Perceptually Consistent Example-based Human Motion Retrieval

Zhigang Deng*
University of Houston

Qin Gu†
University of Houston

Qing Li‡
University of Houston

Abstract

Large amount of human motion capture data have been increasingly recorded and used in animation and gaming applications. Efficient retrieval of logically similar motions from a large data repository thereby serves as a fundamental basis for these motion data based applications. In this paper we present a perceptually consistent, example-based human motion retrieval approach that is capable of efficiently searching for and ranking similar motion sequences given a query motion input. Our method employs a motion pattern discovery and matching scheme that breaks human motions into a part-based, hierarchical motion representation. Building upon this representation, a fast string match algorithm is used for efficient runtime motion query processing. Finally, we conducted comparative user studies to evaluate the accuracy and perceptual-consistency of our approach by comparing it with the state of the art example-based human motion search algorithms.

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

Keywords: Character Animation, Motion Retrieval, Motion Pattern Extraction, Motion Capture, Hierarchical Human Motion Representation, and Perceptual Consistency

1 Introduction

Due to the subtlety and rich styles of human motions, it is extremely difficult for animators to manually make up natural and realistic human movements without considerable efforts. As a consequence, in recent years a large amount of human motion capture data have been increasingly recorded and used in various computer graphics, animation, and video gaming applications. The substantial amount of recorded human motion data imposes a challenging research problem: *given a large motion data repository, how to efficiently retrieve similar motion sequences based on a motion example and rank them in a perceptually consistent order?* Therefore, an efficient, perceptually consistent human motion retrieval scheme serves as a fundamental basis for these applications.

Search engines (*e.g.*, human motion search engines in the context of this work) are typically measured in terms of the following two criteria: (1) *Search Efficiency*. The search engines should be capable of retrieving logically relevant results from a huge data repository in an efficient way. (2) *Search Accuracy*. All the search engines automatically rank search results based on computational scores. Ideally, the computed rankings are expected to be consistent with humans' expectations, which is termed as a *perceptually-consistent* order in this paper.

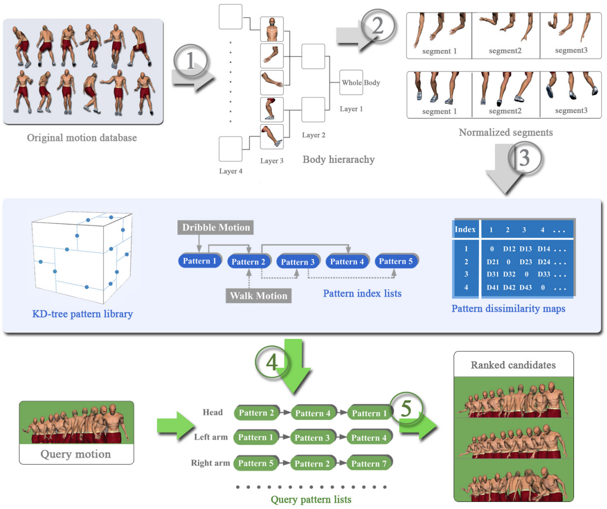

Figure 1: *Pipeline of this example-based human motion retrieval approach. Step 1: human hierarchy construction (Section 3.1); step 2: motion segmentation/normalization (Section 3.2); step 3: motion pattern detection and indexing (Section 3.3); step 4: hierarchical pattern matching (Section 4); and step 5: result ranking (Section 4.1).*

In this paper, we present a novel example-based human motion retrieval approach capable of efficiently retrieving logically similar motions from a large data repository in a perceptually consistent order, given a query motion as input. In this work, we use the term "*logically similar motions*" to refer to the motions that share the same action goal while having different variations. For example, kicking motions may have various variants such as front kicking and side kicking, but they are called "logically similar motions" in this paper.

Fig. 1 illustrates the schematic pipeline of our example-based human motion retrieval approach. Given a human motion repository, in the data preprocessing stage, our approach first divides the human body into a number of meaningful parts and builds a hierarchical structure based on the correlations among these body parts (Fig. 1, step 1). Then, a joint angle based motion segmentation scheme is employed for each part to partition the original motion sequences into part-based motion representations, followed by a segment normalization procedure (Fig. 1, step 2). After these steps, an adaptive K-means clustering algorithm is then performed upon these normalized motion segments to extract motion patterns (part-based representative motion) by detecting and grouping similar motion segments (Fig. 1, step 3). In this procedure, three types of data structures are generated to store the transformed motion representation including a KD-tree based motion pattern library for storing the details of the extracted motion patterns, motion pattern index lists, and a dissimilarity map recording the difference between any pair of motion patterns.

*e-mail: zdeng@cs.uh.edu
†e-mail: ericgu@cs.uh.edu
‡e-mail: qingli12@cs.uh.edu

I3D 2009, Boston, Massachusetts, February 27–March 1, 2009.
© 2009 ACM 978-1-60558-429-4/09/0002 $5.00

In the runtime, given a query motion sequence, our approach first chops it to motion segments and further creates its corresponding motion pattern lists by matching existing patterns in the preconstructed motion pattern library (Fig. 1, step 4). Then, we extend a fast string matching algorithm [Knuth et al. 1977] with a novel hierarchical decision-fusing scheme to efficiently retrieve logically similar motion sequences and rank them accordingly (Fig. 1, step 5).

The major distinctions of this work include: (1) *Flexible search query.* Our approach can take flexible search queries as input, including a human motion subsequence, or a hybrid of multiple motion sequences (*e.g.*, the upper body part of a motion sequence and the lower body part of another motion sequence). In addition, users can specify varied significances (importances) to different human body parts. The significances of different human parts (as a user-preference profile) are incorporated into the search algorithm, which ensures the research results are customized and ranked properly to maximally meet the expectations of individual users. (2) *Perceptually consistent search outcomes.* We conducted comparative user studies to find out the correlations between the search results ranked by computer algorithms including our approach and the supposedly ideal results ranked by humans. Our study results show that the motion sequences retrieved by our approach are more perceptually-consistent than current example-based human motion search algorithms [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005].

2 Related Work

Researchers have pursued various ways of editing and reusing the captured human motion data while maintaining its intrinsic realism [Fod et al. 2002; Kim et al. 2003; Lee et al. 2006; Yu et al. 2007]. Recent efforts include constructing and traversing motion graphs [Kovar et al. 2002; Arikan and Forsyth 2002; Beaudoin et al. 2008], example-based motion control or synthesis of constrained target motion [Lee et al. 2002; Hsu et al. 2004], learning statistical dynamic models from human motion [Brand and Hertzmann 2000; Li et al. 2002], searching in optimized low dimensional manifolds [Safonova et al. 2004; Grochow et al. 2004], and simulating biped behaviors [Sok et al. 2007].

How to efficiently and accurately retrieve proper motion sequences from a large unlabeled motion repository is a challenging problem. Currently, the popularized human motion retrieval idea is to transform original high-dimensional human motion data to a reduced representation and then conduct search in the reduced space [Agrawal et al. 1993; Faloutsos et al. 1994; Chan and Fu 1999; Liu et al. 2003; Chiu et al. 2004; Müller et al. 2005; So and Baciu 2006; Lin 2006]. For example, Faloutsos *et al.* [1994] present an effective subsequence matching method by mapping motion sequences into a small set of multidimensional rectangles in feature spaces. Keogh *et al.* [2004] present a technique to accelerate similarity search under uniform scaling based on the concept of bounding envelopes. Liu *et al.* [2005] index motion sequences by clustering every pose in the database into groups that are represented as simple linear models. Then, a motion sequence can be represented as the trajectory of the pose clusters it goes through. Forbes and Fiume [2005] project motion sequences into a weighted PCA space where the weights are either user-defined or based on the significances of different parts of the human body. Then, they use a dynamic time warping algorithm to measure the similarity between two motion sequences.

Kovar and Gleicher [2004] proposed an efficient motion search and parameterization technique by precomputing the “match webs” to describe potential subsequence matches between any pair of motion sequences. In their approach, a multi-step search strategy is also employed to incrementally and reliably find motions that are numerically similar to the query. However, for a large human motion dataset, it may not be always feasible to build the match webs for every possible pair of motions in advance. In other words, when a new motion query (not existing in the database), this approach need to pre-compute all the match webs between this query and existing motions in the database, which is not efficient in many cases. Our approach does not need to perform precomputation between a novel query motion and the existing motions, but fast transform the query motion into a part-based, high-level representation in the runtime.

Based on the observation that numerically similar human motions may not be logically similar, Müller and his colleagues [Müller et al. 2005; Müller and Röder 2006] present an efficient semantics-based motion retrieval system where users provide a query motion as a set of time-varying geometric feature relationships. In their approach, the motion retrieval problem is essentially transformed to a binary matching problem in a geometric relationship space. This approach successfully demonstrates its scalability, versatility, and efficiency. However, specifying well-defined geometric (semantic) relationships for highly dynamic human motions (*e.g.*, boxing) is non-trivial, especially for novice users.

3 Human Motion Data Preprocessing

The motion data preprocessing step (steps 1, 2, and 3 in Fig. 1) consists of three main steps: *body hierarchy construction* that partitions the human body into a number of parts in the spatial domain, *motion segmentation and normalization* that segment part-based human motions and then group them into a set of basic motion prototypes (called motion patterns), which essentially partitions human motions in the temporal domain, and *motion pattern extraction* that detects and discovers patterns by grouping similar motion segments.

3.1 Body Hierarchy Construction

A hierarchical human structure illustrated in Fig. 2 is constructed based on the spatial connectivity of the human body. The whole human body is first divided into ten meaningful basic parts, *e.g.*, head, torso, left arm, left hand, *etc.* and then a hierarchy with four layers is built accordingly. The hierarchy includes a total of eighteen nodes: ten leaf nodes stand for basic body parts, the parent nodes in the middle layers correspond to meaningful combinations of child nodes, and the root node represents the entire human body.

We choose a human hierarchy representation, because it provides a logical control granularity [Gu et al. 2008]. Furthermore, a multilayer hierarchy naturally take care of the correlations among several human parts that will be exploited for follow-up motion similarity computation (Section 4.1). In this work, we use joint angles rather than 3D marker positions for representing human motion data, due to the fact that the joint angle representation is convenient for unifying the motions of human bodies with different bone sizes [Beaudoin et al. 2008].

3.2 Motion Segmentation and Normalization

Typically, raw data from motion capture systems are long motion sequences with high variances, *e.g.*, tens of minutes. However, the motions of interest in many applications are shorter motion clips that satisfy users’ specific requirements. Therefore, an automated motion segmentation process that adaptively chops the long mo-

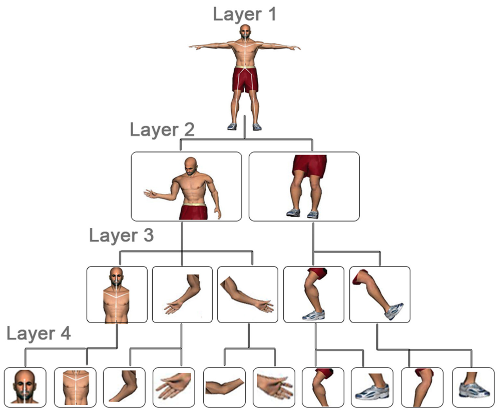

Figure 2: *Illustration of the constructed human hierarchy used in this work.*

tions into short clips is necessary for the later motion retrieval algorithm.

A number of techniques had been developed for human motion segmentation. For example, the angular acceleration of motion has been extensively used for motion segmentation and analysis [Bindiganavale 2000; Zhao 2001; Fod et al. 2002; Kim et al. 2003]. Badler and Bindiganavale [1998] use the zero-crossings of the second derivatives of the motion data to detect significant changes in the motion, *e.g.* spatial constraints. Li *et al.* [2007] employ multi-class Support Vector Machine classifiers to segment and recognize motion streams by classifying feature vectors. Gu *et al.* [2008] use the weighted sum of all the marker velocities as a threshold to segment human motion sequences. Principal Component Analysis (PCA) was also proposed for motion segmentation [Barbic et al. 2004], which is based on the assumption that the intrinsic PCA dimensionality of a motion will increase dramatically when a new type of motion is detected.

In this work, we use the Probabilistic Principal Component Analysis (PPCA) method for motion segmentation proposed by Barbic *et al.* [2004]. Besides dimension reduction, the PPCA-based method [Barbic et al. 2004] further computes a probability value for each data point to measure how well the data point is fitted into the existing data distribution. If a peak value is detected right after a valley value, and the difference between them is larger than a threshold R, then a new segmentation point is claimed. Since we process each body part separately, R value used in this work is expected to be smaller than the original threshold value used in [Barbic et al. 2004].

This motion segmentation process is applied to the joint angle representation of human motion data. Also, for each body part in the constructed human hierarchy (Fig. 2), a separate motion segmentation process is performed. Due to its adaptivity, the chopped motion segments typically have different numbers of frames. A normalization procedure therefore is applied to normalize all the motion segments to have the same number of frames (termed as *the reference frame number*). In this work, we compute the average frame number from previous segment as the reference frame number. Cubic Spline interpolation is applied to generate new frames during this normalization process. Table 1 shows the average frame numbers of motion segments (before normalization) and the variances of eight selected body parts.

Body parts	Head	LeftHand	LeftArm	RightArm
Average frames ± variance	18.32 ± 2.32	8.43 ± 6.43	11.39 ± 6.54	12.75 ± 7.34
Body parts	Torso	RightLeg	LeftFoot	RightFoot
Average frames ± variance	13.43 ± 5.65	11.24 ± 5.12	6.75 ± 5.35	6.05 ± 5.88

Table 1: *The average frame numbers of motion segments (before normalization) and the variances of eight selected body parts.*

3.3 Motion Pattern Extraction

A *motion pattern* is a representative motion segment for a node (*i.e.*, a body part) in the constructed human hierarchy (Fig. 2). To extract motion patterns from the normalized motion segments, an adaptive K-means clustering algorithm [Dehon et al. 2001] is used to group similar motion segments together.

Our motion pattern extraction algorithm can be described as follows. Starting from an initial K (the number of clusters), the K-means clustering algorithm is applied to group the normalized motion segments. Meanwhile, an error metric is computed to measure the quality of the clustering. If the error metric is larger than a specified threshold, then K is increased and the K-means clustering algorithm is called again. This iterative process continues until the clustering error metric is smaller than the threshold. Decreasing the threshold value will generate more clusters at the expense of more computing time. In this work, the initial value of K is experimentally set to 10. The error metric can be defined in the following two ways: the first is the average clustering error, and the other is the maximum clustering error. In our experiment, we found that the maximum error metric gave better results, although its computation is more time-consuming.

As described in the above Section 3.2, the threshold parameter R is used in the above PPCA-based motion segmentation process for the purpose of balancing search accuracy and algorithm runtime efficiency. In our implementation, we experimentally set R to 0.05, 0.1, 1.0 and 15 for level 4 to level 1, respectively. We choose larger R values for body parts at upper levels, because a body part at an upper level usually has a larger motion variation than a part at a lower level due to its higher DOFs. For instance, the "left hand + left arm" part has more final clusters than that of the "left arm" part using the same segmentation and clustering parameters. Therefore, we increase R for the parts at upper levels to prevent the number of clusters being overwhelmingly large. This will greatly speed up the processing time and save the storage space, and the resulted accuracy loss can be made up by the follow-up hierarchical similarity score propagation from lower levels (Section 4.1). Table 2 shows the final cluster numbers of 18 parts of the body hierarchy when the maximum error metric and four databases with different sizes are used.

In the procedure of the human motion data preprocessing, the following three data structures are generated: (1) *Motion Pattern Library:* The mean (representative) motion segments, obtained from the above motion pattern extraction process, are stored as motion patterns in a library. We store the motion pattern library as a kd-tree structure [Moore and Ostlund 2007] due to its efficiency. (2) *Motion Pattern Index Lists:* Each human motion sequence is transformed to a total of eighteen lists of pattern indexes (each node in the hierarchy has a pattern index list). In this step, besides these pattern index lists, we also retain the mappings between these pattern index lists and the original motion subsequences. In other words, the pattern index lists (integer index streams) are used only for the search purpose. (3) *Pattern Dissimilarity Maps:* A dissimilarity

Database Size	56MB	456MB	976MB	1452MB
Level 1				
Wholebody	315	447	758	981
Level 2				
Upperbody	196	314	465	707
Lowerbody	282	402	628	955
Level 3				
Head+Torso	86	167	203	287
Left arm+hand	242	356	498	722
Right arm+hand	213	372	542	804
Left leg+foot	301	441	722	1024
Right leg+foot	288	408	685	978
Level 4				
Head	42	102	134	169
Torso	121	256	376	432
Left arm	133	275	311	405
Right arm	116	308	346	431
Left hand	277	645	902	1221
Right hand	234	667	822	1145
Left leg	115	245	356	412
Right leg	123	272	416	487
Left foot	392	778	1022	1313
Right foot	342	802	1156	1354

Table 2: *Final cluster numbers of the 18 parts of the body hierarchy in four motion databases with different sizes: 56MB (170 motions, 68,293 frames), 456MB (396 motions, 556,097 frames), 976MB (542 motions, 1,190,243 frames), and 1452MB (941 motions, 1,770,731 frames).*

value is computed between any pair of motion patterns based on their Euclidean distance. These dissimilarity values are stored in a map (called *pattern dissimilarity map*). Numerical similarity between two full-body motion sequences is not necessarily equivalent to their logical similarity mainly due to their high dimensionality. Since in this work we break the full-body motion into a part-based hierarchy, the motions of each part has a much lower dimensionality than the full-body motions.

4 Runtime Motion Retrieval

Algorithm 1 Runtime Motion Retrieval Algorithm

Input: the pattern index lists in the library $S[n][18][\,]$, the query pattern index lists of the query motion sequence $Q[18][\,]$, and the user-defined weight vector $W[18]$. Note that each motion sequence contains 18 pattern index lists.
Output: motion similarity rank list $R[n]$, motion similarity score vector $V[n]$.
Define: pattern index list similarity score matrix $M[n][18]$.

```
for $i = 1$ to $n$ do
    for $j = 1$ to 18 do
        $M[i][j]$= PatternIndexListSimilarityComp($S[i][j][\,]$, $Q[j][\,]$, $W[j]$)
    end for
end for
$V[n]$=Similarity_Propagation($M[n][18]$)
$R$ =QuickSort($V$)
return ($R$,$V$)
```

The runtime motion query processing module (steps 4 and 5, Fig. 1) first transforms a given query motion to its motion pattern index lists by locating the closest motion patterns in the pre-constructed pattern library. This transformation is done for each node in the human hierarchy tree separately. Note that this transformation step is very fast, since the adaptive K-means clustering step that occupies the majority of computing time in the motion data processing stage (Section 3) is not needed. Then, our approach computes similarity scores between the query pattern index lists and existing pattern index lists by extending the classical Knuth-Morris-Pratt (KMP) string match algorithm [Knuth et al. 1977], detailed in follow-up Section 4.1. Finally, the search results (motion sequences) are ranked by hierarchically fusing similarity scores.

Alg. 1 describes the runtime human motion retrieval algorithm. It includes two main processes: (1) *PatternIndexListSimilarityComp* computes the motion similarity scores of each body part between the pattern index list of the query motion and that of any existing human motion in the data repository. Note that each full-body motion sequence has been transformed to 18 pattern index lists (Fig. 2). (2) *Similarity_Propagation* computes the weighted rankings of retrieved motions by hierarchically propagating the similarity scores. Our approach also provides a control granularity on its search and ranking strategy as follows: it allows users to set their search preferences by assigning varied significances/weights to different human parts. For instance, users can specify the following search preference: 35% for the left arm, 45% for the right arm, 20% for the legs, and zero for other parts. These weights can be incorporated into the similarity propagation process later to rank the retrieved motions.

4.1 Computation of Motion Similarity

Since both the query motion and existing motion sequences in the data repository are transformed to pattern index lists, the core part of this motion search process is to compute similarity between two pattern index lists with different lengths. Given a pattern index list is an integer index sequence, we can formulate this motion similarity computation problem as a matching problem between two integer/character streams, and it can be efficiently solved using fast string match algorithms. Different from the traditional character string match case where the number of possible characters is fixed and limited, in this work we cannot always expect to find a perfect match between two motion pattern index lists due to the large number of distinctive 3D human motion patterns, *e.g.*, at a thousand level as shown in Table 2. Consequently, a "quasi-match" between two similar motion patterns need to be computed and taken into account in our pattern match process. The pre-computed pattern dissimilarity maps that store the distance measure between any pair of motion patterns are designed for this purpose.

In this work, we extend the classical Knuth-Morris-Pratt (KMP) string match algorithm [Knuth et al. 1977] for motion pattern similarity computation as follows. If the distance (via pattern dissimilarity maps) between two patterns is less than a pre-defined threshold $MaxTolerance$, then the two patterns are considered as a quasi-match. If the number of consecutive quasi-match pattern indexes is no less than a threshold $MinConNum$, the matching score is increased; otherwise, if a pattern index fails to find its quasi-match, the dissimilarity value between the two comparing indexes is subtracted from the matching score. At the end of the algorithm, the matching score is normalized based on the length of the matching pattern index list. Alg. 2 describes this KMP-based motion similarity computation algorithm. Its time complexity is $\Theta(N)$, where N is the number of motion sequences in the data repository. Setting $MaxTolerance$ to 0.02 and $MinConNum$ to 5 achieved good results in our experiments.

Using the above motion similarity computation algorithm (Alg. 2), we compute the matching/similarity scores between the query motion and the motions existing in the data repository, part by part (Fig. 2). As described in Section 3.1, the upper a node is in the human hierarchy tree, the more global motion information it contains; and vice versa. Furthermore, two motions may exhibit high local similarities while having notable differences from the global

Algorithm 2 KMP-based motion similarity computation algorithm

Input: a source pattern index list $S[\ ]$, a query pattern index list $Q[\ ]$, pattern dissimilarity map $DM[\ ][\ ]$, and a user-defined weight for the specific body part $Weight$.
Output: motion similarity score $Score$.

```
Compute next value vector $nextval[\ ]$ of the query pattern index list
$Score \leftarrow 0, MatchCount \leftarrow 0, i \leftarrow 0, j \leftarrow 0$
while $i <$Length($S$) do
    if $DM[S[i]][Q[j]] \le MaxTolerance$ then
        $MatchCount \leftarrow MatchCount + 1, i \leftarrow i+1, j \leftarrow j+1$
    else
        $MatchCount \leftarrow 0$
        $Score \leftarrow Score - DM[S[i]][Q[j]]$
        if $nextval[j] = -1$ then
            $i \leftarrow i+1, j \leftarrow 0$
        else
            $j = nextval[j]$
        end if
    end if
    if $MatchCount \ge MinConNum$ then
        $Score \leftarrow Score + Weight$
    end if
end while
$Score \leftarrow Score/$Length($S$)
return $Score$
```

perspective (Fig. 3 shows an example). Therefore, we want to fuse similarity scores at different layers for the purpose of result ranking. Its basic idea is to propagate the similarity scores of the nodes at the deeper (more local) layers up to their parent nodes (more global) in the hierarchical tree, and after every propagation, we update the similarity score list of the corresponding parent nodes. The propagation continues until the root node (the whole human body) is reached.

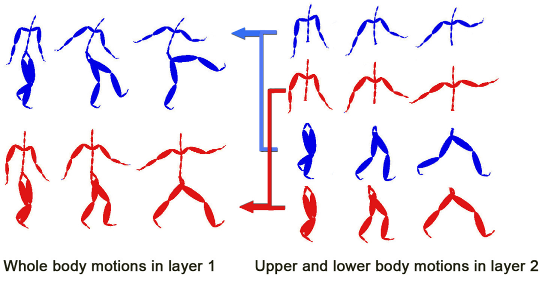
Whole body motions in layer 1
Upper and lower body motions in layer 2

Figure 3: *Two motion sequences have high local similarities (upper/lower part) but different global movements.*

In this similarity propagation process, the similarity scores of child nodes are propagated up to their parent nodes with an even ratio. For instance, assuming the similarity score list of the right foot node is S_{rf}, and the similarity score list of the right leg node is S_{rl}, then the propagated similarity score list $S_{prfl} = 1/2 \times S_{rf} + 1/2 \times S_{rl}$. Meanwhile, we computed a similarity score list for every parent node itself. We need to combine the propagated score list (from the child nodes) with that of the parent node. A parameter, α, is introduced to control the proportion between the global motion similarity (from the parent nodes) and the local motion similarity (propagated from child nodes). For instance, let S_{rfl} be the original computed similarity score list of the $RFoot + RLeg$ node, S_{prfl} be the list propagated from the child nodes, we update the similarity score list as $S_{urfl} = \alpha \times S_{rfl} + (1-\alpha) \times S_{prfl}$. Setting α=0.5 achieved good results in our experiments.

5 Results and Evaluations

To test and evaluate our approach, we collected a large set of human motion sequences from the public CMU motion capture data repository [cmu 2007] as our test dataset. In the remainder of this section, we will describe the processing time and storage statistics information of our approach, comparative accuracy experiment results, and user study experiments.

5.1 Time and Storage

To provide detailed time and storage of our method, we tested it on four datasets with different sizes (56MB, 170 motions,68,293 frames; 456MB, 396 motions, 556,097 frames; 976MB, 542 motions, 1,190,243 frames; and 1452MB, 941 motions, 1,770,731 frames). All experiments were conducted on a computer with a Intel Duo Core 2GHz CPU and 2GB memory. Table 3 shows its processing time and storage information. When computing the average search time per query, the average duration of the used query motions was about 10 seconds.

Database size (MB)	**56**	**456**	**976**	**1452**
Preprocessing time (min)	2	25	64	121
MPL size (MB)	0.45	2.04	3.42	4.28
MPIL size (MB)	0.37	7.30	11.7	18.6
PDM size (MB)	0.72	12.4	26.8	49.9
Search time per query (ms)	11	39	56	72

Table 3: *Search time, processing time, and storage information of our approach when motion databases with different sizes are used. Here MPL represents the constructed "Motion Pattern Library", MPIL represents the constructed "Motion Pattern Index Lists", and PDM represents the constructed "Pattern Dissimilarity Maps".*

5.2 Search Accuracy

To evaluate the search accuracy of our approach, we conducted an accuracy evaluation experiment similar to the one in [Kovar and Gleicher 2004]. Its basic idea is to use the same query motion examples to search in two different types of datasets: the first one (single type motion dataset) is the labeled motion dataset with the same semantic category (*e.g.*, walking), and the second one (mixed motion dataset) is the original, unlabeled motion dataset mixed with various motion categories. For this experiment, we manually collected a set of motions (56 MB) consisting of 170 sequences, 68,293 frames from the CMU motion capture dataset [cmu 2007]. Based on the original semantic labels of the motions, we divided this motion set into five semantic categories: walking, running, jumping, kicking and basketball-playing with 56, 41, 40, 16, and 17 motion sequences, respectively. It should be noted that each individual data file may still contained multiple actions. For example, a basketball-playing motion file may contains several walking cycles.

Based on the search outcomes from the above different datasets (in this experiment, we only consider the top N search results, N=20), we computed a true-positive ratio as the accuracy criterion that is defined as the percentage of the top N results searched from the mixed dataset are in the correct/expected single-type motion dataset. Here we assume the searched results from the labeled single-type motion dataset with the same category are ground-truth. In this experiment, we also compared our approach with three current example-based motion retrieval algorithms [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005]. Fig. 4 plots the computed true-positive ratios from this experiment.

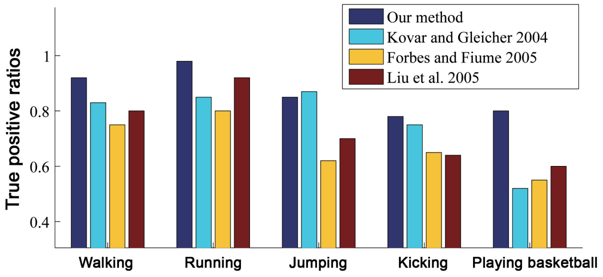

Figure 4: *Comparisons of the true-positive ratio achieved by the chosen four methods in our accuracy evaluation experiment.*

As shown in the above Fig. 4, in terms of the true-positive ratio, our approach generally slightly outperformed the other three methods except the jumping case where the true-positive ratio of the Kovar-Gleicher approach is slightly higher than that of our approach. In particular, our approach achieved a significantly higher true-positive ratio than the other three when querying basketball-play motions. We argue its main reason is that our motion retrieval approach is performed on both the motion segment level (temporal domain) and the part-based level (spatial domain), and thus is particularly suited for the cases where a complex motion sequence (the basketball-play motion in this experiment) is used as a query input.

5.3 Comparative User Study

We also compared our approach with three current human motion search approaches including the match-webs based motion search algorithm proposed by Kovar and Gleicher [2004], the piecewise linear space based algorithm proposed by Liu *et al.* [2005] and the weighted PCA based algorithm proposed by Forbes and Fiume [2005] through a comparative user study. It is noteworthy that we did not choose the semantic-based motion retrieval algorithms [Müller et al. 2005; Müller and Röder 2006] into this comparative user study, because it is difficult to perform sound comparisons between the semantic-based motion retrieval algorithms and pure example-based motion search algorithms due to their significant differences in input requirement: a set of geometric feature relationships over time [Müller et al. 2005; Müller and Röder 2006] versus one motion example required by our approach and the other three algorithms [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005].

In this study, we aim to study the following usability questions: (1) are the retrieved motions by our approach ranked in an approximate perceptually-consistent order? in other words, are they consistent with human perception? and (2) what is the comparative performance of our approach when comparing it with current example-based motion search approaches [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005]? We are aware that thoroughly and comprehensively evaluating the perceptual consistency of the searched motion results is another challenging research topic beyond this work. In our experiment, to approximate the perceptual outcomes from our humans, we asked experiment participants to subjectively rate the motion similarity between a query motion example and each of its corresponding searched motions (Fig. 5). The size of the motion dataset used in this study is about 456 MB, enclosing 396 different motion sequences and 556,097 motion capture frames. Motion types enclosed in the test dataset vary from simple motions (*e.g.*, walking, jumping, and running) to complex motions (*e.g.*, dancing, boxing, and climbing). Since the Kovar-Gleicher method [Kovar and Gleicher 2004] cannot take new query motion (not in the database) at runtime, we pre-computed the match webs between the chosen query motions and all the motions in our test dataset.

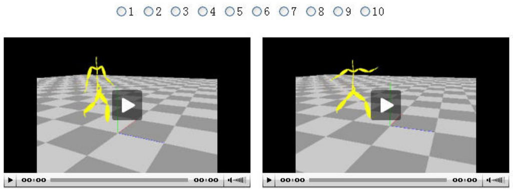

Figure 5: *An example of side-by-side comparison animation clips (one is a query motion example, and the other is one of search results) used in this study.*

We picked a walking motion, a running motion, and a basketball-playing motion as our three query motions. Given the same test inputs (the three chosen query motions), the top-ranked six results by each of the four approaches (our method, and the other three algorithms [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005]) were chosen into our user evaluation. In other words, a total of 72 searched motions (=3 examples × 4 approaches × 6 top-ranked motions per query) were used for this comparative user study. For each of the searched motions, we further expanded it to a corresponding side-by-side comparison clip (one side is a searched motion, and the other side is its corresponding query motion. Refer to Fig. 5). We then mixed the 72 side-by-side comparison clips and showed them in a random order to a total of twenty-four participants. Most of the participants are graduate students in sciences and engineering fields in a university, aging from twenty-two to thirty-five. The participants were first shown a short tutorial to demonstrate how to assign an appropriate similarity scale to a side-by-side comparison clip. In this study, the range of participants' similarity ratings is from 1 to 10, where 10 stands for "identical" and 1 stands for "completely different". After viewing each side-by-side comparison clip for a maximum of four times, the participants were asked to rate a similarity score based on their visual perception.

Based on the participants' ratings, we computed the average similarity ratings and the standard deviation achieved by the four chosen approaches (Table 4). As shown in Table 4, in most of cases, our approach achieved higher average similarity ratings than the other three, which is more obvious at the case of the basketball-playing motion query.

Query motion	Our method	Kovar and Gleicher 04	Forbes and Fiume 05	Liu et al. 05
Walking	7.92 ± 0.82	7.97 ± 1.07	7.09 ± 0.77	7.03 ± 0.61
Running	8.13 ± 1.48	7.69 ± 1.56	6.92 ± 1.45	7.26 ± 2.97
Basketball -playing	7.84 ± 1.58	4.92 ± 1.92	5.52 ± 2.27	4.45 ± 1.55

Table 4: *The average motion similarity ratings ± the standard deviations achieved by the four chosen approaches.*

To look into whether the searched motions by these four chosen approaches are ranked in an approximate perceptually-consistent order, we analyzed the same user rating data from a new perspective. Given a query motion example, any of the retrieved motion results, R_i, has a rank, C_i, assigned by the above four approaches (we called this ranking as its *computer-based ranking*). C_i is from 1 to 6, since we only chose the six top-ranked results from each query into this study. Meanwhile, based on the average user ratings, we also obtained a human-based perceptual ranking, H_i, for R_i. Fig. 6 shows the correlation between the computer-based rankings $\langle C_i \rangle$

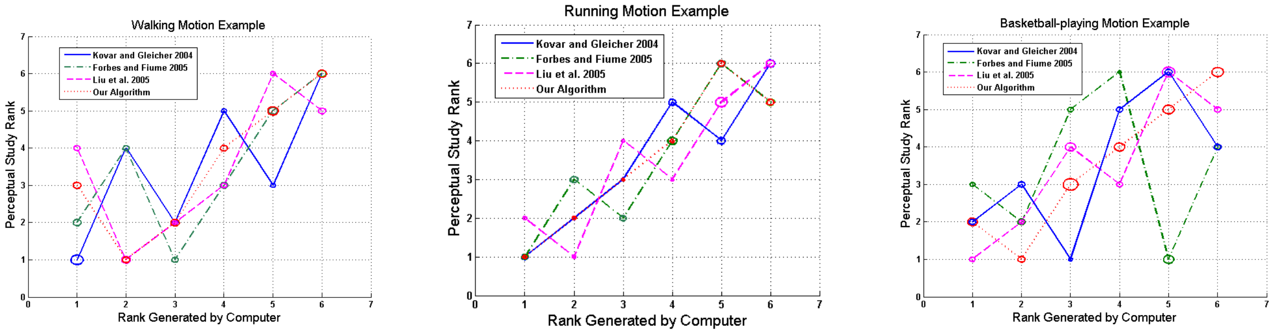

Figure 6: *The plotted correlation analysis of the computer-based rankings $\langle C_i \rangle$ versus the human-based perceptual rankings $\langle H_i \rangle$, for the three chosen query motions. Note that the radius of circles is proportional to the variance of the user ratings.*

and the human-based perceptual rankings $\langle H_i \rangle$. In its three subfigures, X axis is $\langle C_i \rangle$ and Y axis is $\langle H_i \rangle$. Therefore, the perceptual consistency of a motion search approach (*i.e.*, results are ranked in a perceptually-consistent order) is reflected in the linearity of plotted curves in these two figures.

To further quantify the linear correlation between $\langle C_i \rangle$ and $\langle H_i \rangle$, we performed the Canonical Correlation Analysis (CCA) [Dehon et al. 2000] on them, because CCA provides a scale-invariant optimum linear framework to measure the correlations between two streams. Table 5 shows the computed CCA coefficients for the chosen four approaches. These CCA coefficients reveal that our approach is more perceptually consistent than the other three approaches [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005].

Query motion	Our method	Kovar and Gleicher 04	Forbes and Fiume 05	Liu et al. 05
Walking	0.8285	0.7142	0.7142	0.5999
Running	0.9428	0.9428	0.8857	0.8857
Basketball -playing	0.9428	0.6571	0.0857	0.8857

Table 5: *The computed Canonical Correlation Analysis (CCA) coefficients of the four approaches for the three chosen query motion examples.*

6 Discussion and Conclusions

In this paper, we present a perceptually consistent, example-based human motion retrieval technique based on a hierarchical pattern extraction and matching scheme. Given a query motion, our approach can efficiently retrieve logically similar motions from a large motion data repository. The efficiency of our approach directly benefits from the fast performance of the classical KMP string matching algorithm [Knuth et al. 1977] and the KD-tree structure [Moore and Ostlund 2007]. To evaluate the accuracy and usability of our approach, we conducted comparative user studies to measure its search accuracy by comparing our approach with three the state of the art, example-based motion search algorithms [Kovar and Gleicher 2004; Liu et al. 2005; Forbes and Fiume 2005]. By analyzing user study results, we found that our approach is measurably effective in terms of search accuracy, and its search motion results are automatically ranked in an approximately perceptually-consistent order.

Our approach is flexible in terms of query input: a hybrid of multiple motion sequences, *e.g.*, the upper body part of a motion sequence from 1 to 500 frames and the lower body part of a second motion sequence from 1000 to 1500 frames, can be used as a novel query input. We are aware that a random combination of multiple human motion sequences may generate unnatural human motions. However, this function is still useful when users have small or limited query motions, because they can generate (or reorganize) novel query motions using various automatic motion fusing algorithms [Ikemoto and Forsyth 2004].

Certain limitations still exist in our approach. Current approach does not consider the path/motion trajectory of the root of the human in the retrieval algorithm. As such, given one query motion, its search results may enclose human motion sequences with completely different paths/trajectories, which might not be what users expect. For example, when the query motion is a sequence where a character is playing basketball while turning right, our approach may put a turn-left, basketball-playing motion at its top-ranked search result. In addition, most of internal parameters used in current approach are experimentally determined, which may not be optimal. In future work, we plan to perform in-depth analysis on the association between the used parameter values and its algorithm performances, in hope that it could help to resolve the optimized parameter setting for this motion retrieval approach.

Acknowledgments

This work is funded in part by the Texas Norman Hackerman Advanced Research Program (project number: 003652-0058-2007) and UH new faculty startup fund. We would like to thank Jingliang Peng for helpful discussion, and Xiaohan Ma for setting up the user study website.

References

AGRAWAL, R., FALOUTSOS, C., AND SWAMI, A. N. 1993. Efficient similarity search in sequence databases. In *Proc. of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)*, Springer Verlag, Chicago, Illinois, D. Lomet, Ed., 69–84.

ARIKAN, O., AND FORSYTH, D. A. 2002. Interactive motion generation from examples. *ACM Trans. Graph. 21*, 3, 483–490.

BADLER, N. I., AND BINDIGANAVALE, R. 1998. Motion abstraction and mapping with spatial constraints. In *Proceedings of International Workshop CAPTECH'98*, 70–82.

BARBIC, J., SAFONOVA, A., PAN, J.-Y., FALOUTSOS, C., HODGINS, J. K., AND POLLARD, N. S. 2004. Segmenting motion capture data into distinct behaviors. In *Proc. of Graphics Interface'04*, vol. 62, 185–194.

BEAUDOIN, P., COROS, S., VAN DE PANNE, M., AND POULIN, P. 2008. Motion-motif graphs. In *SCA'08: In Proceedings of Sympoisum on Computer Animation'08*.

BINDIGANAVALE, R. N. 2000. Building parameterized action representations from observagion. *Ph.D. Thesis, Department of Computer Science, University of Pennsylvania*.

BRAND, M., AND HERTZMANN, A. 2000. Style machines. In *Proc. of ACM SIGGRAPH '00*, 183–192.

CHAN, K., AND FU, A. W.-C. 1999. Efficient time series matching by wavelets. In *Proc. of ICDE'99*, 126–133.

CHIU, C. Y., CHAO, S. P., WU, M. Y., YANG, S. N., AND LIN, H. C. 2004. Content-based retrieval for human motion data. *Journal of Visual Communication and Image Representation 15*, 3, 446–466.

2007. CMU motion capture library. *http://mocap.cs.cmu.edu*.

DEHON, C., FILZMOSER, P., AND CROUX, C. 2000. Robust methods for canonical correlation analysis. In *Data Analysis, Classification, and Related Methods*, Springer-Verlag, Berlin, 321–326.

DEHON, C., FILZMOSER, P., AND CROUX., C. 2001. *The Elements of Statistical Learning:Data Mining, Inference,and Prediction*. Springer-Verlag, Berlin.

FALOUTSOS, C., RANGANATHAN, M., AND MANOLOPOULOS, Y. 1994. Fast subsequence matching in time-series databases. In *Proc.of ACM SIGMOD'94*, 419–429.

FOD, A., MATARIĆ, M. J., AND JENKINS, O. C. 2002. Automated derivation of primitives for movement classification. *Auton. Robots 12*, 1, 39–54.

FORBES, K., AND FIUME, E. 2005. An efficient search algorithm for motion data using weighted pca. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 67–76.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIĆ, Z. 2004. Style-based inverse kinematics. *ACM Trans. Graph. 23*, 3, 522–531.

GU, Q., PENG, J., AND DENG, Z. 2008. Compression of human motion capture data using motion pattern indexing. *Computer Graphics Forum* (accepted for publication).

HSU, E., SOMMER., AND POPOVIĆ, J. 2004. Example-based control of human motion. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 69–77.

IKEMOTO, L., AND FORSYTH, D. A. 2004. Enriching a motion collection by transplanting limbs. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 99–108.

KEOGH, E. J., PALPANAS, T., ZORDAN, V. B., GUNOPULOS, D., AND M.CARDLE. 2004. Indexing large human-motion databases. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, 780–791.

KIM, T.-H., PARK, S. I., AND SHIN, S. Y. 2003. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Trans. Graph. 22*, 3, 392–401.

KNUTH, D., MORRIS, J., AND PRATT, V. 1977. Fast pattern matching in strings. *SIAM Journal on Computing 6*, 2, 323–350.

KOVAR, L., AND GLEICHER, M. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph. 23*, 3, 559–568.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Trans. Graph. 21*, 3, 473–482.

LEE, J., CHAI, J. X., REITSMA, P., HODGINS, J. K., AND POLLARD, N. 2002. Interactive control of avatars animated with human motion data. *ACM Trans. Graph. 21*, 3, 491 – 500.

LEE, K. H., CHOI, M. G., AND LEE, J. 2006. Motion patches: building blocks for virtual environments annotated with motion data. *ACM Trans. Graph. 25*, 3, 898–906.

LI, Y., WANG, T., AND SHUM, H.-Y. 2002. Motion texture: a two-level statistical model for character motion synthesis. *ACM Trans. Graph. 21*, 3, 465–472.

LI, C., KULKARNI, P. R., AND PRABHAKARAN, B. 2007. Segmentation and recognition of motion capture data stream by classification. *Multimedia Tools Appl. 35*, 1, 55–70.

LIN, Y. 2006. Efficient human motion retrieval in large databases. In *Proc. of Int'l Conf. on Computer graphics and interactive techniques in Australasia and Southeast Asia*, 31–37.

LIU, F., ZHUANG, Y., WU, F., AND PAN, Y. 2003. 3D motion retrieval with motion index tree. *Computer Vision and Image Understanding 92* (Nov/Dec), 265–284.

LIU, G., ZHANG, J., WANG, W., AND MCMILLAN, L. 2005. A system for analyzing and indexing human-motion databases. In *Proc. of ACM SIGMOD '05*, 924–926.

MOORE, A., AND OSTLUND, J. 2007. Simple kd-tree library. *http://www.autonlab.org/*.

MÜLLER, M., AND RÖDER, T. 2006. Motion templates for automatic classification and retrieval of motion capture data. In *In SCA '06*, 137–146.

MÜLLER, M., RÖDER, T., AND CLAUSEN, M. 2005. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph. 24*, 3, 677–685.

SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3, 514–521.

SO, C. K. F., AND BACIU, G. 2006. Hypercube sweeping algorithm for subsequence motion matching in large motion databases. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, ACM, New York, NY, USA, 221–228.

SOK, K. W., KIM, M., AND LEE, J. 2007. Simulating biped behaviors from human motion data. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*.

YU, Q., LI, Q., , AND DENG, Z. 2007. Online motion capture marker labeling for multiple interacting articulated targets. *Computer Graphics Forum (Proceedings of Eurographics 2007)*, 477–483.

ZHAO, L. 2001. Synthesis and acquisition of laban movement analysis qualitative parameters for communicative gestures. *Ph.D. Thesis, Department of Computer Science, University of Pennsylvania*.