Gesture Modeling by Hanklet-based Hidden Markov Model

Liliana Lo Presti[1], Marco La Cascia[1], Stan Sclaroff[2], and Octavia Camps[3]

[1] DICGIM, Universitá degli Studi di Palermo, Palermo, Italy
[2] Computer Science Department, Boston University, Boston, USA
[3] Dept. of Electrical and Computer Eng., Northeastern University, Boston, USA

Abstract. In this paper we propose a novel approach for gesture modeling. We aim at decomposing a gesture into sub-trajectories that are the output of a sequence of atomic linear time invariant (LTI) systems, and we use a Hidden Markov Model to model the transitions from the LTI system to another. For this purpose, we represent the human body motion in a temporal window as a set of body joint trajectories that we assume are the output of an LTI system. We describe the set of trajectories in a temporal window by the corresponding Hankel matrix (Hanklet), which embeds the observability matrix of the LTI system that produced it. We train a set of HMMs (one for each gesture class) with a discriminative approach. To account for the sharing of body motion templates we allow the HMMs to share the same state space. We demonstrate by means of experiments on two publicly available datasets that, even with just considering the trajectories of the 3D joints, our method achieves state-of-the-art accuracy while competing well with methods that employ more complex models and feature representations.

1 Introduction

The detection, recognition and analysis of gestures is of great interest for the computer vision community in well studied fields like surveillance [1], [2], [3], [4] and human-computer interaction [5] and in emerging fields like assistive technologies [6], computational behavioral science [7], [8] and consumer behavior analysis [9].

In this paper, we propose to represent a gesture as a temporal series of 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 motion descriptors (bag-of-words, histogram of flow, histogram of dense trajectories, etc.) within a temporal window.

As for the gesture temporal structure, there are dynamics regulating the sequence of motion templates; for example, handshaking may require the following ordered sequence of movements: moving the whole body for approaching the other person, raising the arm, and shaking the hand.

Many previous works have extracted global features for action recognition and trained models for each gesture-class [10], [11], [12], [13]. Some works have

focused on discriminative learning of models such as HMM [11], [14], [15] and CRF [16], [17], [18]. Most of them assume the gestures "live" in different state spaces. However, gestures may share body motion templates while having different temporal structures. In this paper, each body motion template is assumed to be the output of a linear time invariant (LTI) system and described by means of a Hankel matrix, which embeds the parameters of the LTI system [19].

A gesture is modeled by an HMM where the observations are Hankel matrices computed in a sliding window across time. In the following, we refer to such Hankel matrices as Hanklets. Each hidden state of the HMM represents an LTI system for which only a Hanklet is known. To account for the sharing of body motion templates, we train a set of gesture models that have the same state space but different dynamics, priors and conditional distributions over the observed Hanklets. The parameters of the gesture models are jointly learnt via a discriminative approach.

To summarize, the main contributions of this paper are:

- a novel gesture representation as sequence of Hanklets and
- a novel discriminative learning approach that allows different HMMs to share the same state space.

We show how a gesture can be modeled as a sequence of outputs from atomic LTI systems that are regulated by a Markov process. We describe each LTI system in terms of a Hankel matrix. This is different from other approaches such as [20], which represent body pose frame-by-frame. Instead the observations for our model are body motion templates with an intrinsic temporal duration.

To evaluate our method, we implemented a version of the formulation that takes 3D skeleton tracking data as input. Therefore, our body motion template is a set of trajectories of the 3D joints within a temporal window. Fig. 1 shows examples of body motion templates and gestures. As the figure highlights, gestures may share body motion templates. In experiments with two publicly-available gesture datasets, our approach attains state-of-the-art classification accuracies.

The rest of the paper is organized as follows. In Section 2 we report previous works in gesture recognition. In Section 3 we present our novel feature representation for body motion. In Section 4 we discuss our gesture model and present inference and learning approaches. In Section 5 we present experimental results. Finally, in Section 6 we present conclusions and future work.

2 Related Work

With the introduction of the Kinect sensor and the seminal work by Shotton, et al.[21] for estimating the joint locations of a human body, there has been a proliferation of works on gesture recognition. Most of these works introduce novel body pose representations. Some works [20], [22] use only the joint locations, while others [23], [24], mix descriptors from depth, motion and skeleton data. These works in general use state-of-the-art machinery to learn the temporal structure of gestures and/or to classify them.

Li et al. [24] proposed an action graph for depth action recognition. The depth map is projected onto three orthogonal Cartesian planes. A sub-sampled

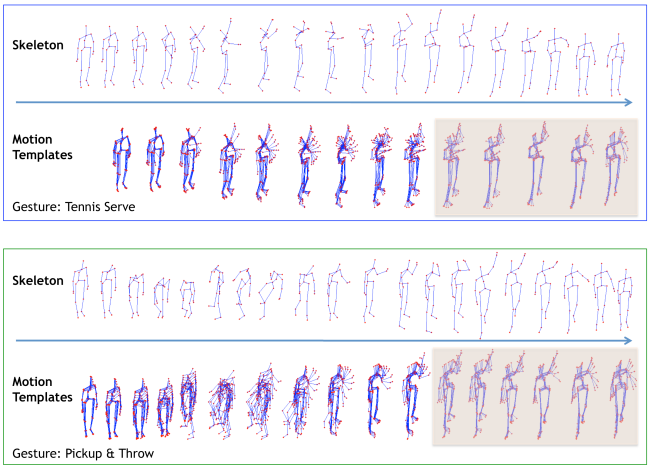

Fig. 1. Samples for two gestures from the MSRA3D-Action dataset. In each box, the first row shows the sequence of skeletons, the second row gives an idea of motion templates. Each image represents the super-imposition of skeletons detected in a temporal window. The last motion templates in the two gestures are generated by the same LTI-system during test; hence the corresponding action models share the same state.

set of points uniformly distributed are extracted and used as a bag of 3D points to encode the body pose. Each of these bags is a node in the action graph, which is used to model the dynamics of the actions. In Wang et al. [25], a 3D action sequence is treated as a 4D shape and a Random Occupancy Patterns (ROP) feature is extracted. Sparse coding and an Elastic-Net regularized classification model are used to classify the sequences. In Vieira et al. [26], space-time occupancy patterns are adopted to represent depth sequences. The features are computed by binarizing the space and time axes and computing what cells are occupied. Then a nearest neighbor classifier is applied for action recognition. In a similar way, Oreifej et al. [13] described the depth sequence as histograms of oriented surface normals (HON4D) captured in the 4D volume, based on depth and spatial coordinates. The quantization of the normals is non-uniform. Classification is performed by SVM classifier. In [27], 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; trajectories are compaired by an elastic distance between their corresponding points in shape space. Classification is performed by KNN on the Riemannian manifold. Other works focus on body pose representation of the given the 3D joint skeleton. In Xia et al. [20] a histogram of the locations of 12 manually selected 3D skeleton joints (HOJ3D) is computed to get a compact representation of the body pose invariant to the use of left and right limbs. LDA is used to project the histogram and compute K visual words used as states of an HMM. In [22], the body pose is represented by concatenating the distances between all the possible joint pairs in the current frame, the distances between the joints in the current frame and in the previous frame, the distances between the joints in the current frame and in a neutral pose. PCA is applied for dimensionality reduction providing a de-

scriptor called EigenJoints. Classification is performed by a naive-Bayes nearest neighbor classifier. In Wang et al. [23], depth data and the estimated 3D joint positions are used to compute the local occupancy pattern (LOP) feature. The set of features computed for a skeleton is called actionlet. Data mining techniques are used to discover the most discriminative actionlets. Finally, a multiple kernel learning approach is used to weight the actionlets.

Other methods combine the joint locations with visual information extracted from the RGB images. For example, Sung et al. [28] combined RGB, depth and hand positions, body pose and motion features extracted from skeleton joints. HOG[29] is used as the descriptor for both RGB and depth images. Then, a two-layer maximum-entropy Markov model is adopted for classification.

In contrast with previous works, we do not present a body pose representation. Instead we adopt the Hanklet representation [19] to describe body motion. Our method only uses body part trajectories (such as locations of joints in a skeleton) to represent a gesture as the output of a sequence of Linear Time Invariant (LTI) systems. We use an HMM to model the transition from one LTI system to another across time. HMM have been widely used for action modeling, for example in [20], [30], [31], [11], [32], [33]. Another model often used for action recognition is Conditional Random Fields (CRF)[16],[17],[18], which is a discriminative approach and has proven to be successful for recognition task. It has been shown that discriminative approaches tend to achieve better performance with respect to the standard HMM. Therefore, previous works [34],[35] have tried to learn the parameters of the HMM with a discriminative approach. In this paper, we adopt an HMM to model a gesture and learn its parameters in a discriminative way. The main difference between our approach and the standard HMM is that we allow classes' models to share the same state space.

Our approach is related to both linear parameter varying model identification [36] and switched system identification [37]. 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 observation. In this sense, our method is more similar to piecewise models and Markovian jump linear models [37], [38], [39] where there is a stochastic process that regulates the switching from one LTI system to another. Unlike previous methods [38], [39], our goal is not that of segmenting the sequence in outputs of different LTI systems; instead, we parse the sequence with a sliding window of fixed duration, and model probabilistically the switching among atomic LTI systems to capture the temporal structure of the whole gesture. Finally, there is an interesting connection with [40]. In [40], each video sequence is associated with a dynamical model. Then a metric is learned in order to optimally classify these dynamical models. Instead, 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.

3 Gesture Representation

We propose to represent a gesture as a sequence of body motion templates each one produced by an LTI system with unknown parameters. In our framework, each LTI system is represented by a Hanklet corresponding to an exemplar output. Associations between observed body motion templates and LTI systems is performed by comparing Hanklets.

Differently than methods like [13], [25], [24], we do not propose a body pose representation, but a new discriminative HMM model for gesture recognition. The novelty of our work stands in the decomposition of a gesture into atomic LTI systems by means of the decoding procedure used at inference time. In this sense, our method implicitly models the gesture as a switched dynamic system where each state is an LTI system. Furthermore, we formulate a discriminative HMM that can model the transition from one LTI system to another.

In the following we summarize the approach proposed in [41] to represent a trajectory and how we employ this descriptor for gesture representation.

3.1 Trajectory Representation by Hanklets

A trajectory may be represented as the output of a linear time invariant (LTI) system. LTIs are dynamic systems where the state and the measurement equations are linear, 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, while $y_k \in R^v$ is the v-dimensional measurement. To associate output measurements with the generating LTI system, we should apply system identification techniques to estimate the parameters of the LTI system, as in [42]. Instead, in our approach, we describe the trajectories produced by the dynamic system through a Hankel matrix. Given a sequence of output measurements $[y_o, \ldots, y_T]$ from (1), its associated (block) Hankel matrix is

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

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

As explained in [41], the Hankel matrix embeds the observability matrix Γ of the system, that is $H = \Gamma \cdot X$, where X is the sequence of hidden states of the system. Therefore H provides information about the dynamics of the temporal sequence. As H is also invariant to affine-transformations of the trajectory points [41], it is particularly appealing to adopt such a descriptor for gesture recognition.

In contrast with [41], which proposes a standard bag-of-words and SVM approach on Hanklet histograms, we propose to model the dynamics that regulates

sequences of Hanklets. We adopt a trajectory representation that is similar to the one used in [41]; while [41] computes a histogram of Hanklets for each action based on the set of detected dense trajectories, we compute a single Hanklet based on all the body joints together in a sliding window approach.

For the sake of demonstrating our idea, we use the 3D joints of the detected skeletons as input to our algorithm; the Hankel matrices are computed using the joint locations in a sliding window, where the temporal window is composed of T frames and the shift of the window happens frame by frame. We believe that the approach may be extended to the case when the skeleton is unknown, for example by detecting and tracking the body parts or correlated features (i.e. optical flow) from the RGB data. The same framework may also be used with frame-based body pose representations. These extensions remain a topic of future investigation.

3.2 Hanklet Computation and Comparison

A Hankel matrix is a powerful mathematical tool that embeds salient information about the dynamics of trajectories generated by LTI systems with unknown parameters. Hankel matrices have been successfully used in previous works on action recognition [41], tracking [43] and dynamic textures [42]. Our approach differs from these previous works in that we use the Hankel matrix space as an intermediary space where it is possible to compare body motion templates and LTI systems. In contrast to [41], which considers the velocities as measurements, we directly consider the joint locations as input measurements. We have empirically found that this representation is more informative than the one suggested in [41] for our gesture recognition task. We have also noticed that a better local representation (i.e. within the temporal window) is achieved by considering Hankel matrices with order lower than 5.

Given a temporal sequence $[y_o, \ldots, y_T]$, where y_t is a vector of the concatenated 3D joint locations in the skeleton at time t, and T is the number of frames in the temporal window, we center the sequence by taking off its average as in [41]. We compute the Hankel matrix and normalize it by its Frobenius norm. Our Hanklet representation for the given temporal sequence is the following:

$$H_p = \frac{H_p}{||H_p||_F}. \tag{3}$$

In contrast to [41], which considers the covariance matrix $H_p \cdot H_p'$ to represent a trajectory, we directly use the Hankel matrix H_p. The matrix $H_p \cdot H_p'$ is invariant to the direction in which the state changes and may not be suitable for gesture recognition. The Frobenius norm may be computed as:

$$||H_p||_F = \sqrt{\sum_{i,j}(H_p(i,j)^2)}. \tag{4}$$

Once we represent the trajectories by means of their corresponding Hankel matrices, we need a way to establish if two trajectories have been generated by the same LTI system or not. We do this by comparing their Hankel matrices. To

convey the degree to which two Hanklets may be considered similar, we use an approximate score similar to that proposed in [41], defined as follows:

$$d(H_p, H_q) = 2 - ||H_p + H_q||_F. \tag{5}$$

4 Hanklet-based Hidden Markov Model

We assume that a gesture is a sequence of body motion templates produced by a set of LTI systems. Each LTI system is represented by a Hanklet S of an exemplar output sequence that the system has produced. The probability that a given sequence of measurements is produced by an LTI system is modeled by the following exponential distribution:

$$p(H|S) = \lambda \cdot e^{-\lambda \cdot d(H,S)} \tag{6}$$

where H is the Hanklet corresponding to the given sequence of measurements, S is the Hanklet used for representing the LTI system, $d(H, S)$ is the dissimilarity score in Eq. (5), λ is a parameter to learn.

We assume that the measurements in a gesture come from a sequence of LTI systems. The switching process that generates a gesture is assumed to be a Markovian process and therefore we employ an HMM to model the transitions from one LTI system to another. The transition matrix T is a stochastic matrix where $T(i, j) = p(S_t^j|S_{t-1}^i)$, and is a parameter of the model. The prior probability π, such that $\pi(i) = p(S_0^i)$, is the probability that the measurement in the first temporal window ($t = 0$) has been generated by the i-th LTI model.

Given these definitions, the joint probability of the sequence of N observed Hankel matrices $H = \{H_t\}_{t=0}^N$ (computed from the observations) and the sequence of LTI systems, represented by means of the corresponding Hanklets $S = \{S_t\}_{t=0}^N$ is:

$$p(H, S|T, \pi, \Lambda) = \prod_{t=0}^{N} p(H_t|S_t) \cdot \prod_{t=1}^{N} P(S_t|S_{t-1}) \cdot \pi(S_0) \tag{7}$$

where $\Lambda = \{\lambda_S\}$ is the set of parameters λ associated with each state.

4.1 Inference and Classification

Given a gesture model with parameters $\{\Lambda^c, T^c, \pi^c\}$, where c is the label of the gesture to which the model refers, the inference of the sequence of LTI-systems is performed via the Viterbi algorithm[44]. 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 the label to assign to a sequence of measurements is performed by maximum likelihood. The predicted label C_P is computed solving:

$$C_P = \min_c \{-\log p(H, S^c|T^c, \pi^c, \Lambda^c)\}. \tag{8}$$

The label corresponding to the model providing the highest likelihood is assigned to the sequence of observations.

Algorithm 1: Inference of Gesture-Class

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

for $i \leftarrow 1$ **to** M **do**
  for $j \leftarrow 1$ **to** T **do**
    $\mathrm{D}(i,j) \leftarrow d(S_i, H_j)$ (eq. 5);
  end
end
for $c \leftarrow 1$ **to** N **do**
  $\mathrm{LL}(c) \leftarrow$applyViterbi$(D, \Lambda^c, T^c, \pi^c)$
end

$C_P \leftarrow \mathrm{argmin}(LL)$

Algorithm 1 shows how the classification of an input Hanklet sequence is performed. As all the models share the state space, it is necessary to compute the matrix of dissimilarity scores between the Hanklets and the shared states only once. Then the Viterbi algorithm is applied N times, once for each gesture class. The negative log-likelihood score is normalized to account for different lengths of the sequences.

4.2 Discriminative Learning

The traditional learning approach for the HMM parameters uses the Baum-Welch algorithm [44]. In many applications, e.g. [34],[35], it has been demonstrated that discriminative learning of the HMM parameters results in better performance. We therefore apply this approach to the parameter learning of our models. The learning procedure learns the parameters of all the HMMs simultaneously while encouraging correct predictions and penalizing the wrong ones.

Discriminative learning tries to minimize the mis-classification measure for an input training sample H defined as follows:

$$loss(H) = \max\{0, g^k(H, \Lambda^k, T^k, \pi^k) - \min_{j \neq k}\{g^j(H, \Lambda^j, T^j, \pi^j)\} + 1\} \quad (9)$$

where g^k represents the negative log-likelihood returned by the correct model k, and $\min_j\{g^j\}$ is the negative log-likelihood of the most competitive but incorrect model. The difference of these two terms represents the margin of the classifier and the loss function in Eq. 9 is the hinge loss. Minimizing this mis-classification error corresponds to increasing the inter-class distances on a training set. Whenever the loss is greater than 0, then the prediction is incorrect and it is necessary to update the parameters. The negative log-likelihood g is defined as:

$$g(H, \Lambda, T, \pi) = -\log(p(H, S|T, \pi, \Lambda)) = \quad (10)$$

$$-\sum_{t=0}^{N} log(p(H_t|S_t)) - \sum_{t=1}^{N} \log(P(S_t|S_{t-1})) - \log(\pi(S_0)), \quad (11)$$

which may be written as:

$$g(H, \Lambda, T, \pi) = \sum_{t=0}^{N} (\lambda_{S_t} \cdot d(H_t|S_t) - \log(\lambda_{S_t}) - \sum_{t=1}^{N} \alpha_{S_t|S_{t-1}} - \beta_{S_0}, \tag{12}$$

where the variables α and β represent the logarithms of the transition probabilities and priors respectively, and we use Eq. (6) for the observation probabilities. As the priors and the transition probabilities must be positive and must sum to one, the original optimization problem should be constrained. As in previous works, such as [45], we consider that α and β do not have any constraints and perform the optimization directly on these variables. Before doing inference, these variables are transformed back to obtain the parameters of the model π and T. In particular, for π we get:

$$\pi(S) = \frac{e^{\beta_S}}{\sum_s e^{\beta_s}}. \tag{13}$$

A similar transformation holds for T.

We minimize the loss over all the samples in the training set via a quasi-Newton strategy with limited-memory BFGS updates where a block-coordinate descent approach is used in turn for updates to the parameters Λ, T and the prior π. In practice we have observed that the block-coordinate descent results in faster convergence of the training procedure.

Algorithm 2 shows the pseudo-code for our training procedure. After initializing all the models with the same parameters (uniform distributions for T and π and 1 for λ), the method iteratively minimizes the objective function $f(\cdot)$ by block-coordinate descent. The function check_convergence() checks if some convergence criteria is met. The variable p_set is used to identify the active parameter subset, that is the subset of parameters considered when minimizing the objective function within the block-coordinate schema. Algorithm 3 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 the negative log-likelihood returned by the most likely incorrect model. If the loss is positive, then the models have produced a wrong prediction; therefore the gradients are accumulated and returned to the L-BFGS algorithm to update the parameters. The variable p_set allows us to accumulate the gradients only for the current active parameter subset.

4.3 Initialization of the State Space

The state space initialization is performed by considering, for each class, a subset of video sequences in the training set. Then the corresponding Hanklets are clustered via K-medoids and the K medoids are used to compose the state space. Therefore, given N classes, the state space has a dimensionality equal to $K \cdot N$.

We have tested two strategies: online learning the state representation while learning the parameters of the model versus not learning the state representation. Learning of the state is done by re-clustering the Hankel matrices, that is, for

Algorithm 2: Discriminative learning of the Parameters

Input : $\{Y_i\}_{i=1}^{W}$: training set of Hanklet sequences;
labels: gesture-classes for each training sequence;
$\{S_i\}_{i=1}^{M}$ state space
Output: $\{\Lambda^c\}_{c=1}^{N}$, $\{T^c\}_{c=1}^{N}$, $\{\pi^c\}_{c=1}^{N}$ parameters of the HMMs;

%% Parameter initialization;
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 $\leftarrow false$;

%% Apply Block-Coordinate Gradient Descent to each active parameter subset (p_set);
while *iter* $<$ *Max_Iter* & *!converged* **do**
  %% Optimize with respect to $\{\Lambda^c\}_{c=1}^{N}$;
  p_set $\leftarrow$ lambdas;
  $\{\Lambda^c\}_{c=1}^{N} \leftarrow \text{argmin}\, f(\{Y_i\}_{i=1}^{W}, \text{labels}, \{\Lambda^c\}_{c=1}^{N}, \{T^c\}_{c=1}^{N}, \{\pi^c\}_{c=1}^{N}, \{S_i\}_{i=1}^{M},$ p_set);

  %% Optimize with respect to $\{T^c\}_{c=1}^{N}$;
  p_set $\leftarrow$ transition matrices;
  $\{T^c\}_{c=1}^{N} \leftarrow \text{argmin}\, f(\{Y_i\}_{i=1}^{W}, \text{labels}, \{\Lambda^c\}_{c=1}^{N}, \{T^c\}_{c=1}^{N}, \{\pi^c\}_{c=1}^{N}, \{S_i\}_{i=1}^{M},$ p_set);

  %% Optimize with respect to $\{\pi^c\}_{c=1}^{N}$;
  p_set $\leftarrow$ priors;
  $\{\pi^c\}_{c=1}^{N} \leftarrow \text{argmin}\, f(\{Y_i\}_{i=1}^{W}, \text{labels}, \{\Lambda^c\}_{c=1}^{N}, \{T^c\}_{c=1}^{N}, \{\pi^c\}_{c=1}^{N}, \{S_i\}_{i=1}^{M},$ p_set);

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

each state we consider all the observed Hankel matrices that have been generated by that state and we compute the medoid of this set of matrices. Our experiments have shown a small change in the performance when learning vs non-learning the state representation. As learning the state representation increases the time complexity, in this paper we do not update the state space online.

In our implementation the number of states is defined a priori. Computing the state space by allowing the introduction of new states, merging/removing of existing states could be certainly done, e.g. using a reversible jump Markov chain Monte Carlo[46] to decide when to add/merge/remove a state. However, this is beyond the scope of this paper and remains a topic for future investigation.

Algorithm 3: f() : Objective Function to Minimize

Input : $\{Y_i\}_{i=1}^{W}$: training set of Hanklet sequences;
labels: gesture-classes for each training sequence;
$\{S_i\}_{i=1}^{M}$ state space;
$\{\Lambda^c\}_{c=1}^{N}$, $\{T^c\}_{c=1}^{N}$, $\{\pi^c\}_{c=1}^{N}$ parameters of the HMMs;
p_set: active parameter subset
Output: Cum_loss: loss over all the samples in the dataset; Grad: gradients

%% Accumulate loss for all the sequences in the training set;
Cum_loss $\leftarrow 0$;
for $i \leftarrow 1$ **to** W **do**
  %% Compute loss for the i-th sequence;
  D $\leftarrow$ compute dissimilarity score matrix between Y_i and $\{S_i\}_{i=1}^{M}$;
  $k \leftarrow$ labels(i);
  $[g^k, z^k] \leftarrow$ applyViterbi$(D, \Lambda^k, T^k, \pi^k)$;
  $[g^c, z^c] \leftarrow \min_{c \neq k}$ applyViterbi$(D, \Lambda^c, T^c, \pi^c)$;
  loss $\leftarrow \max(0, g^k - g^c + 1)$;
  Cum_loss $\leftarrow$ Cum_loss + loss;

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

5 Experiments

We evaluated our method on two datasets: MSRAction3D [23] and UTKinect-Action [20]. The first dataset provides skeleton and depth data; examples of the detected skeletons are shown in Fig. 1. The second dataset also provides RGB data and sample images are shown in Fig. 2. We used only the skeleton data; such data are corrupted by various levels of noise, which affects the recognition accuracy. The code of our implementation has been written in Matlab[4].

5.1 Setting of Parameters and Initialization

The maximal order of the LTI systems (that is the number of rows of the Hankel matrix) determines the minimal length of the temporal window needed to build a square Hankel matrix (8 if the order is 3; 15 if the order is 4). In the datasets we used, some sequences contain fewer than 15 frames. To guarantee a fair comparison with previous methods we have chosen to set the order to 3 instead of removing the shorter training/test sequences.

The state space has been computed by applying the K-medoid algorithm to subsets of Hanklets. For each class we have selected randomly 20 videos and we

[4] Code available at http://www.dicgim.unipa.it/ cvip/people/lopresti/.

Fig. 2. Samples from the UTKinect-Action dataset (source:UTKinect-Action website)

Methods	[47]*	[10]*	[11]*	[12]	[24]	[48]**	[13]**	[25]	[23]	[13]**	**Ours**
Accuracy:	42.5%	54%	63%	65.7%	74.7%	85.5%	85.8%	86.5%	87.2%	88.9%	**89%**

Table 1. Accuracy on the MSRA-3D action dataset. * Results reported in [12]. ** Different splitting of training and test set. We note that [10] uses dynamic time warping, while [11] uses a standard HMM.

have clustered the oberved Hankel matrices. The number of centroids K has been set to 5. Thus, we used a state space composed of 100 Hankel matrices for the MSRA-3D dataset and 50 Hankel matrices for the UTKinect-Action dataset.

5.2 Experiments on the MSRA3D dataset

The MSRAction3D dataset [5] provides the skeleton (20 joints) for 20 gestures performed 2 or 3 times by 10 subjects. The dataset contains 3D coordinates from 557 sequences of the following gestures: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup and throw.

We use the same setting reported on the authors' website: 10 sequences have been filtered out because of the excessive noise on the skeletons; 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.

Table 1 shows the comparison between our proposed approach and previous works in terms of classification accuracy (number of correctly classified sequences over number of sequences). On this dataset our method performs the best. Table 2 and 3 show the confusion matrix and the classification accuracies per class respectively. For half of the actions, our method attains 100% of accuracy. Only for some classes, namely High Arm Waving and Hand Clap, the performance decreases. As for the action High Arm Waving, most of the confusion is with Horizontal Arm Waving. The decrease of performance in this case may be ascribable to the fact we are not considering the relative position of the 3D joints when training our models. Most of the previous works we compare to, i.e. [23] or [13], use more complicated feature representation or machinery. In contrast, we only use information about the dynamics of the 3D skeleton joints.

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

T vs P	HW	HoW	Ham	HC	FP	HT	DX	DT	DC	HC	2HW	SB	Bend	FK	SK	Jog	TSw	TSe	GSw	P-T
HW	**58.3**	33.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8.3	0	0	0
HoW	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ham	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HC	0	8.3	8.3	**58.3**	16.7	0	0	0	0	0	0	0	0	0	0	0	8.3	0	0	0
FP	0	0	0	0	**63.6**	0	0	9.1	0	0	0	0	0	0	0	0	0	27.3	0	0
HT	9.1	0	0	0	0	**63.6**	0	0	0	0	0	0	0	0	0	0	0	0	27.3	0
DX	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0
DT	6.7	0	0	0	0	0	0	**93.3**	0	0	0	0	0	0	0	0	0	0	0	0
DC	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0
HC	0	0	0	0	0	0	0	0	0	**93.3**	6.7	0	0	0	0	0	0	0	0	0
2HW	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0
SB	6.7	0	0	0	0	0	0	0	0	0	0	**73.3**	0	6.7	0	0	13.3	0	0	0
Bend	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0
FK	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0
SK	0	0	0	0	0	0	0	0	0	0	0	0	0	9.1	**90.9**	0	0	0	0	0
Jog	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0	0
TSw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0	0
TSe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0	0
GSw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0
P-T	0	0	0	0	0	0	0	0	0	0	0	0	35.7	0	0	0	0	0	0	**64.3**

Table 2. Confusion matrix for the MSRA-3D dataset.

Acc.	HW	HoW	Ham	HC	FP	HT	DX	DT	DC	HC	2HW	SB	Bend	FK	SK	Jog	TSw	TSe	GSw	P-T
[12]	NA	NA	0	0	NA	14.3	35.7	NA	20	**100**	**100**	NA	NA	**100**	NA	NA	NA	**100**	**100**	NA
[23]	**91.7**	**100**	83.9	25	**72.7**	**72.7**	53.8	**100**	**100**	**100**	**100**	86.7	93.3	**100**	**100**	**100**	**100**	**100**	**100**	**64.3**
Ours	58.3	**100**	**100**	**58.3**	63.6	63.6	**100**	93.3	**100**	93.3	**100**	73.3	**100**	**100**	90.9	**100**	**100**	**100**	**100**	**64.3**

Table 3. Accuracy on the MSRA-3D dataset per class.

5.3 Experiments on the UTKinect-Action dataset

The UTKinect-Action dataset [6] provides the skeleton (20 joints) for 10 actions performed twice by 10 subjects. The dataset contains 200 sequences of the following gestures: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands. Six sequences were too short to compute the Hankel matrices and have been filtered out. As done in [20], we performed the experiments in leave-one-out cross-validation (LOOCV). In this dataset, one of the subject is left-handed and there is a very high variance in the length of the sequences (the length ranges from 5 to 120 frames). Moreover, there is a significant variation among different realizations of the same action: some actors pick up objects with one hand, while others pick up the objects with both hands. The individuals can toss an object with either their right or left arm, producing different trajectories. Finally, actions have been taken from different views and, therefore, the body orientation varies.

As shown in Table 4, our method attains performance that approaches that reported by [20], [27]. Considering the challenges in this dataset, and the limited number of sequences available for training, the accuracy we get is quite high. Accuracy is somewhat limited in this experiment because the Hanklets are sensitive to the order the joints are considered, therefore it cannot discriminate between two samples of the same action in which one involves a left limb and the other one a right limb (i.e. in the class Throw).

In this experiment, we use the same joints as in [20]. We center the points on the hip center at each frame and use the remaining 11 joints to compute the descriptors. Table 5 reports the confusion matrix for the 10 classes. Most of the confusion is between the actions walk and carry. This is probably due to the fact

[6] http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html

Accuracy	Walk	SitDown	StandUp	PickUp	Carry	Throw	Push	Pull	WHands	CHands	Avr
[27]	90%	**100%**	**100%**	**100%**	68.4%	**95%**	**90%**	**100%**	**100%**	80%	**91.5%**
[20]	**96.5%**	91.5%	93.5%	97.5%	**97.5%**	59%	81.5%	92.5%	**100%**	**100%**	90.9%
Ours	63.16%	**100%**	**100%**	**100%**	83.33%	61.11%	**90%**	**100%**	85%	85%	86.76%

Table 4. Accuracy on the UTKinect-Action dataset.

True vs Predicted	Walk	SitDown	StandUp	PickUp	Carry	Throw	Push	Pull	WaveHands	ClapHands
Walk	**63.16%**	0	0	0	31.58%	0	5.26%	0	0	0
SitDown	0	**100%**	0	0	0	0	0	0	0	0
StandUp	0	0	**100%**	0	0	0	0	0	0	0
PickUp	0	0	0	**100%**	0	0	0	0	0	0
Carry	16.67%	0	0	0	**83.33%**	0	0	0	0	0
Throw	0	0	5.56%	5.56%	0	**61.11%**	16.67%	0	5.56%	5.56%
Push	5%	0	0	0	0	5%	**90%**	0	0	0
Pull	0	0	0	0	0	0	0	**100%**	0	0
WaveHands	0	0	0	0	0	0	0	0	**85%**	15%
ClapHands	0	5%	0	0	0	0	0	0	10%	**85%**

Table 5. Confusion matrix for the UTKinect-Action dataset.

that these actions, in terms of the dynamics of many of the joints involved in the actions, are pretty indistinguishable. In such cases, features capturing the 3D joint spatial configuration may help to disambiguate.

6 Conclusions and Future Work

We have proposed a novel representation of a gesture in terms of temporal sequence of body motion templates. We have assumed that each motion template represents the output of an atomic LTI system and can be represented by a Hankel matrix. We have adopted a discriminative HMM to model the transition from one LTI system to the next. We have allowed the discriminative HMMs to share the same state space. This enables the gesture models to share LTI systems and, therefore, body motion templates.

In experiments on two challenging gesture recognition benchmarks, our method achieves state-of-the-art accuracy by considering only 3D joint trajectories. The experiments suggest that dynamics of a suitable body pose/shape descriptor may help to disambiguate in cases where 3D joints dynamics are too similar.

In future work, we will extend the Hanklet-based representation in order to account for the temporal warping in the observed data. A limitation of the Hanklet is that it is sensitive to the order the joints are used when computing the Hankel matrix. This is problematic in cases where the same gesture can be performed either with left or right limbs. We will investigate new techniques to formulate the Hankel matrix that may overcome these limitations.

As for our discriminative HMM, we will investigate techniques that enable the state space to adapt online by adding, removing or merging existing states. We will also investigate the use of more complex dynamic Bayesian networks to account for the temporal warping and switching of the LTI systems, thereby removing the need for a sliding window approach.

Acknowledgement. This work was partially supported by Italian MIUR grant PON01 01687, SINTESYS - Security and INTElligence SYStem and by US NSF grant 1029430.

References

1. Kwak, S., Han, B., Han, J.: Scenario-based video event recognition by constraint flow. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3345–3352
2. Gaur, U., Zhu, Y., Song, B., Roy-Chowdhury, A.: A string of feature graphs model for recognition of complex activities in natural videos. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2595–2602
3. Park, S., Aggarwal, J.: Recognition of two-person interactions using a hierarchical Bayesian network. In: First ACM SIGMM international workshop on Video surveillance, ACM (2003) 65–76
4. Junejo, I., Dexter, E., Laptev, I., Pérez, P.: View-independent action recognition from temporal self-similarities. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011) 172–185
5. Duric, Z., Gray, W., Heishman, R., Li, F., Rosenfeld, A., Schoelles, M., Schunn, C., Wechsler, H.: Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. Proc. of the IEEE (2002)
6. Chang, Y.J., Chen, S.F., Huang, J.D.: A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in developmental disabilities **32** (2011) 2566–2570
7. Rehg, J.M., Abowd, G.D., Rozga, A., Romero, M., Clements, M.A., Sclaroff, S., Essa, I., Ousley, O.Y., Li, Y., Kim, C., et al.: Decoding children's social behavior. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 3414–3421
8. Lo Presti, L., Sclaroff, S., Rozga, A.: Joint alignment and modeling of correlated behavior streams. In: The IEEE International Conference on Computer Vision (ICCV) Workshops. (2013)
9. Jung, N., Moon, H., Sharma, R.: Method and system for measuring shopper response to products based on behavior and facial expression (2012) US Patent 8,219,438.
10. Müller, M., Röder, T.: Motion templates for automatic classification and retrieval of motion capture data. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, Eurographics Association (2006) 137–146
11. Lv, F., Nevatia, R.: Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost. In: Computer Vision–ECCV 2006. Springer (2006) 359–372
12. Masood, S.Z., Ellis, C., Tappen, M.F., LaViola, J.J., Sukthankar, R.: Exploring the trade-off between accuracy and observational latency in action recognition. International Journal of Computer Vision **101** (2013)
13. Oreifej, O., Liu, Z., Redmond, W.: HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In: Computer Vision and Pattern Recognition (CVPR). (2013)
14. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using Hidden Markov Model. In: Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on, IEEE (1992) 379–385
15. Starner, T., Pentland, A.: Real-time American Sign Language recognition from video using Hidden Markov Models. In: Motion-Based Recognition. Springer (1997) 227–243

16. Wang, Y., Mori, G.: Max-margin Hidden Conditional Random Fields for human action recognition. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 872–879
17. Vail, D.L., Veloso, M.M., Lafferty, J.D.: Conditional Random Fields for activity recognition. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, ACM (2007) 235
18. Wang, S.B., Quattoni, A., Morency, L., Demirdjian, D., Darrell, T.: Hidden Conditional Random Fields for gesture recognition. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 2., IEEE (2006) 1521–1527
19. Li, B., Ayazoglu, M., Mao, T., Camps, O.I., Sznaier, M.: Activity recognition using dynamic subspace angles. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3193–3200
20. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3D joints. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, IEEE (2012) 20–27
21. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56** (2013) 116–124
22. Yang, X., Tian, Y.: Eigenjoints-based action recognition using Naive-Bayes-Nearest-Neighbor. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, IEEE (2012) 14–19
23. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1290–1297
24. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, IEEE (2010) 9–14
25. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3D action recognition with Random Occupancy Patterns. In: Computer Vision–ECCV 2012. Springer (2012) 872–885
26. Vieira, A.W., Nascimento, E.R., Oliveira, G.L., Liu, Z., Campos, M.F.: Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer (2012) 252–259
27. Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., Del Bimbo, A.: Space-time pose representation for 3D human action recognition. In: New Trends in Image Analysis and Processing–ICIAP 2013. Springer (2013) 456–464
28. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from RGBD images. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE (2012) 842–849
29. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 1., IEEE (2005) 886–893
30. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3D exemplars. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE (2007) 1–7
31. Martinez-Contreras, F., Orrite-Urunuela, C., Herrero-Jaraba, E., Ragheb, H., Velastin, S.A.: Recognizing human actions using silhouette-based HMM. In: Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on, IEEE (2009) 43–48

32. Lan, T., Wang, Y., Yang, W., Mori, G.: Beyond actions: Discriminative models for contextual group activities. In: NIPS. Volume 4321. (2010) 4322–4325
33. Wilson, A.D., Bobick, A.F.: Parametric Hidden Markov Models for gesture recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **21** (1999) 884–900
34. Sha, F., Saul, L.K.: Large margin Hidden Markov Models for automatic speech recognition. Advances in neural information processing systems **19** (2007) 1249
35. Collins, M.: Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics (2002) 1–8
36. Bamieh, B., Giarre, L.: Identification of linear parameter varying models. International Journal of Robust and Nonlinear Control **12** (2002) 841–853
37. Paoletti, S., Juloski, A.L., Ferrari-Trecate, G., Vidal, R.: Identification of hybrid systems a tutorial. European journal of control **13** (2007) 242–260
38. Sontag, E.D.: Nonlinear regulation: The piecewise linear approach. Automatic Control, IEEE Transactions on **26** (1981) 346–358
39. Gupta, V., Murray, R.M., Shi, L., Sinopoli, B.: Networked sensing, estimation and control systems. California Institute of Technology Report (2009)
40. Cuzzolin, F., Sapienza, M.: Learning pullback HMM distances. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
41. Li, B., Camps, O.I., Sznaier, M.: Cross-view activity recognition using Hankelets. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1362–1369
42. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. International Journal of Computer Vision **51** (2003) 91–109
43. Dicle, C., Camps, O.I., Sznaier, M.: The way they move: Tracking multiple targets with similar appearance. (2013) 2304–2311
44. Rabiner, L.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–286
45. Chang, P.C., Juang, B.H.: Discriminative training of dynamic programming based speech recognizers. Speech and Audio Processing, IEEE Transactions on **1** (1993) 135–143
46. Green, P.J.: Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. Biometrika **82** (1995) 711–732
47. Martens, J., Sutskever, I.: Learning recurrent neural networks with Hessian-free optimization. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). (2011) 1033–1040
48. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on Multimedia, ACM (2012) 1057–1060