Deep Signatures for Indexing and Retrieval in Large Motion Databases

Yingying Wang * Michael Neff †

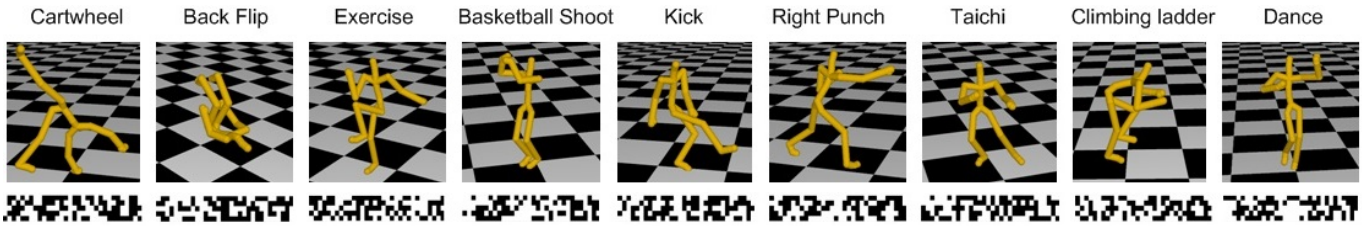


Figure 1: *Motion segments and their "deep signatures".*

Abstract

Data-driven motion research requires effective tools to compress, index, retrieve and reconstruct captured motion data. In this paper, we present a novel method to perform these tasks using a deep learning architecture. Our deep autoencoder, a form of artificial neural network, encodes motion segments into "deep signatures". This signature is formed by concatenating signatures for functionally different parts of the body. The deep signature is a highly condensed representation of a motion segment, requiring only 20 bytes, yet still encoding high level motion features. It can be used to produce a very compact representation of a motion database that can be effectively used for motion indexing and retrieval, with a very small memory footprint. Database searches are reduced to low cost binary comparisons of signatures. Motion reconstruction is achieved by fixing a "deep signature" that is missing a section using Gibbs Sampling. We tested both manually and automatically segmented motion databases and our experiments show that extracting the deep signature is fast and scales well with large databases. Given a query motion, similar motion segments can be retrieved at interactive speed with excellent match quality.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation, Virtual Reality ,

Keywords: motion retrieval, motion indexing, deep learning, character animation

Links: DL PDF

1 Introduction

To generate realistic and expressive human motion, people often turn to data-driven approaches. The effectiveness of these approaches generally relies on being able to work efficiently with large motion databases. This requires effective tools to compress, index, and annotate the data, and fast methods to explore the database and search for relevant motion clips. Working with motion capture data is difficult because it is a high dimensional temporal sequence, being inefficient to work with directly, and showing variation in both time and space. The temporal and spatial variations mean that motions that are logically similar may be numerically quite different, and vice versa.

This work addresses these challenges by using a special type of deep neural network, multichannel autoencoder, to produce "deep signatures" for motion clips. These signatures allow very compact representations of databases to be built for search. For example, our 1.4 GB test database can be represented by a 313 KB signature database. Clip comparisons are reduced to computing the difference between 20 byte binary strings. The approach is very efficient in time and space and produces excellent matches for query motions.

Researchers have recognized that it is inefficient to use high dimensional raw motion data directly, and inaccurate to measure motion similarity in the raw data space. Instead, extracting pertinent features from the data is crucial for motion compression, indexing, retrieval and annotation. Previous research [Forbes and Fiume 2005; Liu et al. 2005] extracted compressed numerical features to represent the original motion on its principle components, which are fast to compute and support numerical reconstruction of the original data. [Müller et al. 2005] and [Müller and Röder 2006] require offline pre-computation of framewise geometric features of the motion. The similarity measure, and the subsequent motion indexing and retrieval are accordingly performed in this new feature space. As observed by [Laban and Ullmann 1971], human motion can have shape features, energy features, body connectivity features and emotional features, some of which are difficult to quantify and measure in an explicit way.

Common motion retrieval algorithms which use PCA and ICA correspond to shallow architectures. Compared to low level, raw data and simple numerical similarity metrics in [Forbes and Fiume 2005; Liu et al. 2005], higher level features and logical similarity metrics are preferable for motion retrieval. Deep architectures such as the deep autoencoder use Restricted Boltzman Machines (RBMs) as building blocks, extract motion features at multiple levels of abstraction in hierarchical ways, and disentangle the varying factors from the underlying data at each step. They can model highly varying, complex functions with a concise representation and greater expressive power. The multi-level non-linear operations could require exponentially

*e-mail: yiwang@ucdavis.edu
†e-mail: neff@cs.ucdavis.edu

more computational elements in a shallow architecture. For large motion databases, deep learning can scale well with the size of the database and its computation time is close to linear.

Although our work also uses binary representation and is capable of high level motion retrieval, it is fundamentally different from [Müller et al. 2005; Müller and Röder 2006]. [Müller et al. 2005] requires user specified motion features (from their predefined 31 features), while our method is fully automatic - the "deep signature" of each motion clip in the database is automatically encoded, with no manual design and feature selection. Our method only takes query motion as input and is able to capture subtle stylistic features which are difficult for users to describe, e.g "taichi". Automatic discovery of motion abstraction is one main advantage of using our deep architecture. It avoids human intervention and allows batch processing of retrieval tasks, especially when people do not know what a good representation of motion features is. During the preprocessing step, manual feature definition could also ignore information in the motion data by excluding it from the feature set. Thus for large datasets, automatically extracted high level features, "deep signatures", suffer less information loss and are far more efficient to compute.

Deep architectures, including Deep Belief Networks (DBN), Stacked RBMs and deep autoencoders, have been successfully applied to image, video, audio, motion and natural language processing. Not only can they handle information from different media, for motion databases, they can also integrate data from different modality channels (including joint rotation vectors from different body parts and text description from annotations), work well with motion that is labeled or unlabeled, capture statistical regularity across different modalities, and fix missing motion parts automatically in a denoising fashion through Gibbs Sampling.

In this paper, we present novel methods for compressing, indexing, retrieval and reconstruction of large motion databases, based on extracting high level, non-linear "deep signatures" from motion data. We developed a stacked learning structure suitable for extracting features from multi-channel human motion data. The process begins by segmenting a large motion capture database into motion segments, using either automatic or manual methods. Each motion segment is divided into five channels: left arm, right arm, left leg, right leg and torso. We use the deep autoencoder to extract a 32-bit feature code for each channel in the segment and concatenate the five 32-bit feature codes into a "deep signature". Our deep signature is highly condensed, representing each motion segment by only 20 bytes, and thus the entire database can be compressed to fit into main memory, avoiding hard drive accesses. The deep autoencoder is capable of recognizing consistent high level features regardless of minor variations.

Our results consistently show that logically similar motions have similar signatures. Fast bitwise operation on the binary deep signature replaces the expensive real-valued computation normally used to compare motions during retrieval. We evaluate our method by experimenting on two different types of motion databases: one is the large scale, diversified CMU Motion Database [NSF0196217], that was automatically segmented using a velocity-based method. The other is a medium size subset of the CMU Motion Database with manual segmentation and annotations. Result shows that our methods work well for both motion databases.

We summarize the main contribution of our work as follows:

- We propose a novel method that uses a multi-channel deep autoencoder to extract a high level "deep signature" from motion segments.
- We demonstrate methods for indexing large motion databases based on the "deep signature", which can make the entire index database fit into main memory.
- We developed a fast retrieval method which can find similar motion clips in a large motion database at interactive speed.
- We propose an effective motion reconstruction method by fixing motion's "deep signature" using Gibbs sampling.

2 Related Work

With the development of motion capture techniques, human motion data has become a common, publicly available resource. Data-driven methods thus have become popular and important in animation research and industry for generating realistic motions. Some previous motion indexing and retrieval research uses raw joint rotations or marker positions to compare human motion [Kovar and Gleicher 2004; Keogh et al. 2004; Meng et al. 2008] or identifies motions by movement from separate body parts [Wu et al. 2009; Deng et al. 2009; Liu et al. 2003]. However, high dimensional, redundant motion data in a continuous space makes a raw data representation inefficient for motion comparison and identification. Feature extraction is becoming a common practice, and finding more compact and representative features that better capture the essences of the motion plays a crucial role. Forbes and Fiume [2005] reduces the original motion data to a weighted PCA space, thus the principal components are then used to identify each pose in the motion sequence. Liu et al. [2005] select principal markers from the raw motion data, and represent the motion sequence using the "transition signature" of poses. In [Krüger et al. 2010], experiments were conducted to evaluate the searching performance on large motion databases of selective joint rotation data, end-effector trajectory data, and PCA reduced feature data.

Logically similar motions are hard to compare numerically, given variations in both temporal and spatial domains for the same motion content. Higher level motion features are thus defined to try to match logically similar motions. The features used in [Kapadia et al. 2013] are a set of motion keys, including computed body contact, energy, balance, shape and other high level information. Muller et al. [2005] introduced an innovative approach using geometric relational features, and successfully applied them for content-based motion indexing, retrieval and segmentation. These geometric relational features are used to train motion templates for motion retrieval and annotation in [Müller and Röder 2006]. However, the definition and specification of high level features requires human intervention and relies on prior knowledge of motion content. It is difficult to guarantee completeness: data that is not involved in the feature computation is thus excluded and lost. Also these high level features require expensive offline pre-computation, especially for large motion databases.

A distance metric is another key ingredient in indexing and comparing motions. Ideally, a distance function should cluster variations of logically similar motions, distinguish motions of different content, and be fast to compute. For raw motion data or real-valued feature representation, Euclidean distance [Kovar and Gleicher 2004; Keogh et al. 2004] or its weighted [Forbes and Fiume 2005; Meng et al. 2008] versions are often used. Geometric relational features in [Müller and Röder 2006] are in binary, thus the Hamming distance is used for motion search. The L1 norm was used in [Keogh et al. 2004] to speed up the search. To address temporal variations, motion sequences are aligned before

feature comparison. Most previous research uses local scaling for temporal alignment either through DTW [Forbes and Fiume 2005; Müller and Röder 2006] or match web [Kovar and Gleicher 2004], while [Keogh et al. 2004; Argyros and Ermopoulos 2003] focus on uniform scaling for sequence matching. The distance computation (real-value operation in most cases) with additional temporal alignment (usually quadratic complexity) makes searching for matching motion clips a very expensive process.

To enhance the indexing performance and search execution time, different data structures have been utilized. Meng et al. [2008] reduce the size of the match-web to a compact matching trellis. Tree structures that support range search such as kd-tree [Krüger et al. 2010], r-tree [Keogh et al. 2004] and trie [Kapadia et al. 2013] are used for finding the nearest neighbors of the motion sample. Other data structure like graph, map are also frequently used for motion search [Chai and Hodgins 2005; Wu et al. 2009; Deng et al. 2009].

The idea of deep learning is inspired by the deep architectural structure of the brain, and Hinton et al. [2006] made a breakthrough by successfully training a deep network. Since then, deep architectures such as DBN [Hinton et al. 2006; Mohamed et al. 2012], stacked RBMs [Salakhutdinov and Hinton 2012; Salakhutdinov and Hinton 2009] and deep autoencoders [Hinton and Salakhutdinov 2006; Ngiam et al. 2011] have been frequently applied to different media: text documents [Hinton et al. 2006], images [Krizhevsky et al. 2012], video [Mobahi et al. 2009] and acoustic speech [Mohamed et al. 2012], for dimension reduction, classification or regression tasks. Previous research also proved that deep architectures like autoencoders [Ngiam et al. 2011] and DBNs [Srivastava and Salakhutdinov 2012] can extract features across multiple modalities, learn a joint representation from the space of multimodal inputs, and fill in missing modality data based on the captured statistical distribution. Taylor and Hinton [2009] use deep neural nets as a replacement of traditional HMMs to model human motion by treating motion as a temporal series of body poses. In comparison, our research is the first work trying to extract features - a "deep signature" from multi-channel human motion databases, and apply the extracted feature to motion indexing, retrieval and reconstruction.

3 Deep Signature Learning Architectures

In this section we introduce some basic deep learning concepts for encoding motion segments. We start with RBMs, the building block of deep architectures, and further introduce Gaussian RBMs that suit the real-valued joint rotation input of motion capture data. Last we discuss the deep autoencoder for extracting deep signatures from separate channels.

3.1 Restricted Boltzmann Machines

Restricted Boltzmann machines (RBMs) have been used as the building blocks for deep learning architectures. An RBM is an undirected graphical model with visible units $v \in \{0,1\}^D$ and hidden units $h \in \{0,1\}^L$. There are symmetric connections w between the hidden and visible variables, but no connections within hidden units or visible units. An RBM constructs a model that defines an energy function (1):

$$E(v,h;\theta) = -a^T v - b^T h - v^T w h \quad (1)$$

where the parameters θ in (1) are $\{a,b,w\}$. The RBM model captures the joint distribution over the visible and hidden units (2):

$$P(v,h;\theta) = \frac{1}{Z(\theta)} exp(-E(v,h;\theta)) \quad (2)$$

where $Z(\theta)$ is the normalizing constant.

3.2 Gaussian RBM

A Gaussian RBM is a special type of RBM that takes real-valued data as input units, $v \in R^D$ and outputs binary hidden units, $h \in \{0,1\}^L$. The energy function of Gaussian RBM is as follows:

$$E(v,h;\theta) = \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^{D}\sum_{j=1}^{L} \frac{v_i}{\sigma_i} w_{ij} h_j - \sum_{j=1}^{L} a_j h_j \quad (3)$$

where $\theta = a, b, w, \sigma$ are the parameters, and it captures the probability distribution:

$$P(v_i|h;\theta) = N(b_i + \sigma_i \sum_{j=1}^{L} w_{ij} h_j, \sigma_i^2) \quad (4)$$

In our learning structure, a Gaussian RBM is at the lowest layer, taking the real-valued joint rotations as input, and outputting binary bits for the higher layer RBMs.

3.3 Deep Autoencoder

The deep autoencoder [Hinton and Salakhutdinov 2006] is a special deep architecture that builds upon stacked RBMs and has frequently proven to perform better than Principal Component Analysis (PCA) in dimension reduction and data reconstruction. The encoding path compresses the input data to stochastic binary output in a non-linear curvy manifold, while the decoding path reconstructs the original data, see Figure 2.

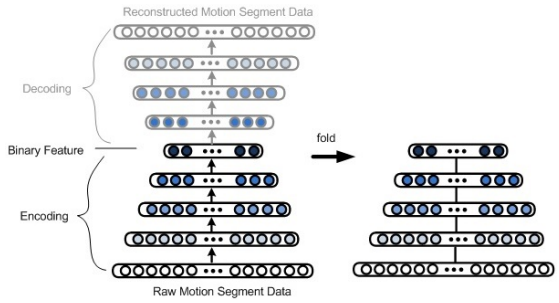


Figure 2: *Deep structure of autoencoder.*

In our work, for each motion channel, we construct a 4-layer deep autoencoder with the first layer being the gaussian RBM and the other 3 layers being normal RBMs. In this deep architecture, low layer output is used as high layer input. We use Contrastive Divergence (CD-1) for the pretraining. As in our application, no decoding path is used for generating motion segments, thus skipping back-propagation fine tuning works just fine.

3.4 Motion Deep Signature Extraction

Deep learning previously has been successfully applied to media like text [Hinton et al. 2006] and images [Krizhevsky et al. 2012]. To make it work for motion captured data, we perform three important process on the segments: Spatial Relocation, Temporal Unfolding and Channel Separation.

Spatial Relocation: This step serves to remove the impact of the difference in characters' scene positions. For each motion

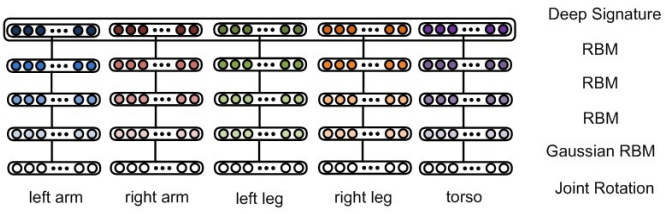


Figure 3: *Multi-channel deep autoencoder for extracting Deep Signature from motion segments.*

segment, we re-locate the character such that it starts at the origin of the scene, facing forward, thus characters' movements during motion segments are all relative to the same starting position.

Temporal Unfolding: Motion segments form a matrix where columns correspond to character DOFs and rows correspond to frames in time. To make it fit to the deep autoencoder in Section 3.3, we unfold joint rotation data along the timeline and make it a one-dimensional input vector.

Channel Separation: Different body parts have different motion patterns and active ranges. Treating joint rotations from different body parts as a whole input could cause mixed errors. Thus we separate body motions into five different channels: left arm, right arm, left leg, right leg and torso. In the multi-channel deep autoencoder in Figure 3, each channel extracts the binary code to represent its original motion in a compact way. The relationship across multiple channels can be modeled further using method in Section 4.3 if necessary. Other benefits of channel separation are that users can retrieve motion segments with preferred movement in specified channels e.g. punching in the arms (Section 4.2) or reconstruct movement of one body part given motions from other channels (Section 4.3).

To represent the whole motion segment, binary output from different channels is concatenated as the "deep signature" of the segment, as illustrated in Figure 3.

4 DS-based Indexing, Retrieval and Reconstruction

4.1 DS-based Motion Compression, Indexing and Database Organization

Motion segments are the fundamental units in our motion database. Original, captured motion clips can be segmented either manually or automatically. It is up to the users how they want to manually segment the motion clips. For large database like the CMU Motion Database with thousands of motion clips, it is less feasible to perform manual segmentation. We provide a very basic velocity-based automatic segmentation method.

We first compute the weighted velocity of joint rotations per frame. Torso joint rotations are given higher weights than end-effector joint rotations like the wrists and fingers. If the weighted velocity of frame i is a local minima, and below a velocity threshold, frame i is then detected as a segmentation point. We use this segmentation method to process large motion databases. Typically, an automatically segmented motion segment is about one hundred frames long and is subsampled to 50 frames. We discard short segments which are less than 20 frames long.

We compute a deep signature for every segment in the database. We leave it to the autoencoder to capture the temporal differences between the segments. The typical skeleton in CMU Motion Database has 62 DOFs. We separate its rotations value into five different channels, and discard DOFs with little significance, such as the fingertips and foot toes. Each channel is processed separately in the deep autoencoder. The first layer input for the five channels are 50×7 (left arm), 50×7 (right arm), 50×4 (left leg), 50×4 (right leg) and 50×15 (torso) rotations unfolded into one-dimensional real-valued vectors. We set the training epoch to 50, and the 4 binary layers output 200 bits, 128 bits, 64 bits and 32 bits, bottom to top. The top 32-bit outputs from the five channels are concatenated as the 20-byte "deep signature" for the motion segment. (Larger top outputs such as 64 or 128-bit per channel work fine. We choose 32-bit outputs per channel as they still provide decent results with smaller memory requirement.)

Compared to framewise PCA, the "deep signature" only takes 20 bytes of memory per segment. The high compression rate is not entirely from the data granularity. The deep autoencoder itself is a better dimension reduction tool, providing flexible non-linear compression. Previous research in document retrieval shows that a 10 dimensional autoencoder output works even better than 50 dimensional PCA. The high compression rate make it possible to load the entire large motion database during runtime.

We maintain a hash table in memory that associates each "deep signature" as the ID with a path pointing to the motion's hard drive location. All the relative motion operations such as retrieval and reconstruction (described in the following sections 4.2 and 4.3) can be processed based on the signature in memory before accessing the hard drive to retrieve the motion clip.

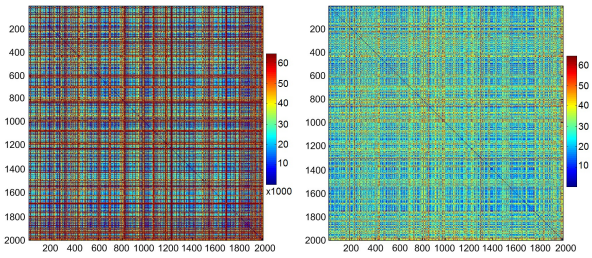
4.2 DS-based Motion Retrieval

The extracted "deep signatures" of the motion segments represent the original data as a binary string. Thus it allows us to use binary Hamming distance as the metric of difference between the original motions. Compared to commonly used Euclidean distance, Hamming distance is basically bit operation, and thus fast to compute. The Hamming distance between two deep signatures is the number of bits that differ. To verify the validity of using the Hamming distance of deep signatures to measure motion difference, we randomly selected 2000 automatically segmented motion segments from the CMU motion database. We calculate both Euclidean distance with dynamic time warping, and the Hamming distance for these motion segments. Then we ran the Pearson Rank Correlation between the two distance metric, the correlation coefficient is 0.607 ($p < .001$). The plotted distance matrices are illustrated in Figure 4.

As a "deep signature" is comprised of binary codes from five different channels, our motion retrieval method allows user to set binary masks to ignore irrelevant motion channels and only focus on searching motion channels of interest. We also support using a weighted Hamming distance as difference metric as described by (5)

$$HD_w(m,n) = \sum_{ch} w_{ch} * HD(m_{ch}, n_{ch}) \tag{5}$$

The "deep signature" also supports fast search for long motion sequence that consists of multiple segments. Given two sequences of "deep signatures" calculated on the segments in long motion clips, a Hamming distance with Dynamic Time Warping can be calculated



(a) Euclidean Distance Matrix (b) Hamming Distance Matrix

Figure 4: *Comparison between two distance metric: (a) Euclidean Distance with Dynamic Time Warping (b) Hamming Distance using a random sample of 2000 motion segments.*

at a coarser granularity (per segment). In Figure 5, each entry is a Hamming distance between motion segments instead of a frame-wise Euclidean distance of the long sequences, and the minimum cost path that provides the best alignment can be found.

4.3 DS-based Motion Reconstruction

As mentioned in Section 3.3, due to the subtlety of human motion, we would not recommend generating synthetic motion directly from "deep signature" using the downward path of the autoencoder, even though "deep signature" has less reconstruction error than PCA. In this section, we talk about how to reconstruct partially corrupted motion in the captured data, by filling in the missing binary feature code in its "deep signature". For example, movement of the left arm could be messed up during motion capture, and our motion reconstruction method can help synthesize plausible left arm movement based on movements of other body channels. The motion reconstruction function infers plausible motion for one body channel given motions of the rest body channels using the motion "deep signature".

Based on the multi-channel deep autoencoder in Figure 3, we add an extra RBM on top of the "deep signature" layer, see Figure 6. The top RBM takes the 160 bit "deep signature" as input, and outputs 128 bits binary data which captures the probability distribution of motion from all channels. The top RBM is trained in a denoising fashion: for the training motion segments, its channel binary code in "deep signature" input is randomly set to zero.

Given a motion segment s_c with corrupted movement in one

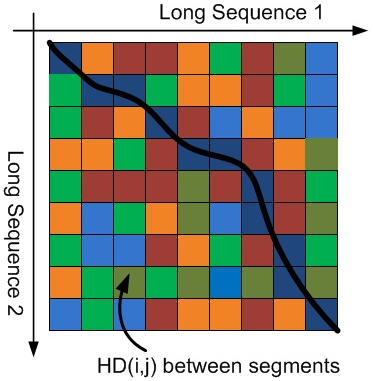


Figure 5: *Temporally aligned long motion sequence matching using Hamming distance of "deep signatures".*

motion channel, we first extract the "deep signature" using multi-channel deep autoencoder. As described in Section 3, the five channels are processed separately. For the non-corrupted channels, meaningful binary channel code can be extracted. For the corrupted channel, the 32 bits are set to zero. Binary code from all channels are then concatenated into the input "deep signature" d_c to the top RBM. We run Gibbs sampling for the top RBM to fix the zero bits in the corrupted channel. A new "deep signature" d_r is generated with reconstructed binary code in the corrupted channel for the original segment s_c.

We retrieve the most similar motion segment s_r in database using the reconstructed "deep signature" d_r as the query, and use the movement in s_r for the corrupted channel in s_c. Temporally s_c and s_r are aligned using DTW based on the movements in non-corrupted channels. Thus the corrupted channel in s_c is fixed by filling in the aligned movement in s_r. DTW alignment only needs to be run once when the nearest match s_r is found, while the retrieval part is a fast binary operation. Thus our "deep signature"-based motion reconstruction is highly efficient in runtime.

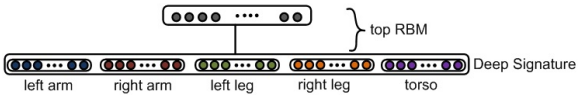


Figure 6: *An extra RBM is added on top of "deep signature" for motion reconstruction.*

5 Experiments and Results

5.1 Motion Compression, Indexing and Database Organization

In this experiment, we use all the motion clips under "Motion Categories" in the CMU motion capture website (http://mocap.cs.cmu.edu/motcat.php) to construct a large motion database. This database contains various human motions including Environment Interaction, Locomotion, Physical Activities and Sports (e.g. boxing, dancing), Situations and Scenarios (e.g. pantomime, gestures), totaling 607 files in 19 categories, 1,810,082 frames, approximately 252 minutes long and 1.4 GB in size. After automatic segmentation (described in Section 4.1) for all these motion files, we get 16,045 motion segments. Though each motion file has a general category description, this information is too rough to describe the motion details in the clip. For example, there are plenty "walking" and "jumping" motion segments in clips belonging to the category "play ground". Using the category description for the motion files could be inaccurate and misleading for the segments. Thus in our motion database, all the motion segments are put together, regardless of its file category.

By using the multi-channel deep autoencoder, motion segments are compressed and indexed using their "deep signatures", which is 20 byte per segment, about 313 KB for the entire motion database. This size can easily fit into the main memory or even most modern caches in runtime, and database-wide operations become far more affordable. We maintain a hash table to associate each "deep signature" to the hard drive path. The motion segment is loaded into memory only when necessary. For the manually segmented motion database, our method works in the same way. Figure 1 illustrates some representative "deep signatures" for motion segments of various categories, more examples are shown in Figure 7.

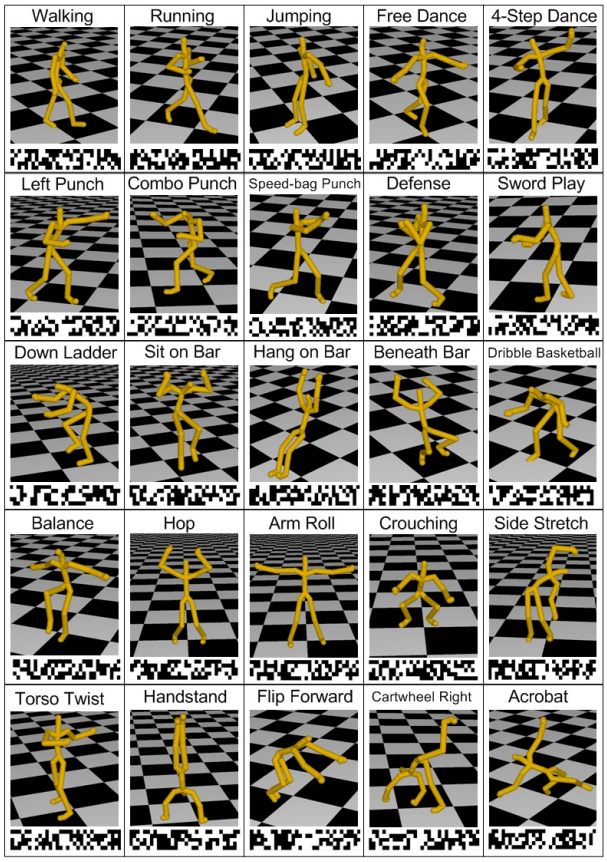


Figure 7: *Motion segments and their "deep signature" for indexing.*

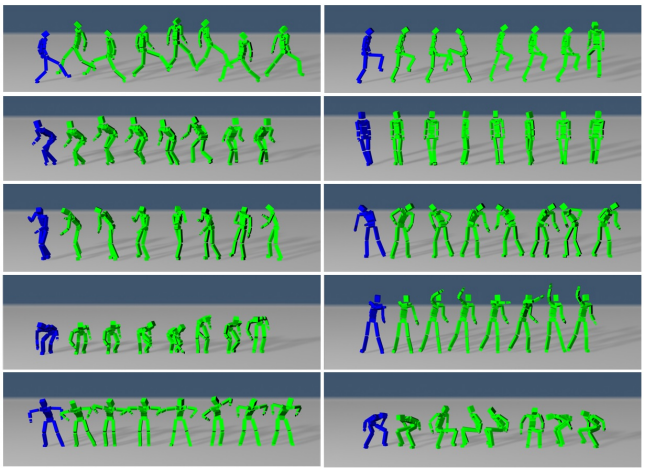


Figure 8: *Retrieval results for the automatically segmented databases. Query motions are displayed in blue and the top 7 matched segments are displayed in green.*

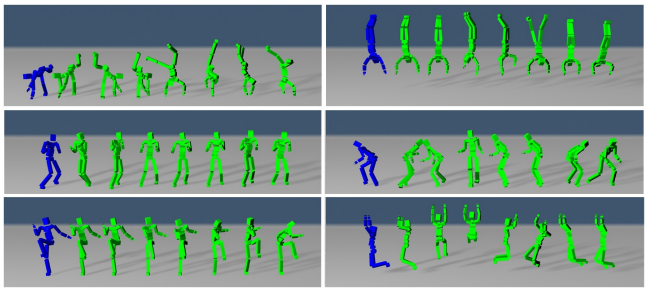


Figure 9: *Retrieval results for the manually segmented databases. "Cartwheel", "hand stand", "boxing defense", "dribbling basketball", "exercise" and "pull up" are illustrated.*

5.2 Motion Retrieval

To verify the efficiency and effectiveness of using the Hamming distance of "deep signatures" as the metric for motion retrieval, we performed two experiments using different motion databases. In the first experiment, we test the efficiency and realtime performance of our retrieval method by using a large motion database with automatically segmented data. To perform a further accuracy check and deepen our understanding of the retrieval quality, we use a second motion database - a medium sized motion database with manually segmented data.

5.2.1 Realtime Performance for a Large Database

In this experiment, the large motion database organized and indexed as described in 5.1 is used, which includes 16,045 motion segments in 19 heterogeneous categories. Given the size of the data, automatic segmentation is applied. The resulting motion segments in the db are typically 100 to 200 frames long, where boundaries are at velocity local minima below a certain threshold. Due to the automatic process, motion segments may or may not have semantic meaning.

During the retrieval, we only keep the "deep signatures" of the database in memory, which takes 313 KB. The original motion files are not loaded. A query motion segment is provided by the user, from which the query "deep signature" is extracted. We search the entire database items to find the closest matches. Bitwise operations are performed to compute the Hamming distance between the query and every segment in database. We support weighted search based on body channels, but for simplicity and generality, we just set all channel weights to 1 in this experiment. Segments with the nearest distances are returned as retrieval results and displayed for the user.

The retrieval process computes Hamming distances at runtime, no pre-computed distance matrix is stored, which greatly reduces the memory demand. The online retrieval using the large database takes approximately 512 ms to go through all the motion segments on a modest laptop machine (Intel Core 2 Duo CPU 2.53GHz, 4GB RAM). Figure 8 illustrates some of the examples of the query results, where the top 7 segments are displayed.

5.2.2 Retrieval Accuracy and Quality

To carefully evaluate and deepen our understanding of the quality of the retrieval method, we performed a second experiment by using a medium-sized motion database with manual segmentation. We selected a subset of motions from the CMU Database, including categories "playground", "acrobatics", "basketball", "boxing", "dance", "general exercise and stretching", "gymnastics", "martial arts" and "soccer". For the motion data, we performed manual segmentation and extracted 847 segments with clear semantic meanings, such as climbing ladder, dribbling basketball, etc. The duration of the manually extracted segments varies, from 40 frames to hundreds of frames, depending on the motion content. For segments with the same semantic meanings, there are various implementations, e.g. for boxing motions, we have straight right

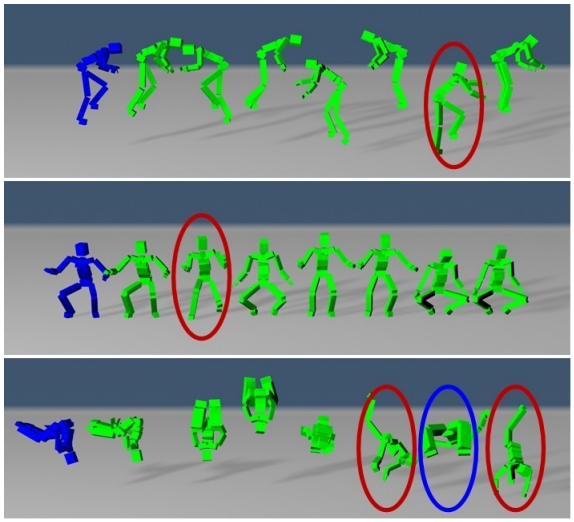


Figure 10: *Salient features are captured by "deep signatures". Red circled motion in row 1 is "upward ladder" among "downward ladder" results. Red circled motion in row 2 is "boxing dodge" among "crouching" results with similar knee bent. Red circled motions in row 3 are "cartwheel" and blue motion is a back flip ending with sitting on the ground.*

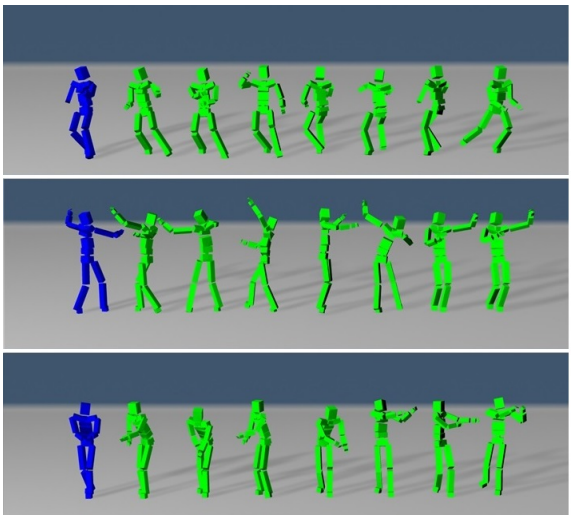


Figure 11: *Stylistic features are captured by "deep signatures". Motion segments, from top to bottom, are "tai chi", "dancing" and "sword play". In the top ranked results, the same style of motion with different implementations are retrieved.*

punch, right hook punch, as well as some other punch and dodge motions.

During the retrieval, users select a query segment from which the query "deep signature" is computed using the same method as described in 5.2.1. Semantic information is not used during retrieval. Retrieved results are the segments with closest Hamming distance to the query. We performed a careful analysis of the results and find that:

- *The Hamming distance between "deep signatures" can effectively find motions with the same content.* We tested segments with various semantic content as the query. The results show that segments with the closest Hamming distance are always those with the same semantic content. In Figure 9, the top ranked results for "cartwheel", "hand stand", "boxing defense", "dribbling basketball", "exercise" and "pull up" are illustrated. With the increase of distance, segments of different motions then start to appear in the result.

  While the results easily passed the visual inspection test, we ran a perceptual study to formally evaluate the retrieval quality and compare the two sizes of databases and the segmentation methods. We picked 12 retrieval cases for both databases. The top 7 ranked results to the query were displayed paired with the query, side-by-side, and subjects were asked if the motions matched. 11 subjects participated in our study. The mean matched value is $5.83/7$ for automatically segmented database and $6.08/7$ for manually segmented databases. A t-test shows no significant difference (t=1.5, p=0.136) with the increasing size of database and the different segmentation method, indicating that the technique scales well with the size of motion database and performs well with manual or automatic segmentation.

- *The "deep signature" has a deep understanding of the motion segments.* This high level understanding is achieved by capturing the most salient features in the motion segments. With increases of the Hamming distance from the query, we start to get result segments with varied implementations of the same motion.

  For example, for "downward the ladder" query, the top 5 result segments (Hamming distance 8 to 28) are all "downward ladder" regardless of the location and the number of steps. Our retrieval method starts to bring in "upward ladder" (Hamming distance 31, rank 6, see Figure 10 row 1) to the results as the Hamming distance increases. Segments ranked from 6 to 20 are a mixture of ""downward or "upward" ladder climbing. Another example is the "straight right punch", the top 9 result segments (Hamming distance 15 to 27) are all "straight right punch". Then "right hook punch" (Hamming distance 27, rank 10) and "combination punch" (Hamming distance 35, rank 17) start to show up in the results together with "straight right punch". The top 8 results for the "crouching" query are all "couching" except a "dodge" segment of boxing with very similar knee bending motion (Hamming distance 31, rank 2, Figure 10 row 2). The query of "back flip" first finds all the "back flips" in the database (Hamming distance 25 to 38, rank 1 to 4), then a failed "cartwheel" (Hamming distance 40, rank 5, Figure 10 row 3), "back flip ending with sitting on ground" (Hamming distance 45, rank 6) and "side flip" (Hamming distance 53, rank 8) show up in the results. For the query "speed bag punch", there are not enough similar segments in the database, so the returned top results are a mixture of "straight right punch" and "combination punch" (Hamming distance 39 to 50, rank 1 to 33). There are numerous other examples in our experiment. These examples prove that the "deep signature" is able to capture salient features like arm punch, knee bend and torso flip in segments despite of minor variations in other body part channels.

- *"Deep signature" is able to capture stylistic information in the motion segment.* In our database, we extracted 28 segments from the "tai chi" motion, all with different pushing and pulling moves. For a "tai chi" query, the top 8 ranked results (Hamming distance 35 to 44, see Figure 11 row 1) returned by our retrieval method are all "taichi" segments, although

they have different moves. For a "4-step dance" query, all the "4-step dance" segments in database are ranked as the top results (Hamming distance 10 to 29, rank 1 to 5). Then our method finds similar "2-step dance" (Hamming distance 34 to 38, rank 6 to 11, see Figure 11 row 2) as the results. For the query "sword play", the top 9 results are all "sword play" (Hamming distance 10 to 41, rank 1 to 9, , see Figure 11 row 3) despite all the different directions of the sword moves.

5.3 Motion Reconstruction

In this experiment, we reconstruct left arm motions from the rest four channels using method discussed in Section 4.3. The test clips were generated by removing motions from the left arm channel in the originally captured data. We evaluated DS-based reconstruction numerically by computing the difference between the reconstructed and original motions. For the reconstructed left arm, the average error is approximately 12.26° per DOF per frame. We also performed a visual plausibility check by randomly selecting 360 reconstructed clips. Results showed that 333 cases looked plausible and 27 were dissimilar to the original or looked unnatural. Figure 12 illustrates the top 7 reconstructed left arm motion in the "upward ladder" segment.

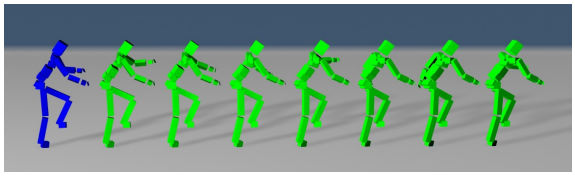


Figure 12: *Top 7 reconstructed left arm motion in "upward ladder" segment. Original motion segment is displayed in blue.*

6 Conclusion and Future Work

In this paper, we presented a novel approach for extracting binary "deep signatures" from motions using deep multi-channel autoencoder. We demonstrated that the "deep signature" is a highly compressed representation of the original motion, yet effectively captures the high level motion features. "Deep signature"-based motion indexing, retrieval and reconstruction methods are thus developed. For a large motion database, the DS compressed database can easily fit into main memory. The online motion retrieval uses the Hamming distance between "deep signatures" as the metric. Compared to the commonly used, real-valued Euclidean distance computation, the binary operation of Hamming distance can be completed with interactive speed, even for an extremely large heterogeneous motion databases. In our experiments, we have verified the effectiveness and efficiency of our retrieval method. The highly compressed motion representation and fast retrieval method are suitable for the new mobile devices. There are several ways we would like to extend and improve this work:

- During "deep signature" extraction, motion segments are all down-sampled, which may cause a loss of temporal information. Our experiment demonstrates the effectiveness of our retrieval method. To incoporate more careful timing control, we can use DS-based retrieval for pre-screening of candidates, and only perform the expensive DTW alignment for matched motion segments.

- We would like to evaluate the retrieval effectiveness by comparing our method to [Kovar and Gleicher 2004] and [Müller et al. 2005], but lack implementations. Though DS-based motion retrieval is faster by using the binary operation, and space efficient, side-by-side comparison would be more convincing to verify the retrieval quality.

- Our "deep signature" is in binary format like Muller et al. [2005], and also captures high level salient motion features. However, we cannot provide a way to interpret the semantic meaning of each bit in the signature. With the development of deep learning research, we hope to have a better understanding of the binary format of the extracted motion "deep signature".

In summary, the deep signature approach provides a method for indexing and retrieval on motion databases that is both fully automated and very efficient in time and space.

Acknowledgments

Financial support for this work was provided by the NSF through grant IIS 1115872.

References

ARGYROS, T., AND ERMOPOULOS, C. 2003. Efficient subsequence matching in time series databases under time and amplitude transformations. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, IEEE, 481–484.

CHAI, J., AND HODGINS, J. K. 2005. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics (TOG) 24*, 3, 686–696.

DENG, Z., GU, Q., AND LI, Q. 2009. Perceptually consistent example-based human motion retrieval. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ACM, 191–198.

FORBES, K., AND FIUME, E. 2005. An efficient search algorithm for motion data using weighted pca. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, 67–76.

HINTON, G. E., AND SALAKHUTDINOV, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science 313*, 5786, 504–507.

HINTON, G. E., OSINDERO, S., AND TEH, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation 18*, 7, 1527–1554.

KAPADIA, M., CHIANG, I.-K., THOMAS, T., BADLER, N. I., KIDER JR, J. T., ET AL. 2013. Efficient motion retrieval in large motion databases. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 19–28.

KEOGH, E., PALPANAS, T., ZORDAN, V. B., GUNOPULOS, D., AND CARDLE, M. 2004. Indexing large human-motion databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, VLDB Endowment, 780–791.

KOVAR, L., AND GLEICHER, M. 2004. Automated extraction and parameterization of motions in large data sets. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 559–568.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, vol. 1, 4.

KRÜGER, B., TAUTGES, J., WEBER, A., AND ZINKE, A. 2010. Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 1–10.

LABAN, R., AND ULLMANN, L. 1971. The mastery of movement.

LIU, F., ZHUANG, Y., WU, F., AND PAN, Y. 2003. 3d motion retrieval with motion index tree. *Computer Vision and Image Understanding 92*, 2, 265–284.

LIU, G., ZHANG, J., WANG, W., AND MCMILLAN, L. 2005. A system for analyzing and indexing human-motion databases. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, ACM, 924–926.

MENG, J., YUAN, J., HANS, M., AND WU, Y. 2008. Mining motifs from human motion. In *Proc. of EUROGRAPHICS*, vol. 8.

MOBAHI, H., COLLOBERT, R., AND WESTON, J. 2009. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 737–744.

MOHAMED, A.-R., DAHL, G. E., AND HINTON, G. 2012. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on 20*, 1, 14–22.

MÜLLER, M., AND RÖDER, T. 2006. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, 137–146.

MÜLLER, M., RÖDER, T., AND CLAUSEN, M. 2005. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics (TOG) 24*, 3, 677–685.

NGIAM, J., KHOSLA, A., KIM, M., NAM, J., LEE, H., AND NG, A. Y. 2011. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 689–696.

NSF0196217, C. Carnegie mellon university mocap database. In *http://mocap.cs.cmu.edu/*, Carnegie Mellon University.

SALAKHUTDINOV, R., AND HINTON, G. E. 2009. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, 448–455.

SALAKHUTDINOV, R., AND HINTON, G. E. 2012. A better way to pretrain deep boltzmann machines. In *NIPS*, 2456–2464.

SRIVASTAVA, N., AND SALAKHUTDINOV, R. 2012. Learning representations for multimodal data with deep belief nets. In *International Conference on Machine Learning Workshop*.

TAYLOR, G. W., AND HINTON, G. E. 2009. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, ACM, 1025–1032.

WU, S., WANG, Z., AND XIA, S. 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*, ACM, 207–214.