA-3D dataset, using a number of frames higher than 9 results in an overall decrease in the accuracy.

6.6. Discussion

Overall, the experiments on the MSRA-3D dataset demonstrate that the discriminative learning and atomic LTI systems learning approaches result in improved classification accuracy for the HMM. It is interesting to note the poor performance of HCRF. In our implementation, we set the number of states of the HCRF equal to #classes$\cdot N$ with $N = 8$ in order to have the same number of states used by our classifier. However, since in HCRF all the classes share the same state space, this choice resulted in a higher number of parameters with respect to our model. Therefore this baseline might be slightly unfair. We have made other attempts to use HCRF together with the Hankelets (for example, by decreasing the number of states and training the model on the whole set of dissimilarity scores as input in order to reduce the number of parameters), but in all these experiments HCRF performed poorly. We believe that HCRF is more difficult to train than our model, particularly in the presence of a high number of states and classes.

Moreover, due to the large number of parameters, the time required to train HCRF is much higher than the time needed for training our DHMM. However, training an SHMM is faster than training a DHMM due to the fact that an SHMM is trained on a subset of the training set and requires fewer evaluations of the log-likelihood. Nonetheless, the classification procedure for the SHMM and DHMM is the same. As it is based on the Viterbi algorithm, its computational complexity is polynomial. To be more specific, the classification procedure has computational complexity of $O(C \cdot n^3)$ where $n = \max(\#\text{states}, \#\text{frames})$ and C is the number of action classes. The Hankel matrix computation requires a re-ordering of the input vector and a normalization by the Frobenius norm,



(a) Varying Order, $\tau = 2{\cdot}\text{Order}{-}1$

(b) Order= 2, Varying τ

(c) Varying Order, $\tau = 9$

Figure 8: Average accuracy of the SHMM classifier with varying parameters for building the Hankelets. On the x-axis, the first number is the order of the Hankel matrix; the number in parentheses is τ (the number of frames in the sliding window). In Fig. 8(a), increasing the order up to 5, the accuracy increases for all the datasets. On the MSRA-3D dataset, an order higher than 5 makes the accuracy to decrease. In Fig. 8(b), keeping the order set to 2 and increasing τ, accuracy on the MSRA-3D dataset decreases for τ higher than 9 frames. In Fig. 8(c), by keeping τ set to 9, increasing the order does increase the accuracy.

whose computational complexity is of about $O(m^3)$ with $m = \max(\#\text{rows of Hankelet}, \#\text{columns of Hankelet})$. A similar cost is required for computing the dissimilarity score of two Hankelets.

We believe that in our approach the most relevant parameter is the duration of the temporal window more than the order of the Hankel matrix. The experiments also support the idea of representing an action as a sequence of outputs of atomic LTI systems, where each atomic LTI system represents very simple dynamics (indeed an order equal to 2 is already effective). Our cross-subjects experiments on the MSRA-3D dataset suggest that the adopted representation and classification framework are quite robust to different speeds of the actions and, to some extent, temporal warping of the action sequences. This might be ascribable to the adoption of sliding windows of short duration with a HMM. With respect to other techniques for time series, such as dynamic time warping, the adoption of HMMs also helps to account for varying number of repetitions of body movements.

As for the discrepancy between the UCF dataset and the MSRA-3D dataset, we believe that the MSRA-3D dataset is noisier than the UCF dataset and this can have negative effects on the adopted dissimilarity score. Moreover, the UCF dataset has a larger number of training sequences, which allows for better estimates of the learned model parameters.

Finally, we want to highlight some difficulties we have encountered in evaluation using the MSRA-3D dataset. In our experiments, we retain the corrupted sequences in the training and test sets to guarantee a fair comparison with former works and we did not apply any interpolation of the data to reduce the noise. However, we noticed that some authors filter out the corrupted sequences from the dataset or ignore some classes. To make things worse, some works adopt arbitrary data splitting. To have a clear understanding of all these challenges, we refer the reader to [86], which presents an analysis of works that conduct evaluation using the MSRA-3D dataset. We stress that, in our experiments, we have adopted a cross-subjects validation where subjects 1, 3, 5, 7 and 9 are used for training the models while subjects 2, 4, 6, 8 and 10 are used to test the models. Filtering of noisy sequences and other splitting of the data would have probably resulted in higher accuracy.

7. Conclusions and Future Work

In this paper we have proposed to represent an action in terms of a sequence of outputs of atomic LTI systems. We represent each atomic LTI system by means of a representative Hankel matrix. We have adopted a discriminative HMM to model the transition from one LTI system to the next. We have also presented a novel method for learning the state representations (the atomic LTI systems) where our discriminative learning formulation encourages correct predictions of the models.

In experiments on two challenging action recognition benchmarks, our method achieves state-of-the-art accuracy by considering only the 3D trajectories of body joints. Our experimental results show that our discriminative learning approach seems to be more effective than the standard generative model learning approach for HMMs. However, our experiments have highlighted limitations of our Hankelet-based action representation when dealing with sequences that share the same dynamics but have different semantic labels due to differences in motion directions (such as moving towards left or towards right). Therefore, one possible future extension of our work could consider encoding such information within the action representation. In contrast to other works at the state-of-the-art, our technique does not require any pre-processing of the data to account for cross-subject biometric differences or varying duration of the action sequences.

A deep analysis of the impact on classification accuracy of varying settings for building the Hankelets has shown that the classification accuracy depends on the temporal duration of the sliding window used to compute the Hankelet. On the other hand, the order of the dynamical model had less influence on classification accuracy, and simple dynamical models suffice to get good classification performance for the datasets tested. This work does not deal with segmentation of the trajectory into atomic systems, which remains an interesting topic for future investigation. Under a generative point of view, a semi-Markov model could be employed, which would increase the overall computational complexity of the model.

The proposed framework is general and is not limited to action recognition. In future work we will study the possibility of applying this framework in other application domains such as event recognition and crowd analysis.

8. Acknowledgments

This work was partially supported by Italian MIUR grant PON0101687, SINTESYS - Security and INTElligence SYStem, and US NSF grant 1029430.

9. References

[1] S. Kwak, B. Han, J. Han, Scenario-based video event recognition by constraint flow, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3345–3352. `doi:10.1109/CVPR.2011.5995435`.

[2] U. Gaur, Y. Zhu, B. Song, A. Roy-Chowdhury, A string of feature graphs model for recognition of complex activities in natural videos, in: Proc. of International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2595–2602. `doi:10.1109/ICCV.2011.6126548`.

[3] S. Park, J. Aggarwal, Recognition of two-person interactions using a hierarchical Bayesian network, in: First ACM SIGMM international workshop on Video surveillance, ACM, 2003, pp. 65–76. `doi:10.1145/982452.982461`.

[4] I. Junejo, E. Dexter, I. Laptev, P. Pérez, View-independent action recognition from temporal self-similarities, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 33 (1) (2011) 172–185. `doi:10.1109/TPAMI.2010.68`.

[5] Z. Duric, W. Gray, R. Heishman, F. Li, A. Rosenfeld, M. Schoelles, C. Schunn, H. Wechsler, Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction, in: Proc. of the IEEE, Vol. 90, 2002, pp. 1272–1289. doi:10.1109/JPROC.2002.801449.

[6] Y.-J. Chang, S.-F. Chen, J.-D. Huang, A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities, Research in developmental disabilities 32 (6) (2011) 2566–2570. doi:10.1016/j.ridd.2011.07.002.

[7] A. Thangali, J. P. Nash, S. Sclaroff, C. Neidle, Exploiting phonological constraints for handshape inference in ASL video, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 521–528. doi:10.1109/CVPR.2011.5995718.

[8] A. Thangali Varadaraju, Exploiting phonological constraints for handshape recognition in sign language video, Ph.D. thesis, Boston University, MA, USA (2013).

[9] H. Cooper, R. Bowden, Large lexicon detection of sign language, in: Proc. of International Conference on Human–Computer Interaction, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 88–97. doi:10.1007/978-3-540-75773-3_10.

[10] J. M. Rehg, G. D. Abowd, A. Rozga, M. Romero, M. A. Clements, S. Sclaroff, I. Essa, O. Y. Ousley, Y. Li, C. Kim, et al., Decoding children's social behavior, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3414–3421. doi:10.1109/CVPR.2013.438.

[11] L. Lo Presti, S. Sclaroff, A. Rozga, Joint alignment and modeling of correlated behavior streams, in: Proc. of International Conference on Computer Vision-Workshops (ICCVW), 2013, pp. 730–737. doi:10.1109/ICCVW.2013.100.

[12] H. Moon, R. Sharma, N. Jung, Method and system for measuring shopper response to products based on behavior and facial expression, uS Patent 8,219,438 (Jul. 10 2012).
URL http://www.google.com/patents/US8219438

[13] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Communications of the ACM 56 (1) (2013) 116–124. doi:10.1145/2398356.2398381.

[14] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: Proc. of Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2012, pp. 20–27. doi:10.1109/CVPRW.2012.6239233.

[15] X. Yang, Y. Tian, Eigenjoints-based action recognition using Naive-Bayes-Nearest-Neighbor, in: Proc. of Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2012, pp. 14–19. doi:10.1109/CVPRW.2012.6239232.

[16] O. Oreifej, Z. Liu, W. Redmond, HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 716–723. doi:10.1109/CVPR.2013.98.

[17] G. Evangelidis, G. Singh, R. Horaud, et al., Skeletal quads: Human action recognition using joint quadruples, in: Proc. of International Conference on Pattern Recognition (ICPR), IEEE, 2014, pp. 4513–4518. doi:10.1109/ICPR.2014.772.

[18] R. Slama, H. Wannous, M. Daoudi, A. Srivastava, Accurate 3D action recognition using learning on the Grassmann manifold, Pattern Recognition (PR) 48 (2) (2015) 556–567. doi:10.1016/j.patcog.2014.08.011.

[19] J. Wang, Z. Liu, J. Chorowski, Z. Chen, Y. Wu, Robust 3D action recognition with Random Occupancy Patterns, in: Proc. of European Conference on Computer Vision (ECCV), Springer, 2012, pp. 872–885. doi:10.1007/978-3-642-33709-3_62.

[20] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1290–1297. doi:10.1109/CVPR.2012.6247813.

[21] N. Ikizler, D. Forsyth, Searching video for complex activities with finite state models, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2007, pp. 1–8. doi:10.1109/CVPR.2007.383168.

[22] R. Li, T.-P. Tian, S. Sclaroff, M.-H. Yang, 3D human motion tracking with a coordinated mixture of factor analyzers, International Journal of Computer Vision (IJCV) 87 (1-2) (2010) 170–190. doi:10.1007/s11263-009-0283-4.

[23] R. Li, T.-P. Tian, S. Sclaroff, Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series, in: Proc. of International Conference on Computer Vision (ICCV), IEEE, 2007, pp. 1–8. doi:10.1109/ICCV.2007.4409044.

[24] B. Li, O. I. Camps, M. Sznaier, Cross-view activity recognition using Hankelets, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1362–1369. doi:10.1109/CVPR.2012.6247822.

[25] B. Li, M. Ayazoglu, T. Mao, O. I. Camps, M. Sznaier, Activity recognition using dynamic subspace angles, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3193–3200. doi:10.1109/CVPR.2011.5995672.

[26] L. Lo Presti, M. La Cascia, S. Sclaroff, O. Camps, Gesture modeling by Hanklet-based hidden Markov model, in: D. Cremers, I. Reid, H. Saito, M.-H. Yang (Eds.), Computer Vision – ACCV 2014, Vol. 9005 of Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 529–546. doi:10.1007/978-3-319-16811-1_35.
URL http://dx.doi.org/10.1007/978-3-319-16811-1_35

[27] T. B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Computer Vision and Image Understanding (CVIU) 81 (3) (2001) 231–268. doi:10.1006/cviu.2000.0897.

[28] S. Mitra, T. Acharya, Gesture recognition: A survey, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37 (3) (2007) 311–324. doi:10.1109/TSMCC.2007.893280.

[29] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing (IMAVIS) 28 (6) (2010) 976–990. doi:10.1016/j.imavis.2009.11.014.

[30] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding (CVIU) 115 (2) (2011) 224–241. doi:10.1016/j.cviu.2010.10.002.

[31] J. Klamka, Controllability of dynamical systems. a survey, Bulletin of the Polish Academy of Sciences: Technical Sciences 61 (2) (2013) 335–342. doi:10.2478/bpasts-2013-0031.

[32] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, IEEE, 2005, pp. 886–893. doi:10.1109/CVPR.2005.177.

[33] P. Scovanner, S. Ali, M. Shah, A 3-dimensional SIFT descriptor and its application to action recognition, in: Proc. of Conference on Multimedia (MM), ACM, 2007, pp. 357–360. doi:10.1145/1291233.1291311.

[34] V. Kellokumpu, G. Zhao, M. Pietikäinen, Human activity recognition using a dynamic texture based method., in: Proc. of British Machine Vision Conference (BMVC), Vol. 1, BMVA Press, 2008, p. 2. doi:10.1.1.165.2894.

[35] J. C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, International Journal of Computer Vision (IJCV) 79 (3) (2008) 299–318. doi:10.1007/s11263-007-0122-4.

[36] I. Laptev, On space-time interest points, International Journal of Computer Vision (IJCV) 64 (2) (2005) 107–123. doi:10.1007/s11263-005-1838-7.

[37] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, International Journal of Computer Vision (IJCV) 103 (1) (2013) 60–79. doi:10.1007/s11263-012-0594-8.

[38] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Proc. of European Conference on Computer Vision (ECCV), Springer, 2006, pp. 428–441. doi:10.1007/11744047_33.

[39] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, in: Proc. of Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2010, pp. 9–14. doi:10.1109/CVPRW.2010.5543273.

[40] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, M. F. Campos, STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (2012) 252–259doi:10.1007/978-3-642-33275-3_31.

[41] A. Yao, J. Gall, G. Fanelli, L. J. Van Gool, Does human action recognition benefit from pose estimation?, in: Proc. of the British Machine Vision Conference (BMVC), Vol. 3, BMVA Press, 2011, pp. 67.1–67.11. doi:10.5244/C.25.67.

[42] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, Space-time pose representation for 3D human action recognition, in: Proc. of the International Conference on Image Analysis and Processing (ICIAP), Springer, 2013, pp. 456–464. doi:10.1007/978-3-642-41190-8_49.

[43] J. Sung, C. Ponce, B. Selman, A. Saxena, Unstructured human activity detection from RGBD images, in: Proc. of International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 842–849. doi:10.1109/ICRA.2012.6224591.

[44] Y. Zhu, W. Chen, G. Guo, Fusing spatiotemporal features and joints for 3D action recognition, in: Proc. of Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2013, pp. 486–491. doi:10.1109/CVPRW.2013.78.

[45] B. Bamieh, L. Giarre, Identification of linear parameter varying models, International Journal of Robust and Nonlinear Control 12 (9) (2002) 841–853. doi:10.1002/rnc.706.

[46] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, R. Vidal, Identification of hybrid systems a tutorial, European Journal of Control 13 (2) (2007) 242–260. doi:10.3166/ejc.13.242-260.

[47] E. D. Sontag, Nonlinear regulation: The piecewise linear approach, IEEE Transactions on Automatic Control 26 (2) (1981) 346–358. doi:10.1109/TAC.1981.1102596.

[48] V. Gupta, R. M. Murray, L. Shi, B. Sinopoli, Networked sensing, estimation and control systems, California Institute of Technology Report.

[49] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, Dynamic textures, International Journal of Computer Vision (IJCV) 51 (2) (2003) 91–109. doi:10.1023/A:1021669406132.

[50] C. Dicle, O. I. Camps, M. Sznaier, The way they move: Tracking multiple targets with similar appearance, in: Proc. of International Conference on

Computer Vision (ICCV), IEEE, 2013, pp. 2304–2311. doi:10.1109/ICCV.2013.286.

[51] A. C. Sankaranarayanan, P. K. Turaga, R. G. Baraniuk, R. Chellappa, Compressive acquisition of dynamic scenes, in: Proc. of European Conference on Computer Vision (ECCV), Springer, 2010, pp. 129–142. doi:10.1007/978-3-642-15549-9_10.

[52] L. Lo Presti, M. La Cascia, Using Hankel matrices for dynamics-based facial emotion recognition and pain detection, in: Proc. of Computer Vision and Pattern Recognition Workshops (AMFG–CVPRW), IEEE, 2015, pp. 1–8, in press.
URL http://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W08/papers/Presti_Using_Hankel_Matrices_2015_CVPR_paper.pdf

[53] L. Lo Presti, M. La Cascia, Ensemble of Hankel matrices for face emotion recognition, in: Proc. of International Conference on Image Analysis and Processing (ICIAP), Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2015, pp. 1–12, in press.

[54] F. Cuzzolin, M. Sapienza, Learning pullback HMM distances., IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 36 (7) (2014) 1483–1489. doi:10.1109/TPAMI.2013.181.

[55] H. Jiang, Discriminative training of HMMs for automatic speech recognition: A survey, Computer Speech & Language 24 (4) (2010) 589–608. doi:10.1016/j.csl.2009.08.002.

[56] T. Jebara, A. Pentland, Maximum conditional likelihood via bound maximization and the cem algorithm, in: Advances in Neural Information Processing Systems (NIPS), Vol. 1, The MIT Press, 1998, pp. 494–500.

[57] L. Bahl, P. Brown, P. De Souza, R. Mercer, Maximum mutual information estimation of hidden markov model parameters for speech recognition, in:

Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE, 1986, pp. 49–52. doi:10.1109/ICASSP.1986.1169179.

[58] R. Meir, Empirical risk minimization versus maximum-likelihood estimation: a case study, Neural computation 7 (1) (1995) 144–157. doi:10.1162/neco.1995.7.1.144.

[59] A. J. Smola, Advances in large margin classifiers, The MIT Press, 2000.

[60] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proc. of International Conference on Machine Learning (ICML), Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[61] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, T. Darrell, Hidden conditional random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29 (10) (2007) 1848–1852. doi:10.1109/TPAMI.2007.1124.

[62] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587597.

[63] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Large margin methods for structured and interdependent output variables, Journal of Machine Learning Research (JMLR) 6 (2005) 1453–1484.

[64] Y. Wang, G. Mori, Max-margin Hidden Conditional Random Fields for human action recognition, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 872–879. doi:10.1109/CVPR.2009.5206709.

[65] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3D exemplars, in: Proc. of International Conference on Computer Vision (ICCV), IEEE, 2007, pp. 1–7. doi:10.1109/ICCV.2007.4408849.

[66] F. Martinez-Contreras, C. Orrite-Urunuela, E. Herrero-Jaraba, H. Ragheb, S. A. Velastin, Recognizing human actions using silhouette-based HMM, in: Proc. of International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2009, pp. 43–48. doi:10.1109/AVSS.2009.46.

[67] F. Lv, R. Nevatia, Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost, in: Proc. of European Conference on Computer Vision (ECCV), Springer, 2006, pp. 359–372. doi:10.1007/11744085_28.

[68] T. Lan, Y. Wang, W. Yang, G. Mori, Beyond actions: Discriminative models for contextual group activities., in: Advances in Neural Information Processing Systems (NIPS), Vol. 4321, The MIT Press, 2010, pp. 4322–4325.

[69] A. D. Wilson, A. F. Bobick, Parametric Hidden Markov Models for gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 21 (9) (1999) 884–900. doi:10.1109/34.790429.

[70] L. E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, The annals of Mathematical Statistics 41 (1) (1970) 164–171. doi:10.1214/aoms/1177697196.

[71] F. Sha, L. K. Saul, Large margin Hidden Markov Models for automatic speech recognition, in: Advances in Neural Information Processing Systems (NIPS), Vol. 19, The MIT Press, 2007, p. 1249.

[72] M. Collins, Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms, in: Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP),

Vol. 10, Association for Computational Linguistics, 2002, pp. 1–8. doi:10.3115/1118693.1118694.

[73] B. Taskar, C. Guestrin, D. Koller, Max-margin markov networks, in: Advances in Neural Information Processing Systems (NIPS), Vol. 16, 2004, p. 25.

[74] Y. Altun, I. Tsochantaridis, T. Hofmann, et al., Hidden Markov support vector machines, in: Proc. of International Conference on Machine Learning (ICML), Vol. 3, 2003, pp. 3–10.

[75] L. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286. doi:10.1109/5.18626.

[76] J. Nocedal, S. J. Wright, Numerical Optimization (2nd edition), Springer, 2006.

[77] S. Z. Masood, C. Ellis, M. F. Tappen, J. J. LaViola, R. Sukthankar, Exploring the trade-off between accuracy and observational latency in action recognition, International Journal of Computer Vision (IJCV) 101 (3) (2013) 420–436. doi:10.1007/s11263-012-0550-7.

[78] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition, Journal of Visual Communication and Image Representation 25 (1) (2014) 24–38. doi:10.1016/j.jvcir.2013.04.007.

[79] J. Martens, I. Sutskever, Learning recurrent neural networks with Hessian-free optimization, in: Proc. of International Conference on Machine Learning (ICML), 2011, pp. 1033–1040.

[80] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3D skeletons as points in a Lie Group, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 588–595. doi:10.1109/CVPR.2014.82.

[81] C. Wang, Y. Wang, A. L. Yuille, An approach to pose-based action recognition, in: Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 915–922. doi:10.1109/CVPR.2013.123.

[82] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: Proc. of International Conference on Multimedia (MM), ACM, 2012, pp. 1057–1060. doi:10.1145/2393347.2396382.

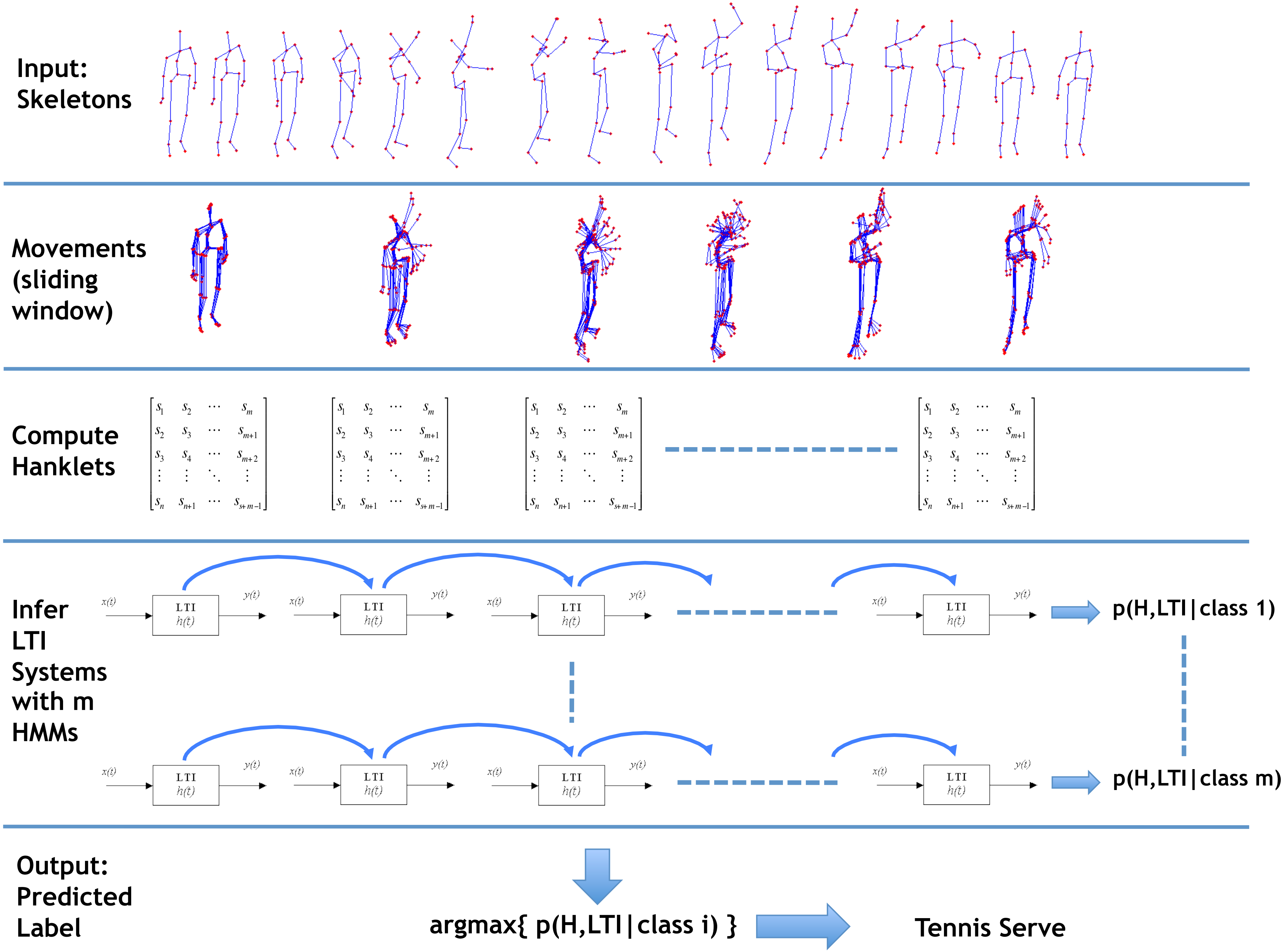
[83] E. Ohn-Bar, M. M. Trivedi, Joint angles similarities and HOG2 for action recognition, in: Proc. of Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2013, pp. 465–470. doi:10.1109/CVPRW.2013.76.

[84] N. Raman, S. J. Maybank, Action classification using a discriminative multilevel HDP-HMM, Neurocomputing,In press. doi:10.1016/j.neucom.2014.12.009.

[85] M. E. Hussein, M. Torki, M. A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations, in: Proc. of International Joint Conference on Artificial Intelligence (IJCAI), AAAI Press, 2013, pp. 2466–2472.

[86] J. R. Padilla-López, A. A. Chaaraoui, F. Flórez-Revuelta, A discussion on the validation tests employed to compare human action recognition methods using the MSR Action3D dataset, CoRR abs/1407.7390.
URL http://arxiv.org/abs/1407.7390

Graphical Abstract



Input: Skeletons

Movements (sliding window)

Compute Hanklets

$$\begin{bmatrix} s_1 & s_2 & \cdots & s_m \\ s_2 & s_3 & \cdots & s_{m+1} \\ s_3 & s_4 & \cdots & s_{m+2} \\ \vdots & \vdots & \ddots & \vdots \\ s_n & s_{n+1} & \cdots & s_{s+m-1} \end{bmatrix}$$

Infer LTI Systems with m HMMs

p(H,LTI|class 1)

p(H,LTI|class m)

Output: Predicted Label

argmax{ p(H,LTI|class i) }

Tennis Serve