Deep Motifs and Motion Signatures

ANDREAS ARISTIDOU, The Interdisciplinary Center
DANIEL COHEN-OR, Tel-Aviv University
JESSICA K. HODGINS, Carnegie Mellon University
YIORGOS CHRYSANTHOU, University of Cyprus & RISE Research Center
ARIEL SHAMIR, The Interdisciplinary Center

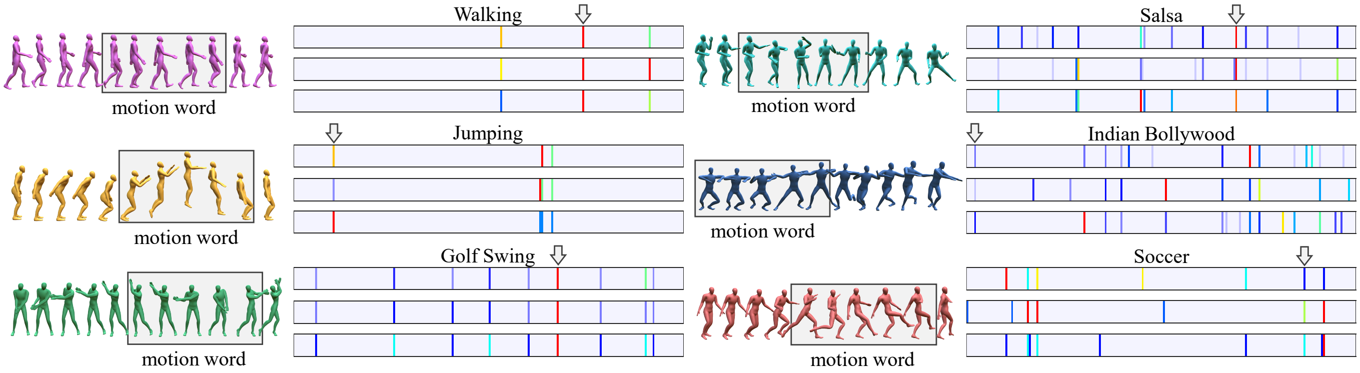

Fig. 1. Our motion signatures are defined using a deep analysis of motion words and selection of motion-motifs. Each signature is represented by a horizontal bar that shows the frequency of motion-motifs using color coding from red (high) through blue (low) to gray (zero). Note that the signatures represent distributions and not time evolution - the horizontal axis is not temporal. Three signatures of sequences are shown for each motion type – as can be seen, motions of similar type produce similar signatures where many motifs align. The rectangles in the sequence of motion to the left of the signatures illustrate motion words associated with the motifs shown by the corresponding arrow above the signature.

Many analysis tasks for human motion rely on high-level similarity between sequences of motions, that are not an exact matches in joint angles, timing, or ordering of actions. Even the same movements performed by the same person can vary in duration and speed. Similar motions are characterized by similar sets of actions that appear frequently. In this paper we introduce *motion motifs* and *motion signatures* that are a succinct but descriptive representation of motion sequences. We first break the motion sequences to short-term movements called motion words, and then cluster the words in a high-dimensional feature space to find motifs. Hence, motifs are words that are both common and descriptive, and their distribution represents the motion sequence. To cluster words and find motifs, the challenge is to define an effective feature space, where the distances among motion words are semantically meaningful, and where variations in speed and duration are handled. To this end, we use a deep neural network to embed the motion words into feature space using a triplet loss function. To define a signature, we choose a finite set of motion-motifs, creating a bag-of-motifs representation for the sequence. Motion signatures are agnostic to movement order, speed or duration variations, and can distinguish fine-grained differences between motions of the same class. We illustrate examples of characterizing motion sequences by motifs, and for the use of motion signatures in a number of applications.

CCS Concepts: • **Computing methodologies** → **Motion capture**; **Motion processing**;

Additional Key Words and Phrases: Animation, Motion Word, Motif, Motion Signature, Convolutional Network, Triplet Loss.

ACM Reference Format:
Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Deep Motifs and Motion Signatures. *ACM Trans. Graph.* 37, 06, Article 187 (November 2018), 13 pages. https://doi.org/10.1145/3272127.3275038

1 INTRODUCTION

The availabilithy of human motion data in big repositories is growing with the emergence of simpler motion capture devices [Mehta et al. 2017; Pavlakos et al. 2017]. Content-based techniques and searching methods become essential to facilitate the use of such data. However, motion data is not always annotated or parameterized, hindering the semantic analysis of motions, the search in motion datasets, and the comparison between motion data. Working directly with the motion sequences is challenging due to the high-dimensional, temporal, nature of the motion, their large variations

Authors' addresses: Andreas Aristidou, The Interdisciplinary Center, Kanfei Nesharim, Herzliya, Israel, 4610101, a.aristidou@ieee.org; Daniel Cohen-Or, Tel-Aviv University, Ramat Aviv, Tel-Aviv, Israel, 6997801, cohenor@gmail.com; Jessica K. Hodgins, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, USA, PA 15213, jkh@cs.cmu.edu; Yiorgos Chrysanthou, University of Cyprus & RISE Research Center, 75, Kallipoleos, Nicosia, Cyprus, 1678, yiorgos@cs.ucy.ac.cy; Ariel Shamir, The Interdisciplinary Center, Kanfei Nesharim, Herzliya, Israel, 4610101, arik@idc.ac.il.

0730-0301/2018/11-ART187 $15.00
https://doi.org/10.1145/3272127.3275038

in time and space, and their sheer size. Human motion is complex and heterogeneous, and motions of similar content may consist of analogous movements but in different ordering.

In this paper, we tackle these challenges by extracting *motion-motifs* and introducing *motion signatures*. Our motion-motifs are small movements that are common inside a motion sequence and reveal its characteristics. Our high-level signatures provide a global, semantically meaningful representation that is both succinct and descriptive, instead of describing temporal similarity of specific movements or poses inside a motion sequence. Motifs and signatures sufficiently capture both the structure and the features of human motion, providing efficient means for matching, clustering, segmenting, and indexing of motions, and supporting advanced applications such as motion synthesis.

The premise of this work is that motion sequences can be broken down to smaller movements, and can be characterized by the distribution of such movements. For example, a walking sequence can be easily broken down to steps, but even more complex motions such as modern dancing contain similar simple movements that can be seen as motifs. We represent simple movements using *motion words*, which are narrow temporal windows around a given time-frame in the motion sequence. Motion words consist of short sequences of joints transformations, and represent the local evolution of pose. By clustering the motion words in feature space, we distill the words extracted from a sequence to a set of motifs which are descriptive and frequent words.

Motion signatures are defined based on a bag-of-motifs. This creates a descriptive and succinct representation for motion sequences. Two motion sequences are considered similar if their signatures are similar, which means that they have a similar distribution of motion-motifs. Motion signatures are oblivious to the temporal order of the motion words, and only consider the distribution of the words. The signatures do not depend on the length of the motion sequence, revealing that two sequences belong to the same semantic group even if they significantly differ in length, and without requiring temporal alignment or exact matching. Furthermore, our signatures are sensitive to fine-grained differences, and are capable of differentiating motions within the same class (for example differentiating between the leader and follower in a salsa dance – see Figure 2).

One of the key challenges we face is defining an effective feature space for motion words, where the distances among words are semantically meaningful, and where variations in speed and duration are handled. Inspired by the recent advances in content-based analysis of text and images, we base our analysis on neural networks features [Schroff et al. 2015; Szegedy et al. 2015]. In contrast to dynamic time warping (DTW) methods [Baak et al. 2008; Forbes and Fiume 2005; Keogh et al. 2004; Müller and Röder 2006], which mainly rely on local numerical cost-measures, we learn the spatiotemporal invariance between motion words by training a deep neural network to embed motion words into a feature space. The embedding process places semantically-similar motion words close together in this feature space, and semantically-different words far apart. We consider motion words as semantically-similar if either they originate from the same temporal context, or they appear in different contexts but represent the same motion, with possible variation in speed or duration. The embedding neural network is trained using a triplet loss, where the positive examples are either motion words that appear temporarily close in the training data, or words that match using dynamic time warping. Then, we apply unsupervised analysis by clustering the motion words in the feature space to find "deep" motion-motifs.

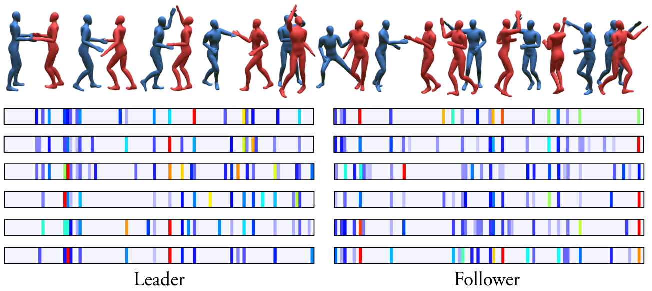

Fig. 2. Our method can distinguish fine-grained differences between similar motions. In this example, we separate the sequences of salsa dancers to leader sequences (illustrated by the blue dancer) and follower sequences (illustrated in red). Although they contain similar salsa movements, they have different distribution of motion words, and their signatures are distinguishable (bottom).

The definition of the motion words feature space is based on a massive training data-set of motions, and yields a universal feature space. However, the analysis and construction of motifs and motion signature can be tuned to capture certain classes, defined by more specific training dataset. Beyond the expected improvement in computational efficiency, we show that our deep learning approach learns semantic information related to the similarity of movements, efficiently extracts similar motifs from high dimensional motion data, and supplies the embedding directly, without the requirement of dimensionality reduction.

Our main contributions are (i) Defining a high-dimensional universal feature space for motion words, where Euclidean distances reflects semantic similarity among local movements represented by words, (ii) Defining motion-motifs as common and discriminative motion-features in animation, and (iii) Defining motion signatures: succinct and descriptive high-level representations of motion sequences, that reflect the distribution of motion-motifs found in the sequences.

We demonstrate that motion-motifs and signatures can be directly used in many applications for indexing, temporal segmentation, retrieval, and synthesis of motion clips. The effectiveness of our method has been evaluated on a wide range of motions, including short and long-length sequences of simple locomotion, or highly stylistic dancing.

2 RELATED WORK

Our work uses motion words as motion features and has several applications such as motion retrieval and clustering. Such applications are also built on extracting features from motions. Therefore, we describe previous approaches for extracting features from motion, as well as works that tackle specific applications we demonstrate. Lastly, we shortly review related works that use bag-of-features and neural networks similar to ours.

Motion Retrieval. Systems based on keyword queries [CMU 2018], or annotation [Arikan et al. 2003], are widely used for retrieval as they are cost-effective. However, they require manual labelling, are not robust to the semantics of motion, and cannot apprehend the complexities and particularities of motion data. Thus, motion retrieval is most often performed by matching similar poses [Beaudoin et al. 2008; Kovar et al. 2002; Lee et al. 2002], or other sketch-based methods [Chao et al. 2012; Choi et al. 2012; Thorne et al. 2004]. Such methods provide an intuitive means of query specification, but cannot capture the temporal evolution, and dynamics of human motion.

To integrate some aspects of the temporal evolution of motion, some works define pose-metrics that explicitly include dynamic features (e.g., joint velocities and accelerations) [Chai and Hodgins 2005], or use short-time windows around the frame of interest, that are temporally aligned using DTW [Rakthanmanon et al. 2012; Yi et al. 1998], or uniform scaling [Keogh et al. 2004]. However, nearest neighbors (NN) search and range queries do not scale well as they are time-consuming and computational expensive. Many efforts have been devoted in data mining to develop indexing schemes and optimizations to accelerate NN search and allow fast dynamic-time-warping computation on single [Yeh et al. 2018] or multi-dimensional time-series [Hu et al. 2013; Yeh et al. 2017]. To improve scalability and accelerate retrieval in animation, Kovar and Gleicher [2004] use match webs as an index structure to find numerically similar motions, Forbes and Fiume [2005] use principal component analysis (PCA) to compute compact representations of motions in lower dimensions, and Chai and Hodgins [2005] build a graph to allow fast nearest neighbor search. Others decompose motions into body parts and use a hierarchical motion representation (R-trees and kd-trees, respectively) for fast access [Deng et al. 2009; Keogh et al. 2004; Krüger et al. 2010; Liu et al. 2005; Tautges et al. 2011]. In contrast, our metric learning approach allows to expand beyond specific DTW rules by defining a high-dimensional universal feature space where Euclidean distances reflect semantic similarity among local movements represented by words.

Cost metrics that utilize other features include the relational features in Müller *et al.* [2005], features that exploits the geometric properties and relationships between different human body parts in Xiao *et al.* [2015], or the Laban Movement Analysis (LMA) features, which encode both the geometric and dynamic properties of motion in [Aristidou et al. 2015; Kapadia et al. 2013]. These methods are able to efficiently extract spatiotemporal information, but cannot describe close numerical similarity between poses, or global similarities between sequences. Other methods model motion data with a smaller set of features using an angular skeleton representation [Raptis et al. 2011], a temporal hierarchy of covariance descriptors [Hussein et al. 2013], or points in a Lie group [Vemulapalli et al. 2014]. However, these methods have difficulty handling heterogeneous and complex motions, similar motions with temporal variation in duration and speed, or motions that consist of non periodic actions (e.g., salsa dancers perform similar dance pirouettes, but at different times and in arbitrary order).

Müller and Röder [2006] represent motion using motion templates (Boolean values at selected keyframes), and compute their distance using a quantized-DTW approach. Sun *et al.* [2011] convert motion frames into a 2D representation, and then creates volumes to capture the dynamic of motion. These volumes are later decomposed to lower dimensional representations, and their similarity is computed using cross-correlation or DTW. Wang and Neff [2015] train deep autoencoders to extract a compact representation of motion with binary values, and compute their similarity using the Hamming distance. However, these methods can only handle short-time sequences, they cannot deal with complex and dynamic motion sequences, or motions with extreme reordering (e.g., dancing). Kapadia *et al.* [2013] describe motion using a number of LMA-derived keys, and then integrate an *m*-ary tree structure to provide mapping from those key sequences to motions. In contrast, we use *motion signatures* which are time-scale and temporal-order invariant, offering a succinct and descriptive representation of motion sequences. Note that, Vasilescu [2002] used the term motion signatures in a different manner; in that paper, motion signature captures the distinctive pattern of an individual's movement, and allows synthesis of new motions in that particular style.

Motion Clustering. Müller *et al.* [2009] propose to represent motion as an explicit matrix that captures the consistent and variable aspects of learnt motion classes. Unknown motion inputs are segmented and annotated by locally comparing them with the available motion templates. Bernard *et al.* [2013] developed MotionExplorer to cluster and display motions as a hierarchical tree structure. Their method combines a number of visualization techniques to support user overview and exploration. The authors apply the divisive hierarchical clustering algorithm to the low-level pose features. For neighboring construction, the authors used the self-organizing map (SOM) on joint position features of poses to train a topology preserving grid of poses. Similarly, Wu *et al.* [2009], and later Hu *et al.* [2010], cluster motion on hierarchically structured body segments, and measure the temporal similarity of each partition using SOM, which is computationally expensive. Chen *et al.* [2015] used hierarchical affinity propagation (HAP) to perform data abstraction on low level pose features to generate multiple layers of data aggregations. Recently, Bernard *et al.* [2017] present a visual-interactive approach for the semi-supervised labeling of human motion capture data; users assign labels to the data which can subsequently be used to represent the multivariate time series as sequences of motion classes.

Motion Segmentation. A common method for segmentation is to automatically detect changes in low-level kinematic features that are correlated to segment boundaries. However, simple kinematic features fail to semantically segment highly dynamic and complex movements such as dancing. Liu *et al.* [2003] proposed a key frame extraction method; first, a k-means algorithm is applied to segment the input motion clip into many short subsequences, and then, one frame of each subsequence is selected as the key frame. Then k-means is used to divide all frames into K clusters, and each frame is labeled by a cluster number. Then, the whole clip is segmented into short subsequences. Barbic *et al.* [2004] proposed the Mahalanobis distance between a Gaussian distribution of a specific time frame and the subsequent sample to detect a significant change. Zhou *et al.* [2013] solves the segmentation problem using hierarchical cluster analysis to find a partition of given multivariate time series into disjoint segments. Vögele *et al.* [2014] describe a method based on a neighborhood graph that finds primitive motion units. Field *et al.* [2015] employed Gaussian Mixture Models to represent human motion as a sequence of postures or motion primitives. Subsequences are identified through frequency analysis and compared via dynamic time warping in order to cluster similar sequences. More recently, Bouchard and Badler [2015] segment motions semantically by examining their LMA-inspired qualitative properties.

Bag-of-Words. The Bag-of-Words (BoW) model is a popular method for encoding text documents [Witten et al. 1994] and images [Csurka et al. 2004; Li and Perona 2005]; the main idea of BoW is to divide the subject into small feature descriptors, and use the distribution of the features as a signature. Kapsouras and Nikolaidis [2014] introduce a BoW framework for human action recognition in motion capture data. Similarly, Liu *et al.* [2017] achieved motion retrieval via temporal adjacent bag-of-words, while Takano *et al.* [2015] abstracts the dynamics of motion by symbolizing motion patterns through a Hidden Markov Model. However, they only use poses which cannot capture the semantic and temporal evolution of motion. In contrast, our method learns the feature space of short-time motions sequences using a deep network, that is semantically meaningful and time-scale invariant. Furthermore, in a similar manner to methods that learn discriminative features in image processing [Doersch et al. 2012; Singh et al. 2012], our method learns motion-motifs that are repeating and discriminative, and their frequency characterizes different motion classes. Similarly in data mining, [Li and Lin 2017; Lin et al. 2012] used a Bag-of-Patterns framework to compare non-synchronized, one dimensional, time series data.

Deep Learning Networks. Deep learning using neural networks became very popular for classification, regression or synthesis in text, image, and audio processing [Bengio et al. 2013; LeCun et al. 2015], and for behavioral recognition and annotation in video [Donahue et al. 2017]. Recently, neural networks have been used in animation for motion synthesis and character control [Holden et al. 2017; Liu and Hodgins 2017; Peng et al. 2017]. Indeed, one way to deal with motion classification and retrieval, in a similar manner as Holden *et al.* [2016], is to learn motion manifolds using deep learning. However, the availability of motion capture data is still limited in terms of amount and diversity, compared to images and text. This can create difficulties in training a deep neural network models. In addition, highly dynamic movements combine a number of different actions (e.g., a basketball player walks, runs, jumps, shoots, defends), and also consist of unimportant and redundant movements. It seems that a vast amount of motion data is essential to efficiently train a neural network and an enormous amount of time and manual effort are required to label the data. In contrast, we choose to work with motion words instead of whole motion sequences. Dividing motion sequences into shorter windows allows easier access to large amounts of data and is also oblivious to the length of the sequences. More importantly, our method does not require manual labeling at the word-level, as we exploit both the temporal coherency of motion and a similarity measure that is time-scale invariant (using dynamic time warping), to express the word semantics.

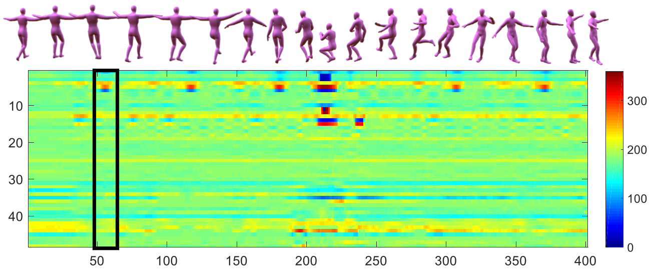

Fig. 3. A motion sequence is represented using a color coded matrix: the horizontal axis is time (frames), and each column displays the degrees of rotations for all joints in one frame using color coding. A motion word (illustrated by a black rectangle) is a narrow temporal-window of joint rotations.

3 UNIVERSAL MOTION WORDS FEATURE SPACE

The premise of this work is that motion sequences can be broken down to smaller movements and represented by the distribution of such movements. Towards this end, we extract motion words from a large set of various motion sequences, and then map them into a universal feature-space $\mathbb{R}^d$, where distances represent semantic similarity.

The challenge is to define an effective feature space $\mathbb{R}^d$, where similar motion words will be positioned closely, and non-similar ones far apart. Defining a similarity measure between motion words is challenging because motion words are temporal entities, their variability is extremely high (as opposed e.g., to text words), and similar motions can vary in speed and duration. Moreover, we want a simple and fast embedding of words so that the signature calculations that depend on the embedded space are cost effective.

Our key idea is to use a neural network to map motion words into a latent feature space $\mathbb{R}^d$. Such a mapping is both simple and effective. We train a network to learn the mapping based on semantic similarity of motion words, and then use the mapping to calculate similarity between words. In the following, we first describe what motion words are, define the semantic similarity between words, and then present our mapping algorithm using a neural network.

3.1 Motion Words Definition

We use joint rotation angles to represent a motion sequence instead of joint positions. This representation allows finding similarity of motions regardless of the global position, and supports descriptors that are invariant to local translation and local orientation. Each joint defines three rotation values that are in the range of [0; 360] degrees. A *motion word*, is a narrow temporal-window of all joint rotations around a given frame [Aristidou et al. 2018]. Motion words divide a motion sequence into smaller, overlapping, feature descriptors (see Figure 3), defining a local spatiotemporal descriptor.

To define our universal motion words feature-space, we gather motion words from a large dataset $\mathcal{D}$ of motion-capture data of various activities including: walking, jumping, dancing, sports, and more. All words gathered from these sequences form the vocabulary of words we use to define the embedding feature space.

3.2 Semantic Similarity of Words

In contrast to text words that have a fairly small and well defined dictionary, motion words have no dictionary. Furthermore, the possible degrees of freedom for human articulation motion is extremely high. Hence, the definition of similarity between motion words is a challenge.

We use two characteristics to define semantic similarity of motion words. First, we assume that motion words that are temporally close are also semantically close, as they represent motions of similar content. Second, we use motion words from the same sequence that are far apart, but contain similar content in terms of the sequence of poses. There are various cost functions for measuring pose-based similarities; a discussion and evaluation on cost metrics for matching motion segments can be found in Wang and Bodenheimer [2003]. However, similar human motions may vary in duration and speed. Thus, we cannot simply compare fixed time-window motion words. We employ a Dynamic Time Warping (DTW) measurement similar to [Aristidou et al. 2018] (with no optimizations), so as to align two motion words of different durations as well as ones that have non-uniform variations in time.

Learning an embedding space for motion words where similarity can be measured, instead of directly computing their distance, reduces the required computational time, and also provides a semantic embedding. For instance, we show in Section 5, that our deep feature space can save up to 80% of the time required compared to DTW calculations, while presering the semantics.

3.3 Triplet loss Neural Network

Deep convolutional neural networks have been used successfully to learn semantic representations of data such as images, videos and text. Siamese Neural Networks are a class of network architectures that contain two identical sub-networks. They can learn a mapping from input space to an embedding space where distances represent similarity between inputs [Chopra et al. 2005; Zagoruyko and Komodakis 2015]. These networks are trained by minimizing a contrastive-loss function where the distance between the embedding of two input vectors is small for similar inputs and large for dissimilar ones.

The contrastive-loss function is restrictive as it forces all positives vectors to be close, while the negatives are separated by a certain fixed distance. Hence, we use an extension of Siamese networks to a triplet loss objective function [Schroff et al. 2015] that only requires negative vectors to be farther away than any positive vector on a per-example basis [Wu et al. 2017]. In our experiments we found the triplet loss to be more accurate. Specifically, our triplet loss network (see Figure 4) strives for an embedding function $f(x)$ that minimizes the distance between an *anchor* motion word x^a and a *positive* x^p sample, that is semantically close to x^a, and maximizes the distance between x^a and a *negative* x^n sample that is semantically different. The loss for a single triplet is defined as:

$$L(x^a, x^p, x^n) = [\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + \alpha]_+, \quad (1)$$

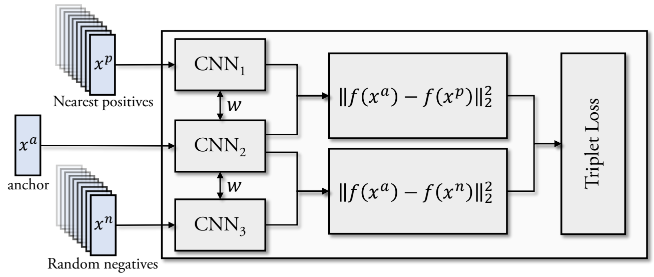

Fig. 4. Our triplet loss network.

where α is a margin that is enforced between positive and negative pairs. The triplet loss is a sum over all anchor-positive-negative triplets in the dataset $\mathcal{D}$:

$$L = \sum_{(x^a, x^p, x^n) \in \mathcal{D}} L(x^a, x^p, x^n). \quad (2)$$

Our network creates a 736 × 1 embedding by integrating the Inception model [Szegedy et al. 2015], using a similar architecture as [Schroff et al. 2015]. The actual implementation of the network (in Torch) is given in our supplemental material.

3.4 Network Training

The training-set motion words are taken from our motion dataset $\mathcal{D}$ that are not labeled. Hence, selecting appropriate triplets x^a, x^p, x^n for training the network is crucial for learning the correct embedding and for fast convergence.

In general, good training triplets are those closer to the separation margin between positive and negative examples. Therefore, given x^a, we should select difficult positive samples x^p by finding $\arg\max_{x^p} \|f(x^a) - f(x^p)\|_2^2$, and difficult negative samples by finding $\arg\min_{x^n} \|f(x^a) - f(x^n)\|_2^2$. However, given that our vocabulary of motion words is large, it is infeasible to compute the optimal arg max and arg min for the positive and negative triplets. Moreover, many times it is impossible to find the optimum since the motion words are *not labeled*. Instead, in an *unsupervised* manner, we use two types of positive examples, and random words for negative examples. The first type of positive pairs are words from a similar context as the anchor word x^a, and the second are words that are similar in terms of motion.

For the first type, we take the z nearest motion words ($z = 4$) that are temporally closest to x^a with no overlap. For the second type, we take the k nearest neighbors ($k = 5$) of the anchor motion word x^a by employing a time-warped extension of the Lee *et al.* [2002] distance metric, that uses a weighted sum of the difference in rotation between joints. The time-warped distance between motion words is defined as:

$$dist_{ij}^2 = \sum_{m=1}^{l} \| \log\left(q_{j,m}^{-1} q_{i,m}\right) \|^2, \quad (3)$$

where m is the number of joints in the motion word, and $q_{i,m}, q_{j,m} \in \mathbb{S}^3$ are the complex forms of the quaternion for the m-th joint in the i and j frames, respectively. The log-norm term $\| \log\left(q_{j,m}^{-1} q_{i,m}\right) \|^2$ represents the geodesic norm in quaternion space, which yields the distance from $q_{i,m}$ to $q_{j,m}$ on $\mathbb{S}^3$. The final distance between the two motion words is defined as the average distance of the matched frames in the optimal matching sequence found by DTW. We compared different metrics for measuring the distance between motion words (e.g., the Euclidean distances between joints [Kovar et al. 2002]), and observed similar results in terms of the efficiency and the required computational time.

The negative samples are randomly sampled motion words that are temporally far away from x^a in the sequence or taken from another motion sequence. Together, for each anchor motion word we build $k + z$ triplets using positive and negative samples, taken from both the motion sequence of the anchor and other motions in the dataset $\mathcal{D}$. Note that, some nearby pairs can be outliers, but these do not harm the convergence of the metric learning. In contrast, learning an embedding based on a large number of observations that may contain noise is in fact the strength of our neural network based method. Lastly, to reduce sensitivity in learning, we augment the training set by algorithmically time-warping motion words found in $\mathcal{D}$, and using the augmented words as additional positive examples. This produces triplets that are closer to the bounds while considering contextual similarities.

Although the training of a neural network is time-consuming (see Section 5), once the training is done, the embedding of new motion words is extremely fast. Using a single feed forward, the network produces the vector representation of the word in $\mathbb{R}^d$ feature space.

4 MOTION SIGNATURES

Bag-of-Words (BoW) characterizes the object being represented using the distribution of individual features. We use this approach for classifying and indexing motions. However, we do not use all motion words, but concentrate on the most common and discriminative words that we term *motifs*. The *motion signature* of a motion sequence is defined by a Bag-of-Motifs, which models the distribution of these motion-motifs. We expect that motion sequences of the same type will have signatures with similar characteristics.

4.1 Motion Word Motifs

Because of the variability of motion words it is impossible to define a finite universal dictionary for motion words. Instead, given a set of motion sequences (e.g., a set $\mathcal{D}_1$ of sport activities, or a set $\mathcal{D}_2$ of dances), we map all motions words in the set into the d-dimensional universal feature space $\mathbb{R}^d$, and cluster the words in this space. We use K-means clustering algorithm and group the motion words into K mutually exclusive clusters. K-means creates compact and separated clusters.

Each cluster i, is represented by a *motif* motion word ($\bar{x}_i$) which is the centroid of the cluster. The importance of a motif is defined by the density of its cluster; strong motifs are densely distributed, and thus, have larger importance. In contrast to previous works that find motifs in multi-dimensional time series, [Yeh et al. 2017] we assign all motion words to some motif (or cluster). Given any motion-word, we can associate it to a cluster simply by finding the closest motif using L2 distance in $\mathbb{R}^d$. Figure 5 shows examples of motion-motifs from the dance database $\mathcal{D}_2$.

4.2 Motion Signatures

The *signature* of a motion sequence S is defined as the normalized histogram of its words in all K clusters. In other words, given a motion sequence S, we first extract all its motion words, map them to the universal feature space, assign each word to its representative motif, count the number of words in each of the K clusters, and divide by the total number of words in S. This creates a comparable signature for every motion sequence regardless of its length.

A common problem with naive frequency counting is that highly frequent motion words dominate the motion dataset, but may not be as informative or descriptive as some less common motion words (similar to stop-words in text vocabulary). For instance, a walking-pose sequence can be found in walking, but also in football, basketball, fighting sequence and more.

To alleviate this problem, we rescale the frequency of the motion-motifs by the frequency of the motifs in the corpus. We re-weight motion signatures using tf-idf (term frequency – inverse document frequency [Croft et al. 2009]). The importance of a motif is proportional to its frequency in the clip and inversely proportional to its frequency in the corpus.

4.3 Distance Between Signatures

Once we have the signatures of motion-sequences, we define the similarity between the sequences as the distance between the signatures. We use the Earth Mover's Distance (EMD) [Rubner et al. 2000] to compare signatures as they represent distributions. Assume the motion signatures P and Q that both have m clusters with:

$$P = \{(p_1, w_{p_1}), (p_2, w_{p_2}), ..., (p_m, w_{p_m})\}, \text{ and} \tag{4}$$

$$Q = \{(q_1, w_{q_1}), (q_2, w_{q_2}), ..., (q_m, w_{q_m})\} \tag{5}$$

where p_i is the motifs quantity (after tf-idf re-weighting) in the i-th cluster of the P signature, and w_{p_i} is the corresponding cluster's dense weight. Let $\mathbf{D} = [d_{ij}]$ be the ground distance between cluster p_i and q_j. EMD wants to find a flow $\mathbf{F} = [f_{ij}]$, with f_{ij} the flow between p_i and q_j, that minimizes the overall cost, subjects to the following constraints:

$$f_{i,j} \geq 0, \quad 1 \leq i \leq m, \quad 1 \leq j \leq m$$

$$\sum_{j=1}^{m} f_{i,j} \leq w_{p_i}, \quad 1 \leq i \leq m$$

$$\sum_{i=1}^{m} f_{i,j} \leq w_{q_j}, \quad 1 \leq j \leq m$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} f_{i,j} = \min \left\{ \sum_{i=1}^{m} w_{p_i}, \quad \sum_{j=1}^{m} w_{q_j} \right\}$$

The optimal flow F is found by solving this linear optimization problem. The earth mover's distance is defined as the work normalized by the total flow:

$$EMD(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} f_{i,j} d_{i,j}}{\sum_{i=1}^{m} \sum_{j=1}^{m} f_{i,j}} \tag{6}$$

Signatures of motion sequences can be used in learning for both labeled and unlabeled data.

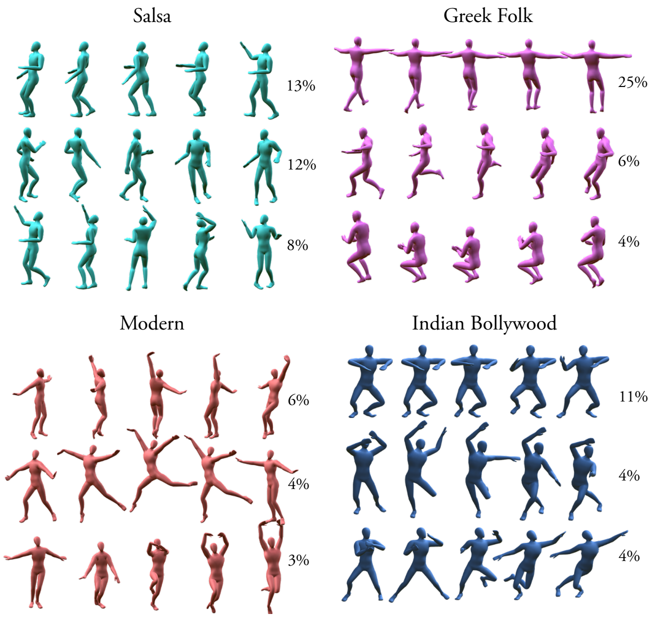

Fig. 5. Motion-motifs that are repeating and discriminative for different motion classes (from dataset $\mathcal{D}_2$). For instance, in salsa the partner right-turning is a unique and repetitive motion-motif, arabesque (jumping and stretching in the air) is for modern dancing, hitting a foot with a hand is for Greek folk dancing, and stretching left and right with bent knees is for Indian Bollywood. The percentage shown on the right of each motion-motif indicates the frequency of appearance of that motif cluster in the motion.

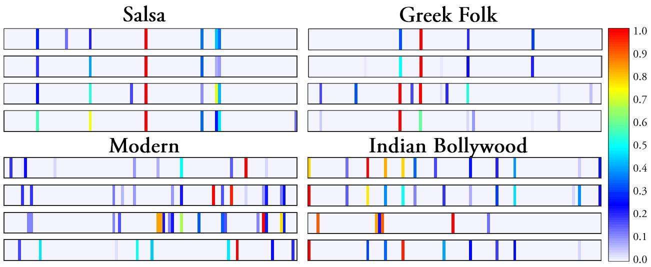

Fig. 6. Motion Signatures for the $\mathcal{D}_2$ dataset. In this example, we show four signatures of sequences from each dance motion class. The frequency of motion-motifs (the horizontal axis is all the motifs in the database) are illustrated by the colors (hot for high frequency, and cold for low frequency). Highly dynamic dances, such as modern dancing, have larger distribution of motifs compared to other dances such as Greek folk dancing, that are more structured with a lot of repetitions.

Unlabeled Data. To cluster unlabeled motions clips of similar contents into several clusters, we first measure the similarity between all the pairs of their signatures and then use Multi-Dimensional Scaling (MDS) to map the motion clips into an n-dimensional space. Thereafter, we can use clustering such as K-means to group similar motions into mutually common classes.

Labeled Data. For labeled motion sequence data, we can train a classifier based on the signatures of the labeled data. Then, given a new motion sequence of unknown class, we first create its signature and then use the signature for classification. For example, we can use the K-Nearest Neighbor (KNN) classifier, where the distance between the signatures is measured using EMD.

5 EVALUATION & COMPARISON

In this section, we provide several experiments to evaluate the performance of our content-based motion analysis in terms of efficiency and classification accuracy, and compare it with alternative methods. We evaluate our method on various data, including generic data (e.g. actions such as walking, running, jumping, dancing, playing sports) and more specific dynamic motions (e.g., different forms of dancing).

5.1 Implementation Details

For our experiments we use the $m = 16$ most informative joints with their relative joint angles. We used data taken from the Carnegie Mellon University motion capture database [CMU 2018], the Dance Motion Capture Database of the University of Cyprus [DMCD 2018], and motion data acquired in our laboratory. The motion capture data were originally sampled between 120 to 480 frames per second, but since human motion is locally linear, we reduce it to 24 frames per second without much loss of temporal information (see Forbes and Fiume [2005]). We tested different motion word sizes (8, 16, 24, and 48 frames) and found that using 16 frames, that reflects 0.66 seconds, with a skip of 4 frames, to reduce the computational time, is the most efficient. This length is long enough to cover simple movements, but short enough to promote similarities. The definition of motion signatures is also dependent on the number of the clusters used. We tried different numbers of clusters on databases with different size and complexity, and empirically conclude that $K = 100$ is a sufficient number for datasets which consist of many different types of motions. Note that we did not observe large effects in motion classification when slightly different numbers of clusters were used. We have implemented our system in Matlab R2017b. All experiments were run on a six-core PC with Intel i7-6850K at 3.6GHz, 32GB RAM, and with nVIDIA Titan XP GPU.

We have prepared 1 million motion words from many different types of motions for training the deep learning network (equal number of positive and negative examples). The training of the triplet network took approximately seven days but this is performed just once to learn the network weights. Approximately 85% of computation time was dedicated to compute the motion words' pairwise distances using DTW, for the selection of the positive and negative samples. Note that, using an optimized implementation for faster DTW calculations can allow faster learning for our deep network.

We created three different datasets for training and testing the ability of signatures to organize and classify motion sequences: $\mathcal{D}_1$ consists of 150 motion clips from 15 different motion classes: walking, running, sitting, jumping, weight-carrying, climbing, swinging, placing ball, placing tee, kicking, soccer and basketball playing, boxing, swimming, salsa and Indian Bollywood dancing; $\mathcal{D}_2$, consists of 30 motion clips of five types of dances: waltz, Latin (salsa, bachata, reggaeton), modern, Indian Bollywood, and Greek folk, with a large deviations in duration, ranging from 20 to 200 seconds; $\mathcal{D}_3$, which

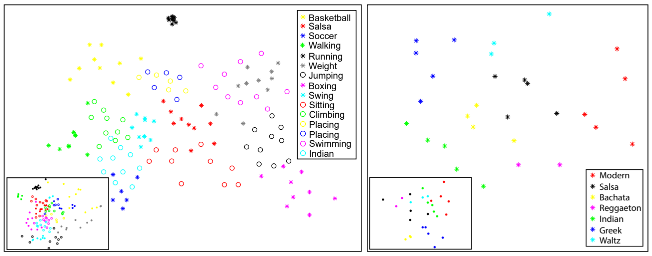

Fig. 7. Clustering of motion types using signatures: we show a 2D embedding (using MDS) of motion clips using the signatures for datasets $\mathcal{D}_1$ (left), $\mathcal{D}_2$ (right), the smaller windows show the 2D embedding of the corresponding motion clips when only DTW is used.

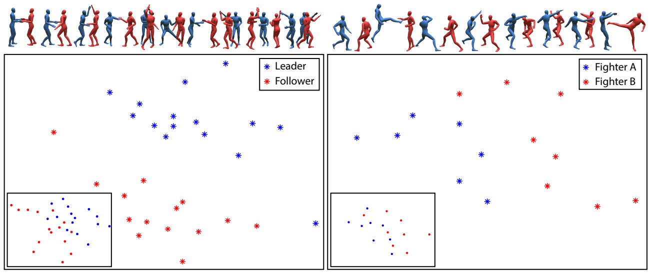

Fig. 8. Distinguishing fine grained motion differences: we show the 2D embedding of signatures of two salsa dancers (left) and two kung-fu fighters (right). Using signatures we are capable of distinguishing the movements of the leader and follower in Salsa and the differences in the technique used by each fighter in kung-fu. Note, for example, how a simple linear classifier can be used to separate the two movements using our signatures, but cannot be used when DTW distances are used for embedding (smaller windows).

has been selected to evaluate whether motion signatures can capture the fine details of motion, consists of 30 salsa dance sequences, half of them presenting the leader, and half the follower (CMU subjects 60 and 61).

Using the distance feature space, our system requires about six hours to train a classifier for $\mathcal{D}_1$. This includes computing the motion words' embedding for all motion clips in the dataset, defining the $k = 200$ motifs by clustering, and calculating the distribution of their motion words to define the motion signatures. For example, it takes 15 seconds (again, with no optimizations) to assign the motion words to motifs for an input motion clip of approximately 2000 frames (83 seconds), while the total time for classifying an unknown motion clip into a motion class (action) is approximately 20 seconds.

5.2 Motifs and Signatures

Figure 5 shows a number of motion-motifs for different motions sequences from the $\mathcal{D}_2$ dataset. We illustrate motifs that occur often in a specific motion class and not in others. It is important to note that not all types of motion have both repetitive and discriminative motifs; for instance, salsa and waltz have similar motifs but they appear at different frequencies.

Figure 6 shows four motion signatures for each type of dance from the $\mathcal{D}_2$ dataset. The signatures of related motions have similar distribution of motion-motifs (see also Figure 1). The ability of our method to classify gross categories at large granularity is demonstrated in Figure 7, showing the 2D embedding of motion clips for all datasets using MDS. As can be seen, distances using our deep learning feature space separates the different types of motion-clips better than using DTW.

Table 1. Motion clustering performance of our method for the datasets $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ compared to the Müller and Röder [2006], Sun *et al.* [2011], and Kapadia *et al.* [2013] methods.

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$
Ours	90.7%	90.0%	96.7%
Müller and Röder 2006	82.9%	69.4%	64.1%
Sun *et al.* 2011	85.3%	65.5%	59.3%
Kapadia *et al.* 2013	86.7%	77.3%	75.5%

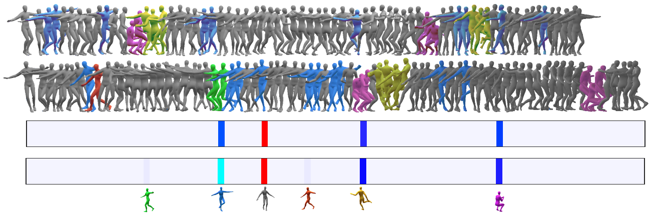

Fig. 9. Two sequences of Greek folk dancing with different order of movements and different length. The motifs are visualized using colors. As the two sequences have similar distribution of motion motifs, their motion signatures are similar.

Our method is also capable of clustering fine-grained differences in motions of the same class, as shown in Figure 8 (see also Figure 2). We demonstrate that motion signatures can differentiate between movements of two different actors performing the same type of motion: either the leader and follower in Salsa (the $\mathcal{D}_3$ dataset), or two different fighters in kung-fu (for kung-fu we used are 16 sequences, eight from each fighter).

5.3 Organizing Motion Collections

To evaluate our signatures for classification of motion sequences we have used the One-vs-All strategy [Bishop 2006]. A single classifier is trained for each class of motion with the sequences of that class as positive samples and all other sequence as negatives. We use leave-one-out cross validation to test the accuracy of the classification (approximately 80% of the data was used for training, and 20% for testing). Experiments using all datasets ($\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$) demonstrate that our method classifies motions into classes that share common characteristics with accuracy of 91.4%, while the corresponding accuracy when using DTW as the distance metric is 81.0%. Moreover, the time needed for our method to create the motion word embedding for the $\mathcal{D}_1$ dataset is 6 hours, and for $\mathcal{D}_2$ and $\mathcal{D}_3$, less than 3 hours. In contrast, the time of calculations using DTW to compare the words and create signatures is approximately 30 hours for $\mathcal{D}_1$, and 16 hours for $\mathcal{D}_2$ and $\mathcal{D}_3$.

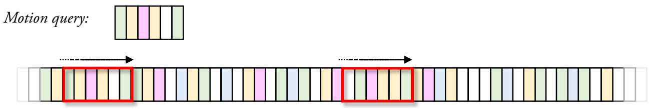

Fig. 10. Our motion retrieval method: we compare the signature of the query sequence of size $k = 6$ motion words (top) to a sliding window in any target sequence of motion. Each colored rectangle represents a motion motif (cluster). In this case two matches are found (red frames).

We also compared our method with other well-known motion indexing and classification approaches. Table 1 lists the classification accuracy of our method compared to the methods by Müller and Röder [2006], Sun *et al.* [2011], and Kapadia *et al.* [2013], for each dataset separately. Note that $\mathcal{D}_2$ and $\mathcal{D}_3$ are fine-grained datasets consisting of long sequences of complex, highly dynamic movements that are difficult to index or classify using other methods. This is because DTW or cross-correlation approaches cannot handle temporal variations and extreme reordering of motion motifs, a common phenomenon for long and heterogeneous movements. In contrast, as demonstrated in Figure 9, our method is invariant to the motif order and the sequence length.

Our deep Bag-of-Motifs framework also differs qualitatively with other neural network methods that may learn the distance function directly e.g., [Holden et al. 2015]. Learning a distance function for latent variable representations of motion requires a vast amount of labelled data. Such motion data availability is still limited in terms of amount and diversity. In contrast, our use of motion words allows easier access to large amounts of data, since manual labelling at the word-level is not required, and we can collect data in an unsupervised manner. Another critical difference is that our method is oblivious to the length and temporal order of the motion sequences, that makes it suitable for highly diverse motions such as dances. Learning a deep motion representation directly is applicable only for short time sequences, and single actions e.g., see [Wang and Neff 2015].

6 APPLICATIONS

In this section, we demonstrate several application examples that utilize motion signatures: retrieval of similar motions using a query sequence, segmenting motion sequence to primitive motion types, retrieval of sequences in a database, and using signature to define better context for motion synthesis.

6.1 Query-by-Example Retrieval

Given a short stream of frames, taken from any motion sequence as a query, our method can find similar motions inside sequences in the database using the universal feature space. The query motion can be of any size but it must contain more than one motion word to build a signature. Assume the query is of size k words. The key idea is to scan each sequence in the database using a sliding window of size k words, build a signature for the window sub-sequence, and compare it to the query's signature. Note that we do not compare the sequence of words in order but rather the signature (distribution of words) in each window. Figure 10 illustrates the motion retrieval method.

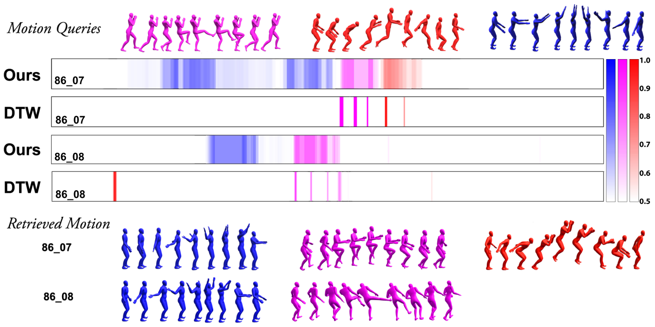

Fig. 11. An example of motion retrieval. At the top we show the three motion sequence queries, indicated by different colors (magenta: kicking, red: jumping, blue: stretching), and at the bottom, the corresponding similarity of the queries to two sequences (subjects: 86_07 and 86_08). The bars color indicates the corresponding motion query and the strength indicates the level of similarity (darker is more similar). Our method highlights the areas where motion is contextually similar to the query, while DTW only detects the time where the two motion sequences are synchronized (shown as Dirac peaks).

Using the EMD-distance between the signatures, gives an order independent measure. This way we can retrieve motions with similar content which are not temporally or spatially exactly the same. For example, motions that are reversed in time or in space. Figure 11 shows some examples. We use three motion queries and search for these motions in two sequences (subjects: 86_07 and 86_08). In this example, the size of the motion query used is $k = 5$ motion words, corresponding to a motion sequence of 32 frames. We compare our results with a time-warped version of the Lee *et al.* [2002] method (DTW) with a query motion sequence of 32 frames.

DTW reveals similarities only at the times when the two sequences are synchronized, while our method highlights the temporal areas where a motion with similar context as the query exists. Moreover, the stretching example in blue shows how we can match reversed motions (the arms move clockwise in the query sequence, and counterclockwise in the searched sequence) while our DTW implementation cannot. This ability is a result of using temporally close reverse motions as positive samples in the training. In case that such a property is not desired, users can train the network without these examples.

Working on a database of 15 motions (CMU subject_86, with total length 16.2 minutes), the preprocessing step of creating the motion words embedding and finding motifs takes one hour and fifty minutes, compared to tens of hours with DTW method. After the feature space is defined, it only takes a few seconds to assign the motion words of the query into the corresponding cluster, to define motion signatures, and hence retrieve contextually similar motions from the sequences.

6.2 Temporal Segmentation

Temporal segmentation of human motion into distinct motion primitives is crucial for synthesizing, classifying, and understanding human actions. Using motion signatures, we can efficiently classify motion sequences' parts into pre-trained classes of actions, and

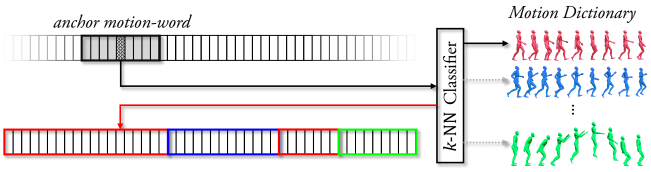

Fig. 12. Temporal segmentation method: each central motion word is classified according to the signature of a window around it.

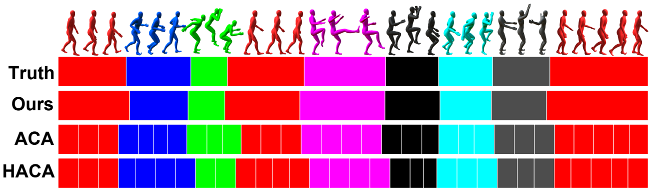

Fig. 13. Comparison of different temporal segmentation methods on motion capture data (subject: 86_03, from the CMU motion capture database). Different colors correspond to distinct actions. The first row illustrates the ground truth classes, the second row shows the segmentation achieved using our method, the third and forth rows present the results of the ACA and HACA algorithms, respectively [Zhou et al. 2013].

segment the sequence to shorter coherent motion clip parts. In our example, we train a dictionary of signatures for 10 different classes of motion using sequences from the CMU motion capture database containing single actions including: walking, running, long-jumping, kicking, punching, stretching, squats, sitting, jumping on left leg, jumping on right leg. Then, given an input motion clip, we use a sliding window with a size of $n = 9$ motion words (corresponding to 2 seconds of movement) and extract the signature of the window segment. We assign the motion word in the middle of the window to the motion class (action) that matches the window's signature using a KNN classifier (with $K = 5$). Figure 12 illustrates this procedure.

To evaluate the performance of our method, we use 10 motion sequences (taken from CMU database: subject 86), each of which is a combination of the 10 actions that our method has been trained with. Figure 13 shows an example result. The motion streams are (temporally) segmented, based on signatures, into parts that correspond to seven different actions from the 10 classes. The sequence contains 1680 frames (70 seconds), resulting in 420 motion words. The performance of our method is visually compared against the ACA and HACA algorithms [Zhou et al. 2013], and the ground truth. The overall segmentation accuracy of our method for the 10 sequences used in this experiment is 92.9%, that is roughly similar to HACA with 92.7%, while for ACA is lower, at 90.3%. Note however, that our method cannot find the repetitions within an action, like ACA and HACA algorithms (indicated by the white lines). In terms of computational time, our method requires 45 minutes for creating the motion word embedding (preprocessing for all motion clips), and then approximately 10 seconds to define the window's signatures and segmentation for each motion. The preprocessing time for ACA and HACA is 10 seconds for each motion clip, while motion segmentation is achieved in about 12 seconds for ACA and 20 seconds for HACA.

Table 2. Motion sequence retrieval results using signatures, compared to the Müller and Röder [2006], Sun *et al.* [2011], and Kapadia *et al.* [2013] methods.

	NN	Top 5	First Tier	Second Tier
Running	*100%*	*100%*	*100.0%*	*100.0%*
Salsa	*80%*	*70%*	*58.9%*	*82.2%*
Walk	*100%*	*96%*	*90.0%*	*100.0%*
Soccer	*80%*	*70%*	*62.2%*	*74.4%*
Basketball	*100%*	*78%*	*70.0%*	*97.8%*
Boxing	*80%*	*76%*	*68.9%*	*82.2%*
Swing	*100%*	*90%*	*91.1%*	*100.0%*
Carry	*90%*	*76%*	*66.7%*	*85.6%*
Jump	*80%*	*74%*	*61.1%*	*78.9%*
Sit	*80%*	*72%*	*62.2%*	*78.9%*
Climb	*80%*	*84%*	*65.6%*	*86.7%*
Placing tee	*60%*	*60%*	*65.0%*	*95.0%*
Placing ball	*40%*	*60%*	*60.0%*	*95.0%*
Swimming	*80%*	*60%*	*63.3%*	*75.6%*
Indian Bollywood	*70%*	*56%*	*66.7%*	*82.2%*
Ours (average)	83.6%	76.1%	71.0%	86.8%
Müller and Röder 2006	70.0%	67.4%	64.0%	76.5%
Sun *et al.* 2011	68.6%	68.8%	65.3%	75.8%
Kapadia *et al.* 2013	77.1%	72.0%	66.4%	82.2%

6.3 Motion Sequence Retrieval

The motion signatures of a whole sequence can also be used to extract similar sequences from a database. To illustrate this we created a database of 150 sequences of 15 different types of motions, 10 sequences for each motion. We then use the leave-one-out method and use each motion sequence as a query and measure the accuracy rate of retrieving the correct type of motions from the database. Table 2 shows our motion retrieval results. We measured the accuracy using four well known measures for database retrievals. NN indicates the average accuracy of the first results, Top-5 indicates the accuracy in the top 5 results. First Tier measures the ratio of correct retrievals in the top $K - 1$ results ($K = 10$ in our case) results to the total number of possible correct results (9 in our case). Second Tier measures the ratio of correct retrievals in the top $2(K - 1)$ results to the total number of possible correct results. Table 2 also reports the performance of the Müller and Röder [2006], Sun *et al.* [2011], and Kapadia *et al.* [2013] methods. As can be seen, in all measures we get very high accuracy rates using our signatures, and higher than any other method used, indicating that our signatures are descriptive and effective for encoding motion sequences. For this application, our method took approximately 6 hours to define motion signatures for all motion clips (preprocessing). Having represented motion sequences as low-rank vectors (motion signatures) for similarity measure, which largely reduces the run time for large-scale databases, it only took a couple of seconds to process a query and retrieve the most similar motions from the database. The computational time for Müller and Röder [2006] was 35 seconds for motion retrieval (approximately 450 seconds for defining motion templates), for Sun *et al.* [2011] was 370 seconds (plus approximately 1 hour to

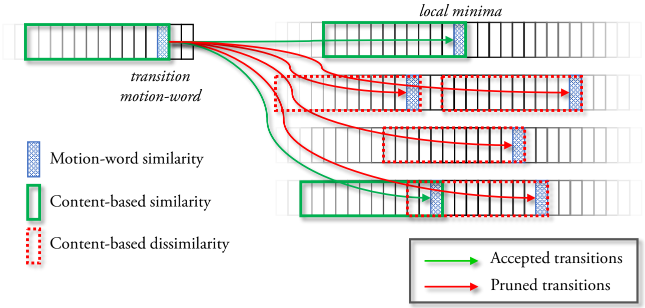

Fig. 14. Constraining motion synthesis using signatures. When constructing a motion transition graph we evaluate the contextual similarity between motions near the transition motion words and prune those that differ in content (red arrows), allowing only transitions that have a contextual similarity (green arrows).

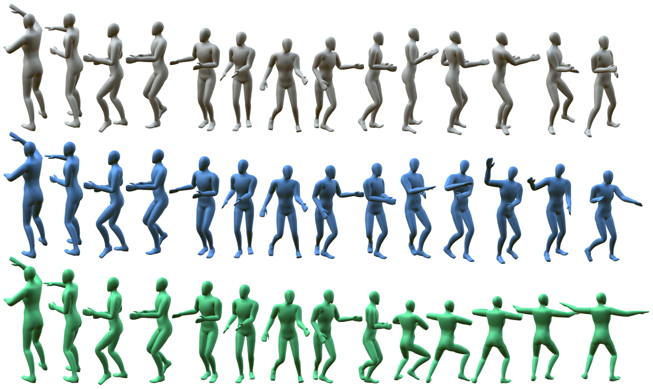

Fig. 15. Constrained motion synthesis example. The original motion is in gray, the contextually consistent synthesis is blue, and the unconstrained motion graphs synthesis is green. Using signatures constrains the transitions only to contextually similar motions (in this example, from one salsa motion to another), while the unconstrained motion graph may connect contextually dissimilar motions, such as salsa to Indian Bollywood style.

define the motion sequence volumes and decomposed vectors), and for Kapadia *et al.* [2013] was just 1-2 seconds (plus approximately 300 seconds to define motion keys).

6.4 Constraining a Motion Graph

To illustrate the usefulness of signature representation, we use them to constrain motion graph synthesis for contextual consistency. Motion synthesis allows connecting previously captured motion segments by finding transitions points where poses are similar. There is an extensive line of works regarding motion graphs, but most approaches focus on the geometric relations between specific body parts and pose similarity. Using signatures and motion words allows to connect motions with similar content that go beyond the body's postural configuration. We illustrate this on the well-known Motion Graphs method [Kovar et al. 2002], but it can be added to other, more recent, methods as well.

While building the graph of transitions between motion words we embed a contextual assessment of larger windows and prunes incoherent transitions (see Figure 14). We compare the content of the input motion, near the transition frame, with the content of the candidate transition motions at the transition points. We use a window of $n = 12$ motion words (a 2.5 seconds of motion), anchored on the transition motion word. Any transition with a large distance of motion signatures are discarded. Figure 15 illustrates an example of constrained motion synthesis using a dataset of 80 dances of different kinds. The synthesis achieved by only traversing motion graphs allows transitioning to contextually dissimilar motions, such as moving from salsa dance to Indian Bollywood style dance. Adding contextual similarity ensures only contextually consistent transitions.

7 LIMITATIONS AND CONCLUSIONS

Using motion words and deep learning we defined a method to extract motifs from motion sequences, and use them to define descriptive and succinct signatures based on the BoW representation. Motion words encode the local change of pose in a small window over time. The challenge of defining an effective distance measure between motion words was addressed by embedding them into a feature-space using a triplet loss convolutional neural network. In this universal feature space the distances reflect semantic similarity between motion words. We demonstrated the use of motion-motifs and signatures in several applications and evaluated their performance by comparing to several alternatives. The motifs we found are both frequent and descriptive movements that can characterize different motion types, and the signatures encode well the class of motions and can be used for recognition, retrieval, segmentation and synthesis. Our Bag-of-Motifs framework is invariant to the temporal ordering of motion motifs. Being oblivious to the motif's order is important in animation, especially when comparing two highly dynamic motions (e.g., dances), or motions with similar styles (e.g., salsa). This allows comparing motions with similar frequencies of motifs regardless of their exact temporal location. Our motion signatures allow comparing sequences of different speed and duration, enriching diversity in comparisons and analysis.

There are several limitations of our work. First, we have used a large dataset of motions, but many human motions are still not represented in our data and can possibly change the metric defined in our universal feature space. As data become available, the network can be retrained to learn the metric space again using the same method we employed. However, this process is time consuming; an optimized DTW implementation will significantly improve the computational time. Second, while building a signature of a motion sequence, we cluster *all* motion words of the sequence to the closest motifs. Some motion words may be outliers and cannot be correctly assigned to a cluster or motif. This may reduce the accuracy of the signature created. Third, because of the large variability of motion types our method does not define a universal signature for *all* motion types. To compare two motions, their signatures must be encoded in the same database. Creating a universal signature for all motions remains a problem for future research. Finally, even though the capability of dealing with temporal reorderings is one of the features of our method, it may leads to some false positive errors in motion retrieval. To make our method more sensitive to reorderings, one possible solution is to segment motion into shorter sequences and work independently. Another avenue of future research is to use other feature descriptors different than motion words to define signatures. For example, style words [Aristidou et al. 2017; Müller et al. 2005], that do not just encode the geometric features of the movement but also have stylistic or relational components.

ACKNOWLEDGMENTS

This research was supported by the Israel Science Foundation as part of the ISF-NSFC joint program grant number 2216/15; the work was also partially supported by Israel Science Foundation grant 2366/16, and the European Union's Horizon 2020 research and innovation programme H2020-WIDESPREAD-01-2016-2017-Teaming Phase 2 under grant agreement No 739578. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

Okan Arikan, David A. Forsyth, and James F. O'Brien. 2003. Motion Synthesis from Annotations. *ACM Trans. Graph.* 22, 3 (July 2003), 402–408.

Andreas Aristidou, Panayiotis Charalambous, and Yiorgos Chrysanthou. 2015. Emotion Analysis and Classification: Understanding the Performers' Emotions Using the LMA Entities. *Comput. Graph. Forum* 34, 6 (Sept. 2015), 262–276.

Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, and Ariel Shamir. 2018. Self-similarity Analysis for Motion Capture Cleaning. *Comput. Graph. Forum* 37, 2 (May 2018), 297–309.

Andreas Aristidou, Qiong Zeng, Efstathios Stavrakis, Kangkang Yin, Daniel Cohen-Or, Yiorgos Chrysanthou, and Baoquan Chen. 2017. Emotion Control of Unstructured Dance Movements. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '17)*. ACM, New York, NY, USA, 9:1–9:10.

Andreas Baak, Meinard Müller, and Hans-Peter Seidel. 2008. An Efficient Algorithm for Keyframe-based Motion Retrieval in the Presence of Temporal Deformations. In *Proceedings of the ACM International Conference on Multimedia Information Retrieval (MIR '08)*. ACM, New York, NY, USA, 451–458.

Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. 2004. Segmenting Motion Capture Data into Distinct Behaviors. In *Proceedings of Graphics Interface 2004 (GI '04)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 185–194.

Philippe Beaudoin, Stelian Coros, Michiel van de Panne, and Pierre Poulin. 2008. Motion-motif Graphs. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08)*. Eurographics Association, 117–126.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (Aug. 2013), 1798–1828.

Jürgen Bernard, Eduard Dobermann, Anna Vögele, Björn Krüger, Jörn Kohlhammer, and Dieter Fellner. 2017. Visual-Interactive Semi-Supervised Labeling of Human Motion Capture Data. In *Proceedings of Visualization and Data Analysis (VDA '17)*. Society for Imaging Science and Technology, 34–45.

Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. 2013. MotionExplorer: Exploratory Search in Human Motion Capture Data Based on Hierarchical Aggregation. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2257–2266.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Durell Bouchard and Norman I. Badler. 2015. Segmenting Motion Capture Data Using a Qualitative Analysis. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (MIG '15)*. 23–30.

Jinxiang Chai and Jessica K. Hodgins. 2005. Performance Animation from Low-dimensional Control Signals. *ACM Trans. Graph.* 24, 3 (July 2005), 686–696.

Min-Wen Chao, Chao-Hung Lin, Jackie Assa, and Tong-Yee Lee. 2012. Human Motion Retrieval from Hand-Drawn Sketch. *IEEE Transactions on Visualization and Computer Graphics* 18, 5 (May 2012), 729–740.

Songle Chen, Zhengxing Sun, and Yan Zhang. 2015. Scalable Organization of Collections of Motion Capture Data via Quantitative and Qualitative Analysis. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR '15)*. 411–418.

Myung G. Choi, Kyungyong Yang, Takeo Igarashi, Jun Mitani, and Jehee Lee. 2012. Retrieval and Visualization of Human Motion Data via Stick Figures. *Comput. Graph. Forum* 31, 7 (2012), 2057–2065.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*. 539–546.

CMU. 2018. Carnegie Mellon University MoCap Database: http://mocap.cs.cmu.edu/. (2018).

Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA.

Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cèdric Bray. 2004. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*. 1–22.

Zhigang Deng, Qin Gu, and Qing Li. 2009. Perceptually Consistent Example-based Human Motion Retrieval. In *Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D '09)*. 191–198.

DMCD. 2018. Dance Motion Capture Database: http://dancedb.cs.ucy.ac.cy/. (2018).

Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. 2012. What Makes Paris Look Like Paris? *ACM Trans. Graph.* 31, 4 (July 2012), 101:1–101:9.

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2017. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (April 2017), 677–691.

Matthew Field, David Stirling, Zengxi Pan, Montserrat Ros, and Fazel Naghdy. 2015. Recognizing Human Motions Through Mixture Modeling of Inertial Data. *Pattern Recogn.* 48, 8 (Aug. 2015), 2394–2406.

Kevin Forbes and Eugene Fiume. 2005. An Efficient Search Algorithm for Motion Data Using Weighted PCA. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '05)*. 67–76.

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned Neural Networks for Character Control. *ACM Trans. Graph.* 36, 4 (July 2017), 42:1–42:13.

Daniel Holden, Jun Saito, and Taku Komura. 2016. A Deep Learning Framework for Character Motion Synthesis and Editing. *ACM Trans. Graph.* 35, 4 (July 2016), 138:1–138:11.

Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning Motion Manifolds with Convolutional Autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs (SA '15)*. 18:1–18:4.

Bing Hu, Yanping Chen, Jesin Zakaria, Liudmila Ulanova, and Eamonn Keogh. 2013. Classification of Multi-dimensional Streaming Time Series by Weighting Each Classifier's Track Record. In *IEEE 13th International Conference on Data Mining (ICDM'16)*. 281–290.

Yueqi Hu, Shuangyuan Wu, Shihong Xia, Jinghua Fu, and Wei Chen. 2010. Motion track: Visualizing variations of human motion data. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*. 153–160.

Mohamed E. Hussein, Marwan Torki, Mohammad A. Gowayyed, and Motaz El-Saban. 2013. Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 2466–2472.

Mubbasir Kapadia, I-kao Chiang, Tiju Thomas, Norman I. Badler, and Joseph T. Kider, Jr. 2013. Efficient Motion Retrieval in Large Motion Databases. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '13)*. 19–28.

Ioannis Kapsouras and Nikos Nikolaidis. 2014. Action Recognition on Motion Capture Data Using a Dynemes and Forward Differences Representation. *J. Vis. Comun. Image Represent.* 25, 6 (Aug. 2014), 1432–1445.

Eamonn Keogh, Themistoklis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle. 2004. Indexing Large Human-motion Databases. In *Proceedings of the International Conference on Very Large Data Bases (VLDB '04)*. 780–791.

Lucas Kovar and Michael Gleicher. 2004. Automated Extraction and Parameterization of Motions in Large Data Sets. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 559–568.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion Graphs. *ACM Trans. Graph.* 21, 3 (July 2002), 473–482.

Björn Krüger, Jochen Tautges, Andreas Weber, and Arno Zinke. 2010. Fast Local and Global Similarity Searches in Large Motion Capture Databases. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*. Eurographics Association, 1–10.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (5 2015), 436–444.

Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. 2002. Interactive Control of Avatars Animated with Human Motion Data. *ACM Trans. Graph.* 21, 3 (July 2002), 491–500.

Fei-Fei Li and Pietro Perona. 2005. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*. 524–531.

Xiaosheng Li and Jessica Lin. 2017. Linear Time Complexity Time Series Classification with Bag-of-Pattern-Features. In *IEEE International Conference on Data Mining (ICDM*

’17). 277–286.

Jessica Lin, Rohan Khade, and Yuan Li. 2012. Rotation-invariant Similarity in Time Series Using Bag-of-patterns Representation. *J. Intell. Inf. Syst.* 39, 2 (Oct. 2012), 287–315.

Feng Liu, Yueting Zhuang, Fei Wu, and Yunhe Pan. 2003. 3D Motion Retrieval with Motion Index Tree. *Comput. Vis. Image Underst.* 92, 2-3 (Nov. 2003), 265–284.

Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan. 2005. A System for Analyzing and Indexing Human-motion Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD ’05)*. 924–926.

Libin Liu and Jessica Hodgins. 2017. Learning to Schedule Control Fragments for Physics-Based Characters Using Deep Q-Learning. *ACM Trans. Graph.* 36, 3 (June 2017), 29:1–29:14.

Xin Liu, Gao-Feng He, Shu-Juan Peng, Yiu-ming Cheung, and Yuan Yan Tang. 2017. Efficient Human Motion Retrieval via Temporal Adjacent Bag of Words and Discriminative Neighborhood Preserving Dictionary Learning. *IEEE Transactions on Human-Machine Systems* 47, 6 (Dec 2017), 763–776.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graph.* 36, 4 (July 2017), 44:1–44:14.

Meinard Müller, Andreas Baak, and Hans-Peter Seidel. 2009. Efficient and Robust Annotation of Motion Capture Data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’09)*. 17–26.

Meinard Müller and Tido Röder. 2006. Motion Templates for Automatic Classification and Retrieval of Motion Capture Data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’06)*. Eurographics Association, 137–146.

Meinard Müller, Tido Röder, and Michael Clausen. 2005. Efficient Content-based Retrieval of Motion Capture Data. *ACM Trans. Graph.* 24, 3 (July 2005), 677–685.

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. 2017. Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’17)*. 1–10.

Xue Bin Peng, Glen Berseth, Kangkang Yin, and Michiel Van De Panne. 2017. DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning. *ACM Trans. Graph.* 36, 4 (July 2017), 41:1–41:13.

Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’12)*. ACM, New York, NY, USA, 262–270.

Michalis Raptis, Darko Kirovski, and Hugues Hoppe. 2011. Real-time Classification of Dance Gestures from Skeleton Animation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’11)*. 147–156.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The Earth Mover’s Distance As a Metric for Image Retrieval. *Int. J. Comput. Vision* 40, 2 (Nov. 2000), 99–121.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’15)*. 815–823.

Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. 2012. *Unsupervised Discovery of Mid-Level Discriminative Patches.* Springer Berlin Heidelberg, Berlin, Heidelberg, 73–86.

Chuan Sun, Imran N. Junejo, and Hassan Foroosh. 2011. Motion Retrieval Using Low-Rank Subspace Decomposition of Motion Volume. *Comput. Graph. Forum* 30, 7 (November 2011), 1953–1962.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’15)*. 1–9.

Wataru Takano and Yoshihiko Nakamura. 2015. Symbolically Structured Database for Human Whole Body Motions Based on Association Between Motion Symbols and Motion Words. *Robot. Auton. Syst.* 66, C (April 2015), 75–85.

Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. 2011. Motion Reconstruction Using Sparse Accelerometer Data. *ACM Trans. Graph.* 30, 3 (May 2011), 18:1–18:12.

Matthew Thorne, David Burke, and Michiel van de Panne. 2004. Motion Doodles: An Interface for Sketching Character Motion. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 424–431.

M. Alex O. Vasilescu. 2002. Human motion signatures: analysis, synthesis, recognition. In *Proceedings of the International Conference on Pattern Recognition*, Vol. 3. IEEE, 456–460.

Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human Action Recognition by Representing 3D Skeletons As Points in a Lie Group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’14)*. 588–595.

Anna Vögele, Björn Krüger, and Reinhard Klein. 2014. Efficient Unsupervised Temporal Segmentation of Human Motion. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’14)*. Eurographics Association, 167–176.

Jing Wang and Bobby Bodenheimer. 2003. An Evaluation of a Cost Metric for Selecting Transitions Between Motion Segments. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA ’03)*. Eurographics Association, 232–238.

Yingying Wang and Michael Neff. 2015. Deep Signatures for Indexing and Retrieval in Large Motion Databases. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games (MIG ’15)*. ACM, New York, NY, USA, 37–45.

Ian H. Witten, Timothy C. Bell, and Alistair Moffat. 1994. *Managing Gigabytes: Compressing and Indexing Documents and Images* (1st ed.). John Wiley & Sons, Inc.

Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. 2017. Sampling Matters in Deep Embedding Learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV ’17)*. 2859–2867.

Shuangyuan Wu, Zhaoqi Wang, and Shihong Xia. 2009. Indexing and Retrieval of Human Motion Data by a Hierarchical Tree. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST ’09)*. 207–214.

Jun Xiao, Yinfu Feng, Mingming Ji, Xiaosong Yang, Jian J. Zhang, and Yueting Zhuang. 2015. Sparse Motion Bases Selection for Human Motion Denoising. *Signal Process.* 110, C (May 2015), 108–122.

Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamon Keogh. 2017. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *IEEE 17th International Conference on Data Mining (ICDM’17)*. 1317–1322.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Zachary Zimmerman, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2018. Time Series Joins, Motifs, Discords and Shapelets: A Unifying View That Exploits the Matrix Profile. *Data Min. Knowl. Discov.* 32, 1 (Jan. 2018), 83–123.

Byoung-Kee Yi, H. V. Jagadish, and Christos Faloutsos. 1998. Efficient Retrieval of Similar Time Sequences Under Time Warping. In *Proceedings of the Fourteenth International Conference on Data Engineering (ICDE ’98)*. IEEE Computer Society, Washington, DC, USA, 201–208.

Sergey Zagoruyko and Nikos Komodakis. 2015. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’15)*. 4353–4361.

Feng Zhou, Fernando De la Torre, and Jessica K. Hodgins. 2013. Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 3 (March 2013), 582–596.