Metric Learning from Poses for Temporal Clustering of Human Motion

Adolfo López-Méndez[1]
adolf.lopez@upc.edu
Juergen Gall[2]
juergen.gall@tue.mpg.de
Josep R. Casas[1]
josep.ramon.casas@upc.edu
Luc van Gool[3]
vangool@vision.ee.ethz.ch

[1] Technical University of Catalonia (UPC)
Barcelona, Spain
[2] MPI for Intelligent Systems
Tuebingen, Germany
[3] ETH Zurich
Switzerland

Abstract

Temporal clustering of human motion into semantically meaningful behaviors is a challenging task. While unsupervised methods do well to some extent, the obtained clusters often lack a semantic interpretation. In this paper, we propose to learn what makes a sequence of human poses different from others such that it should be annotated as an action. To this end, we formulate the problem as weakly supervised temporal clustering for an unknown number of clusters. Weak supervision is attained by learning a metric from the implicit semantic distances derived from already annotated databases. Such a metric contains some low-level semantic information that can be used to effectively segment a human motion sequence into distinct actions or behaviors. The main advantage of our approach is that metrics can be successfully used across datasets, making our method a compelling alternative to unsupervised methods. Experiments on publicly available mocap datasets show the effectiveness of our approach.

1 Introduction

The automated segmentation of a human motion sequence into plausible and semantically meaningful human behaviors is a central problem in computer vision and in computer graphics. Addressing this problem from the perspective of human poses obtained by motion capture systems is becoming more relevant due to the proliferation of motion capture databases and recent advances in markerless motion capture [1, 15]. Such an approach is not only interesting because of the potential availability of data, but also because human poses have potential for learning motion patterns that can be robustly employed across datasets and domains.

Segmenting human motion into distinct actions is a highly challenging problem. From the motion analysis perspective, segmentation is difficult due to large stylistic variations,

This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación, under project TEC2010-18094 and by EC project TANGO (FP7-ICT-249858)

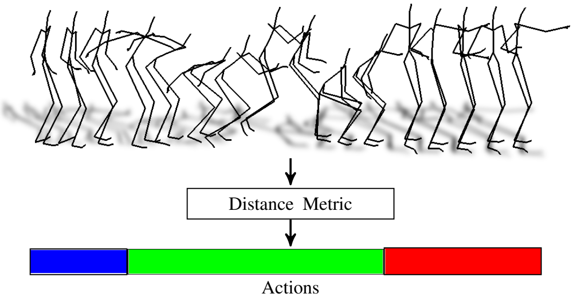

Figure 1: System Overview: Human motion sequences are clustered into different actions using a learned distance metric. We use annotations available in a mocap dataset to learn a distance metric that captures the semantic similarity between skeleton motion.

temporal scaling, changes in physical appearance, irregularity in the periodicity of human motions and the huge number of actions and their combinations. From a semantic viewpoint, segmentation is inherently elusive and difficult because in the vast majority of cases it is not clear when a set of poses describes an action. For instance, punching with the left hand and punching with right hand can be different actions, but it might be also regarded as punching or even more general as boxing.

In this paper, we propose to learn what makes a sequence of poses different from others such that it should be annotated as an action, as illustrated in Fig. 1. To this end, we make use of already annotated motion capture datasets and formulate action segmentation as a weakly supervised temporal clustering problem for an unknown number of clusters. Since publicly available datasets might contain different motions and action labels than the test sequences, we can not use the annotation directly for action segmentation. Instead, we use the annotations to learn a distance metric for skeleton motion using relative comparisons in the form of *samples of the same action are more similar than they are to a different action*. This is very intuitive since the sequences of a single database are usually labeled based on a semantic similarity. The learned distance metric is then used to cluster the test sequences. To this end, we employ a hierarchical Dirichlet process that also estimates the number of clusters.

The main advantage of our method is that it can be used for unseen actions and across datasets as we will show in our experiments.

2 Related Work

Metric learning from pose data has been mainly proposed in the computer graphics community in order to learn human-like perceptual similarities between poses [16]. The learned distance metric is then applied to the task of finding suitable transitions and content-based pose retrieval [4, 5, 19]. Metric learning has proven to be also useful in recognizing actions from video. Tran and Sorokin [18] extract silhouette and optical flow features from videos and they use them in conjunction with Large Margin Nearest Neighbors (LMNN) [21] to learn a metric that properly separates different action classes. More recently, Kliper-Gross et al. [10] have proposed a metric learning approach for one-shot learning of actions in videos.

In order to efficiently annotate actions in large collections of video or mocap data, some researchers have focused on unsupervised segmentation and clustering of human actions. Barbic et al. [2] propose a change detection algorithm for mocap data. They provide accurate results, but their method is not able to cluster the temporal segments into the different behaviors. Ozay et al. [12] overcome the clustering problem by modeling the first three principal components of the data as an autoregressive model. The coefficients of the model are then clustered with k-means. Similarly, the Aligned Cluster Analysis (ACA) proposed by Zhou et al. [22] extends the k-means concept to cluster time series. They show that ACA can accurately find different behaviors in sequences of mocap data. However, [12] [22] are limited by having to manually set the number of clusters (actions) k. In [13], this limitation is tackled by using a spike-train driven dynamical model that can detect motion transitions and clusters them into different behaviors, without having to manually set the number of clusters k. As far as video data is concerned, approaches such as [9][11] have proposed variants and extensions of hierarchical Dirichlet processes (HDP) [17] in order to find activities using optical flow features mainly. In [7], HDPs are used as a prior for HMM parameters in order to cluster time series data into distinct behaviors. This latter approach is applied to synthetic data, stock indices and dancing honeybee data.

3 Our approach

We aim at a temporal clustering of human actions in which one can provide some knowledge learned from data. The training data might be from a different database containing actions that are not relevant for the testing data. To meet these requirements, we learn a distance metric from pose-based features, and we use this metric to cluster pose feature vectors (Section 3.2) as illustrated in Fig. 2a. The outcome of such a clustering is then provided to a hierarchical Dirichlet process (HDP) in order to obtain the different activities of a motion capture sequence (Section 3.3). This strategy allows us to cluster motion sequences into different behaviors without knowing the exact number and types of actions in a test sequence.

3.1 Pose-based Features

The features employed in this paper are a rather simple yet efficient way of exploiting the pose information. We start by removing the orientation and translation of the input poses, in order to set them in a reference system that will allow an invariant comparison between action sequences. From these rotation and translation invariant poses, we obtain a set of 14 relevant joint positions $\{\mathbf{q}_1,\ldots,\mathbf{q}_{14}\}$ that can be easily obtained in different datasets [2]; see Fig. 2a. These joint positions are used to compute the following feature vector:

$$\mathbf{x} = \{\mathbf{q}_1,\ldots,\mathbf{q}_{14},\dot{\mathbf{q}}_1,\ldots,\dot{\mathbf{q}}_{14},\ddot{\mathbf{q}}_1,\ldots,\ddot{\mathbf{q}}_{14}\} \qquad (1)$$

where $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ denote joint velocity and acceleration respectively (derivatives are computed by time differences). In practice, we subsample mocap data (recorded at 120Hz) at 30 Hz.

3.2 Learning a metric for pose-based features

Given a set of feature vectors $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ in $\mathbb{R}^D$, we aim at learning a positive semi-definite matrix $\mathbf{A}$ such that the distance

$$d_A(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{A}(\mathbf{x}_i-\mathbf{x}_j) \qquad (2)$$

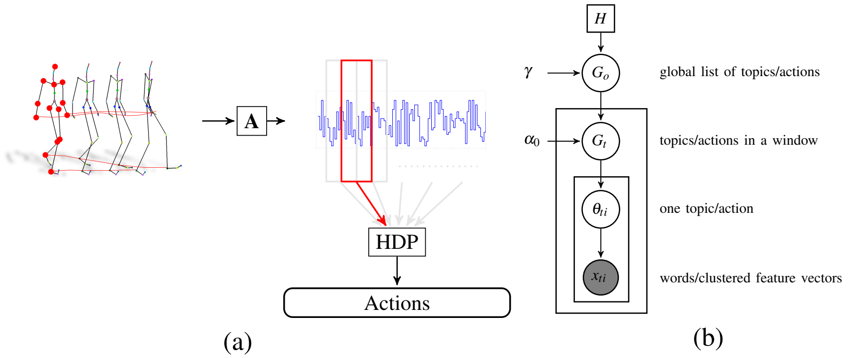
A
HDP
Actions
(a)
H
γ
G_o
global list of topics/actions
α_0
G_t
topics/actions in a window
θ_{ti}
one topic/action
x_{ti}
words/clustered feature vectors
(b)

Figure 2: (a) Detailed overview of our approach. A set of pose-based features are extracted using 14 relevant joints (marked with red spheres). These features are subsequently clustered into primitives using a metric (**A**) learned on related action sequences. In order to infer the different actions in a sequence, we first group the primitives using a sliding window. Then, we provide the resulting sets of primitives to a hierarchical Dirichlet process. (b) Detail of the hierarchical Dirichlet process.

satisfies a set of constraints defined in terms of relative comparisons of the form "$\mathbf{x}_i$ is closer to $\mathbf{x}_j$ than to $\mathbf{x}_k$". Using action labels, we can formulate these constraints in terms of similarity and dissimilarity between triplets of feature vectors. Under such constraints, we learn the matrix **A** by employing Information-Theoretic Metric Learning (ITML) [6]. ITML finds a suitable matrix **A** by formulating the problem in terms of how similar is **A** to a given distance parameterized by $\mathbf{A}_0$ (typically, the identity or the sample covariance). Provided that (2) is a Mahalanobis distance, one can treat the problem as the similarity of two Gaussian distributions parameterized by **A** and $\mathbf{A}_0$ respectively. That leads to an information theoretic objective in terms of the Kullback-Leibler divergence between both Gaussians. This divergence can be expressed as a LogDet divergence [6], thus yielding the following optimization problem:

$$\begin{aligned} \underset{\mathbf{A},\xi}{\text{minimize}} \quad & D_{ld}(\mathbf{A},\mathbf{A}_0) + \lambda D_{ld}(\text{diag}(\xi),\text{diag}(\mathbf{c})) \\ \text{s.t.} \quad & \delta_{(i,j)}(\xi_{(i,j)} - \text{tr}(\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T) \geq 0 \\ & \mathbf{A} \succeq 0, \xi \geq 0 \end{aligned} \tag{3}$$

where D_{ld} is the LogDet divergence, **c** is the vector of constraints, ξ is a vector of slack variables (initialized to **c** and constrained to be component-wise non-negative) that guarantees the existence of a solution and λ is a parameter controlling the tradeoff between satisfying the constraints and minimizing the similarity between distances.

In order to learn the metric (2) for the pose features (1), we have to define the constraints $d_A(\mathbf{x}_i,\mathbf{x}_j) \leq c_{(i,j)}$ or $d_A(\mathbf{x}_i,\mathbf{x}_j) \geq c_{(i,j)}$ for a pair of feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Since for each feature $\mathbf{x}_i$ we have only an action label y_i, we define the constraints based on triplets of points $(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)$ with class labels (y_i,y_j,y_k), where feature vectors with the same label should be closer to each other than to feature vectors with different labels. Using $\delta_{(i,j)} \in \{-1,1\}$ as similarity indicator (3), i.e., $d_A(\mathbf{x}_i,\mathbf{x}_j) \leq c_{(i,j)}$ if $\delta_{(i,j)} = 1$ and $d_A(\mathbf{x}_i,\mathbf{x}_j) \geq c_{(i,j)}$ otherwise, we formulate the following constraints:

$$\begin{array}{rll} y_i = y_j = y_k & \delta_{(i,j)} = 1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \leq \max(d(\mathbf{x}_i,\mathbf{x}_j), d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k)) \\ y_i = y_j \wedge y_j \neq y_k & \delta_{(i,j)} = 1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \leq \min(d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k)) \\ y_i \neq y_j \wedge y_i = y_k & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq d(\mathbf{x}_i,\mathbf{x}_k) \\ y_j \neq y_i \wedge y_j = y_k & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq d(\mathbf{x}_j,\mathbf{x}_k) \\ y_i \neq y_j \neq y_k & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq \min(d(\mathbf{x}_i,\mathbf{x}_j), d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k)) \end{array}$$

The values on the right hand side of the inequalities, $c_{(i,j)}$, are defined on the Euclidean distances $d(\,)$ between the features $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$. When the features have the same or completely different labels, the distance is constrained to be less or greater than the Euclidean distances, respectively. When only two features have the same label, the distance is constrained to be less than the Euclidean distances of the feature vector pairs with different labels.

For learning the metric, we randomly draw the triplets for generating the constraints from the training set. Furthermore, we estimate the tradeoff parameter λ by means of cross-validation, where our goal is to cluster pose-based features into a set of K primitives. To this end, we rely on a hierarchical clustering algorithm [20] to overcome the dependency on the initial point. We set a sufficiently high K (typically ranging from 16 to 64 clusters) and we find λ by minimizing the *purity* of the clusters obtained in cross-validation:

$$\mathcal{C}(\lambda) = 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\max_y(n_k^y) \tag{4}$$

where n_k is the number of feature vectors in the cluster k, and $\max_y(n_k^y)$ denotes the number of feature vectors of the class y appearing most frequently. Note that the dependence on λ comes from the fact that such parameter influences the resulting clusters.

Learning a metric from the proposed pose-based features can be seen as a *data-driven* transferring of implicit semantic distances derived from the class labels. In order to reduce the bias towards certain performance styles and to keep some temporal constraints, we investigate two additional variants of the pose-based metric learning framework.

Symmetry Unbiasing In order to reduce the bias towards action examples performed exclusively with right or left limbs, we *mirror* the poses. For instance, if we learn the metric with examples of *raising right hand*, we mirror the pose-based feature vectors in order to represent *raising left hand* and we assign the same action label (*raising hand*) to all these examples.

Temporal Alignment Two motion sequences of the same action class can be aligned by dynamic time warping [14]. Then, if under such alignment, a feature vector $\mathbf{x}_i$ from one sequence matches another feature vector $\mathbf{x}_j$ from the other sequence, we say that they are aligned. If two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to the same action class are aligned, they should be more similar than a third feature vector $\mathbf{x}_k$ of the same class that is not aligned with $\mathbf{x}_i$ and $\mathbf{x}_j$. Therefore, for any randomly drawn triplet $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ such that $y_i = y_j = y_k$, we define the following inequalities:

$$\begin{array}{rll}
i,j,k \ aligned & \delta_{(i,j)} = 1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \leq \max(d(\mathbf{x}_i,\mathbf{x}_j), d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k)) \\
i,j \ aligned & \delta_{(i,j)} = 1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \leq \min(d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k)) \\
i,k \ aligned & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq d(\mathbf{x}_i,\mathbf{x}_k) \\
j,k \ aligned & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq d(\mathbf{x}_j,\mathbf{x}_k) \\
i,j \ !aligned & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq \max(d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k)) \\
i,j,k \ !aligned & \delta_{(i,j)} = -1 & d_A(\mathbf{x}_i,\mathbf{x}_j) \geq \min(d(\mathbf{x}_i,\mathbf{x}_j), d(\mathbf{x}_i,\mathbf{x}_k), d(\mathbf{x}_j,\mathbf{x}_k))
\end{array}$$

where a pair of indices followed by *aligned* denotes a unique aligned pair within the triplet (!*aligned* expresses the contrary, the unique unaligned pair in the triplet) and three indices followed by *aligned* indicate that all the samples are aligned (a !*aligned* triplet means that any sample is aligned). These constraints replace the initial constraint for the case $y_i = y_j = y_k$, which was fulfilled from the beginning.

3.3 Discovering Actions

Given a sequence of pose-based feature vectors $\mathcal{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T\}$, we address the problem of inferring the performed actions as a temporal clustering problem in which we rely on weak supervision to learn semantic similarity in the form of a metric. Contrarily to other approaches [22], we want to address the temporal clustering problem for an unknown number of clusters or actions. For that matter, we rely on a hierarchical Dirichlet process (HDP) [17].

In our approach, two clustering levels are considered. The *low-level clustering* aims at quantizing the feature vectors into K primitives, such that discrete data can be provided to the HDP. The *low-level clustering* is performed by combining a hierarchical clustering algorithm (see Section 3.2) with the learned metric $\mathbf{A}$. In contrast, the *high-level clustering* is the temporal clustering of the different actions. Using a topic modeling metaphor (see Fig. 2b), *low-level clustering* is the step of computing *words* while the *high-level clustering* consists in finding the *topics* within a sequence. Actions are hence understood as co-occurring words in specific segments (G_t) of the sequence (G_0). The implications of such a model are two-fold. First, we assume that quantized feature vectors follow a multinomial probability distribution within each action and, consequently, temporal ordering is ignored. Second, the *low-level clustering* step is crucial, since producing better words will produce better clustering results. To compute temporal segments, we employ a sliding window of a given length and overlap. Using validation sets of mocap data, we found that a window of 7-15 frames and 1/2 of overlap worked well. Similarly, we found that values for concentration parameters in the range of 0.5 to 1.0 for γ and between 1 to 2 for α_0 (see Fig. 2b) provided good results. The base probability measure H (see Fig. 2b) is a symmetric Dirichlet distribution of parameter 0.5 [17].

4 Experimental results

We conduct several experiments on two publicly available mocap datasets to show the effectiveness of our method. The first dataset is the CMU mocap dataset [3]. This dataset contains a huge collection of motions performed by 144 subjects. Sequences include examples of one action as well as complex activities involving a combination of simple actions. One of the main drawbacks of this dataset is that the labeling of sequences is rather imprecise and the availability of action examples is biased towards locomotion mainly. The second dataset is the HDM05 dataset [8]. The HDM05 dataset contains more than three hours of motion capture data, involving more than 70 motion classes in 10 to 50 realizations executed by various actors.

In our experiments, we employ the following training sets:

CMU Sequences from several subjects containing examples of *walk*, *jump*, *run*, *boxing*, *drinking*, *lean forward to reach*, *bend* and *kicking a ball* actions. Examples of actions such as *boxing* and *jump* present a number of punching and jumping styles and variations.

HDM05 Sequences from the 4 available subjects containing examples of *walk*, *run*, *grab*, *kick*, *clap*, *jog*, *punch*, *hop* actions. These examples are taken from the cut sequences, and contain a huge variation of styles. For instance, action *clap* involves clapping in front of the torso and above head.

The testing sets are the following:

CMU Sequences 1 to 14 of subject 86 as in [22].

HDM05 We generate 10 long sequences by concatenating cut sequences not included in the training set.

4.1 Evaluation Metrics

Manually annotating different actions in a human motion sequence is a difficult task. Annotators have to precisely determine motion transitions and action labels. Without a specific guidance, the annotation variability for a dataset would make action labels useless. This also renders the evaluation a challenging task, since it is difficult to objectively determine the goodness of an approach given some labels with a potential annotation bias. We therefore employ several evaluation metrics to measure the accuracy of our approach.

Firstly, we use the same metric as [13], that does not penalize oversegmentation as far as the estimated labels consistently match different actions. Since in [13] the transitions are not evaluated, we use two versions of this evaluation metric. The first version (M1) evaluates all the frames, whereas the second version (M2) does not take into account the frames around ground truth transitions (we simply remove 0.2 seconds around the transition). The third evaluation metric (M3) is that of [22] applied to the case where the number of found clusters may differ from the ground truth. We compute the best label assignments for the number of clusters provided by the ground truth, hence under- and oversegmentation are strongly penalized. Finally, we provide the average error in the estimated number of clusters (Error k).

4.2 Experiments and Discussion

We learn different metrics employing the two training sets described in the previous section. Specifically, we learn metrics on the CMU and HDM05 training sets and we cross-test each of them on both the HDM05 and CMU test sets. Note that in learning a metric with CMU data, we use less labels than in the CMU test data (actions such as *stretching*, *basketball dribble* or *climbing a ladder* are not present in the training examples). Additionally, we investigate the impact of mirroring and alignment. In all the experiments, we employ a sliding window of 15 frames with 1/2 overlap and 21 primitives or words. We test with two

	M1			M2			M3			Error k		
A	CMU	HDM05	**I**	CMU	HDM05	**I**	CMU	HDM05	**I**	CMU	HDM05	**I**
Normal	82.7	88.1	78.4	84.3	89.9	79.8	61.6	70.2	66.1	3.3	2.5	1.4
Mirror	88.0	90.3	78.4	89.7	92.2	79.8	67.4	70.5	66.1	2.9	2.8	1.4
Mirror+Align	86.9	89.5	78.4	88.5	91.3	79.8	67.2	69.4	66.1	3	2.8	1.4

Table 1: Clustering results for the HDM05 concatenated sequences. For each one of the 4 proposed metrics, we show the results when learning a metric on the CMU and HDM05 datasets and when using the Euclidean distance (**I**). See Section 4.1 for the definition of the evaluation metrics.

	M1			M2			M3			Error k		
A	CMU	HDM05	**I**	CMU	HDM05	**I**	CMU	HDM05	**I**	CMU	HDM05	**I**
Normal	87.3	89.5	88.5	88.2	90.5	89.4	73.9	82.2	80.6	3.1	2.5	2.5
Mirror	88.8	90.9	88.5	90.0	91.9	89.4	77.0	80.4	80.6	2.8	3.1	2.5
Mirror+Align	89.5	90.5	88.5	90.5	91.5	89.4	78.2	81.2	80.6	3.1	3.1	2.5

Table 2: Clustering results for the CMU sequences (14 sequences of subject 86). For each one of the 4 proposed metrics, we show the results when learning a metric on the CMU and HDM05 datasets and when using the Euclidean distance (**I**). See Section 4.1 for the definition of the evaluation metrics.

sets of HDP concentration parameters, $\gamma = 0.7, \alpha_0 = 1$ and $\gamma = 1, \alpha_0 = 2$. We provide the average performance over these two sets of parameters. Results are shown in Tables 1 and 2.

The performance on the HDM05 cut sequences (Table 1) shows that using a metric to cluster the feature vectors boosts the performance of the HDP temporal clustering. Best performance is achieved when using a metric learned on the HDM05 dataset. Such an outcome was expected, since action labels are the same as in the test data. Interestingly, using a metric learned with CMU data outperforms the Euclidean distance on the HDM05 test sequences. In both cases, we observe that, although the rest of metrics show a superior performance, the error in the estimated number of clusters is higher when using a learned metric. However, the clusters provided by using the Euclidean distance also imply a higher number of mismatches between cluster labels and ground truth labels. When using the Euclidean clustering, actions such as *walk* and *jog* often get merged together into the same cluster. These errors cause the number of estimated clusters to be closer to the ground truth, but several of the obtained clusters are lacking semantic meaning, as rather different labels get merged. On the contrary, although oversegmenting some actions into different stylistic performances, using the learned metric generally provides semantically meaningful clustering of motion into different behaviors.

Method	Known k?	Accuracy	Notes
ACA [22]	Yes	92.1%	Computed using the software provided by [23]
SAR [12]	No	72.3%	As reported in [13]
STS [13]	No	90.9%	As reported in [13]
Our HDP-E	No	89.4%	
Our HDP-$\mathbf{A}_{\text{CMU}}$	No	90.5%	
Our HDP-$\mathbf{A}_{\text{HDM05}}$	No	91.9%	

Table 3: Comparison to state-of-the-art approaches on the CMU dataset. HDP-E stands for hierarchical Dirichlet process using Euclidean distance for feature-vector clustering while HDP-$\mathbf{A}_Z$ means that feature-vector clustering is performed with the metric learned with Z data. Note that methods are not directly comparable since they rely on different assumptions.

Results on the CMU sequences of subject 86 confirm that using a metric provides better temporal clustering results. Interestingly, the best performance is achieved by learning a metric on the HDM05 dataset (see Table 2). This result yields two conclusions. First, the learned metric provides a good performance across datasets. Second, the benefits of learning a metric for temporal clustering of actions not only depend on the extent to which the training data could potentially explain the test data, but also on the labeling precision of the training examples.

Clustering results for the CMU test sequences are provided in Fig. 3. Using a learned distance metric for clustering the pose-based feature vectors involves a more semantically meaningful clustering of motion into actions. This can be clearly observed in sequences 1 to 9 and 11, where a number of noisy transitions are clustered as distinct behaviors when using the Euclidean distance. The exception to this performance is found in sequences 12 and 13. In these sequences, the metric learned from the HDM05 dataset helps in clustering action walk (red label in sequence 12 and 13 of Fig. 3) from the rest of the actions, but the examples employed in learning the metric do not help in achieving a semantically correct clustering of classes such as sweeping and dragging, and hence transitions between such actions, or even phases of the same action, are clustered as different behaviors.

When comparing the performance using mirroring and alignment constraints in Tables 1 and 2, we see that mirroring the poses improves the performance. The alignment improves the results only for training and testing on CMU; otherwise the performance degrades. This indicates that the alignment is only beneficial when the training sequences are not precisely segmented and labeled as it is the case for the CMU sequences.

In Table 3 we provide a comparison between state-of-the-art approaches for temporal clustering of human actions. In this comparison, we report the results using metric M2, since is the most similar to that employed to evaluate [12] and [13]. Note that the results provided by [13] are computed on a subset of sequences (1-3 and 5-6) for subject 86, which are easier to segment than the other sequences (see Fig. 3). In spite of that, we report a better overall performance. We also show that our approach is a compelling alternative to ACA, since we can obtain accurate clustering results by resorting to action examples from other datasets instead of requiring the exact number of clusters.

5 Conclusions

We have presented an approach for temporal clustering of human behaviors. The method is based on learning a metric from pose-based features, such that the semantics of action labeling are learned in the form of a distance. Our experimental results have shown that the learned metrics improve the clustering results even across datasets and do not require that the actions of the test sequences are present in the training data. The benefit of the learned metric, however, depends on the similarity of the poses in the training and test set, but also on the labeling precision of the training examples. While this needs to be addressed in the future, the proposed approach, which exploits publicly available mocap datasets for temporal clustering, is a compelling alternative to unsupervised methods.

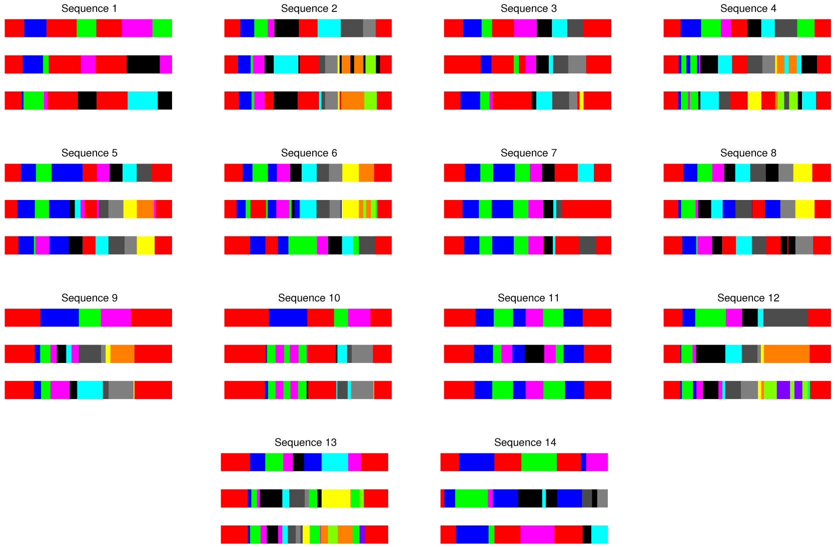

Figure 3: Temporal clustering of the 14 subject's 86 CMU sequences (best viewed in color). In each caption (top row) Ground Truth Labels (obtained from [23]), (mid row) temporal clustering with HDP and (bottom row) temporal clustering with HDP + metric learned with HDM05 data. The mocap sequences can be viewed at `http://mocap.cs.cmu.edu/search.php?subjectnumber=86`.

References

[1] G. Fanelli A. Yao, J. Gall and L. Van Gool. Does human action recognition benefit from pose estimation? In *Proceedings of the British Machine Vision Conference*, pages 67.1–67.11. BMVA Press, 2011.

[2] J. Barbič, A. Safonova, J. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004*, GI '04, pages 185–194, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

[3] Carnegie Mellon University Motion Capture Database. http://mocap.cs.cmu.edu. URL `http://mocap.cs.cmu.edu`.

[4] C. Chen, Y. Zhuang, J. Xiao, and Z. Liang. Perceptual 3d pose distance estimation by boosting relational geometric features. *Comput. Animat. Virtual Worlds*, 20(2âĂŘ3): 267–277, jun 2009.

[5] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao. Learning a 3d human pose distance metric from geometric pose descriptor. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1676 –1689, nov. 2011.

[6] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 209–216, 2007.

[7] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric Bayesian learning of switching linear dynamical systems. In *NIPS*. MIT Press, 2008.

[8] HDM05 Mocap Dataset. http://www.mpi-inf.mpg.de/resources/hdm05/index.html. URL http://www.mpi-inf.mpg.de/resources/HDM05/index.html.

[9] K.M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3241 –3248, june 2011.

[10] O. Kliper-Gross, T. Hassner, and L. Wolf. One shot similarity metric learning for action recognition. In *Proceedings of the First international conference on Similarity-based pattern recognition*, SIMBAD'11, pages 31–45, Berlin, Heidelberg, 2011. Springer-Verlag.

[11] D. Kuettel, M.D. Breitenstein, L. Van Gool, and V. Ferrari. What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1951 –1958, june 2010.

[12] N. Ozay, M. Sznaier, and O.I. Camps. Sequential sparsification for change detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –6, june 2008.

[13] M. Raptis, K. Wnuk, and S. Soatto. Spike train driven dynamical models for human actions. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2077 –2084, june 2010.

[14] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26 (1):43 – 49, feb 1978.

[15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, Colorado Springs, 2011. IEEE.

[16] J. K. T. Tang, H. Leung, T. Komura, and H. P. H. Shum. Emulating human perception of motion similarity. *Comput. Animat. Virtual Worlds*, 19(3-4):211–221, sep 2008.

[17] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[18] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 548–561, Berlin, Heidelberg, 2008. Springer-Verlag.

[19] J. Wang and B. Bodenheimer. An evaluation of a cost metric for selecting transitions between motion segments. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 232–238, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[20] Jr. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[21] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, June 2009.

[22] F. Zhou, F. De la Torre, and J. K. Hodgins. Aligned Cluster Analysis for Temporal Segmentation of Human Motion. In *IEEE Conference on Automatic Face and Gestures Recognition (FG)*, September 2008.

[23] Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *Accepted for publication at IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, 2012.