Efficient Indexing of 3D Human Motions

Petra Budikova
Masaryk University
Brno, Czech Republic
budikova@fi.muni.cz

Jan Sedmidubsky
Masaryk University
Brno, Czech Republic
xsedmid@fi.muni.cz

Pavel Zezula
Masaryk University
Brno, Czech Republic
zezula@fi.muni.cz

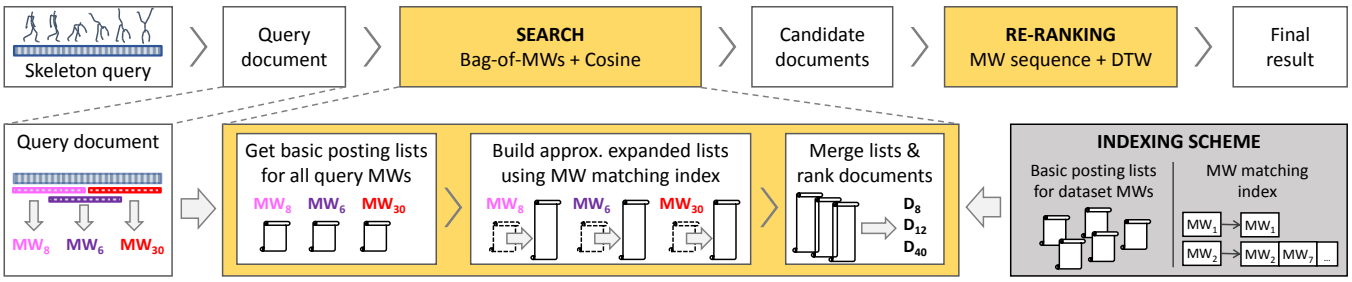

Figure 1: Efficient skeleton-data retrieval. In a pre-processing phase, skeleton sequences are transformed into motion documents, i.e., compact text-like representations composed of structured motion words (MWs). The motion documents are organized using a new indexing scheme that extends the traditional inverted files. During query processing, candidate documents are efficiently retrieved by a proposed approximate search algorithm and finally re-ranked using the DTW alignment.

ABSTRACT

Digitization of human motion using 2D or 3D skeleton representations offers exciting possibilities for many applications but, at the same time, requires scalable content-based retrieval techniques to make such data reusable. Although a lot of research effort focuses on extracting content-preserving motion features, there is a lack of techniques that support efficient similarity search on a large scale. In this paper, we introduce a new indexing scheme for organizing large collections of spatio-temporal skeleton sequences. Specifically, we apply the motion-word concept to transform skeleton sequences into structured text-like motion documents, and index such documents using an extended inverted-file approach. Over this index, we design a new similarity search algorithm that exploits the properties of the motion-word representation and provides efficient retrieval with a variable level of approximation, possibly reaching constant search costs disregarding the collection size. Experimental results confirm the usefulness of the proposed approach.

CCS CONCEPTS

• **Information systems** → **Data structures**; **Multimedia databases**; **Information retrieval query processing**.

ICMR '21, August 21–24, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463646

KEYWORDS

human motion data, skeleton sequences, motion word, text-based processing, indexing, extended inverted files, ranked retrieval, approximate searching, scalability

ACM Reference Format:
Petra Budikova, Jan Sedmidubsky, and Pavel Zezula. 2021. Efficient Indexing of 3D Human Motions. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463646

1 INTRODUCTION

Human motion can be described by a sequence of skeleton poses, where each pose keeps 2D/3D coordinates of important body joints in a specific time moment. Such spatio-temporal skeleton data have enormous application potential in many domains, e.g., in sports to automatically assess a figure-skating performance or detect fouls during a football game without emotions of human referees; in healthcare to remotely evaluate the progress in rehabilitation exercising or to discover movement disorders as indicators for choosing suitable treatments; in security to detect potential threats like a running group of people; or in computer animation to find previously-captured animations relevant for building a new movie scene. Until recently, specialized hardware technologies were required to record the 3D positions of the moving body, so the amount of digital motion data was fairly limited. However, new pose-estimation software tools [1, 6] allow to extract skeleton data (2D or 3D) from ordinary videos. As a result, we expect an explosion of skeleton data in the near future, which requires content-based processing techniques that perform efficiently on a large scale.

Current research mainly focuses on recognizing classes of pre-segmented actions [5, 10, 12], detecting actions in a stream [15, 23], or searching for query-relevant subsequences within a long motion [2, 22]. These tasks often employ *query-by-example retrieval*

as the underlying operation; e.g., in the subsequence search task, a long motion is usually partitioned into a large number of short motion segments that need to be effectively and efficiently matched against a user query. However, these two objectives—efficiency and effectiveness—are difficult to achieve at the same time. Many existing retrieval techniques [2, 18, 19, 24] focus solely on search quality and do not discuss the efficiency at all, which leads to expensive sequential scan over the whole dataset. The efficiency-oriented works either propose very compact features that allow fast sequential scanning [12, 13], or utilize various indexing schemes to organize the motion data (e.g., the binary tree [25], kd tree [9], R* tree [4], inverted file index [14], or tries [8]). To optimize the efficiency-effectiveness trade-off, a two-phase retrieval model is often used, where the candidate objects identified within an efficient *search* phase are submitted to a *re-ranking* phase that refines the result using more expensive techniques (e.g., traversal of a graph structure [9] or ranking by the Dynamic Time Warping [14, 20]). A more thorough discussion and comparison of all these methods can be found in the recent survey [21]. However, even the index-based approaches [4, 8, 9, 14, 25] are designed to operate on collections of only thousands, or maximally dozens of thousands of motions [22], and their application to large-scale collections is disputable.

In this paper, we propose a new retrieval approach that can efficiently process a very large collection of spatio-temporal motions and enables controlling the retrieval costs by a user-specified parameter. In particular, we transform the complex 3D skeleton sequences into text-like documents that are composed of so-called motion words. The motion words are internally structured, which allows them to preserve the motion content but does not enable straightforward application of standard text-retrieval techniques. To support efficient searching over such motion documents, we propose a new indexing scheme and a new retrieval algorithm that is able to gradually provide the most promising candidate results with respect to a query example. This allows us to apply approximate searching with high effectiveness and near-constant processing times, which should scale well even to very large data collections. A high-level overview of the whole concept is provided in Figure 1.

2 PROBLEM ANALYSIS

To facilitate large-scale motion retrieval, it is necessary to find a well-balanced combination of compact yet descriptive motion features on one hand, and efficient and scalable index structure on the other hand. Our solution builds upon the recently proposed idea of text-based motion processing [20], which is a promising but little researched direction. In this section, we discuss the overall architecture and the open challenges of text-based motion processing, and provide the preliminary information needed for describing the proposed indexing scheme. In particular, we describe the transformation of skeleton sequences into sequences of structured motion words, introduce our experimental dataset along with the evaluation methodology, and present the retrieval results of several baseline sequential-scan algorithms.

2.1 Text-like Motion Processing

Transformation of unstructured data into a text-like representation is generally a promising approach that has been successfully applied in other domains, e.g., in image retrieval [16]. The obvious benefit of such approach is the possibility to utilize mature text-processing methods for efficient retrieval. However, for the complex spatio-temporal skeleton data, such transformation is far from straightforward. To apply the text-based retrieval approach for skeleton data, the following two challenges need to be addressed:

- transformation of high-dimensional skeleton sequences into a text-like representation; and
- adaptation of text-processing techniques to the specific needs of the motion-text.

The task of finding content-preserving text-like representations of skeleton data is discussed in [20], resulting in the proposal of two variants of *motion words* (MWs): (i) one-dimensional *hard MWs*, which are compared by simple equality, and (ii) more complex *soft MWs*, which are compared by a non-trivial matching function that allows different MWs to be considered mutually relevant. The soft MWs demonstrate much better ability to preserve the motion content, but the non-trivial MW matching makes their processing more computationally demanding.

In this paper, we focus on the second challenge of text-like motion processing, i.e., adapting text-retrieval techniques to allow efficient searching in the motion-text. To the best of our knowledge, this topic has not been studied yet. Our specific objective is to find an indexing and retrieval strategy for motions transformed into documents of soft MWs. Such documents are artificial constructs that have no clear semantics, therefore their processing involves several challenges that are not present in standard text retrieval. First, the non-trivial soft-MW matching needs to be respected within query processing, which requires a huge expansion of query terms, much larger than expansions used in text retrieval. Second, the motion queries are significantly longer than typical text queries, which further increases the processing costs. Third, the text-retrieval uses different weighting schemes that improve search efficiency and allow optimizations of search costs using approximate searching, but these weighing schemes may not work with the motion data. Due to these factors, a simple application of text-based processing would be neither efficient nor scalable, therefore it is necessary to find new ways of implementing the text-retrieval principles. At the same time, the artificial motion words also offer some opportunities that are not present in text processing: the MWs have a regular structure that can be exploited during indexing, and the size (granularity) of the motion vocabulary can be adjusted so that it works well with a chosen indexing structure.

2.2 Problem Statement

A *skeleton sequence* represents human motion as a discrete sequence of poses, where each pose keeps the 3D coordinates of important body joints in a specific time moment (i.e., frame). Our objective is to efficiently search a large collection of skeleton sequences and retrieve those that are the most similar to a given query sequence.

As a first step towards efficient large-scale retrieval, we transform the complex spatio-temporal skeleton data into a compact text-like representation of structured soft motion words. We denote such representation as a *motion document*. Given a *collection* $\mathcal{D} = \{D_1, \ldots, D_r\}$ of *data documents* $D_i = (d_1, \ldots, d_m)$, we aim to efficiently retrieve the $k \in \mathbb{N}$ most similar data documents with

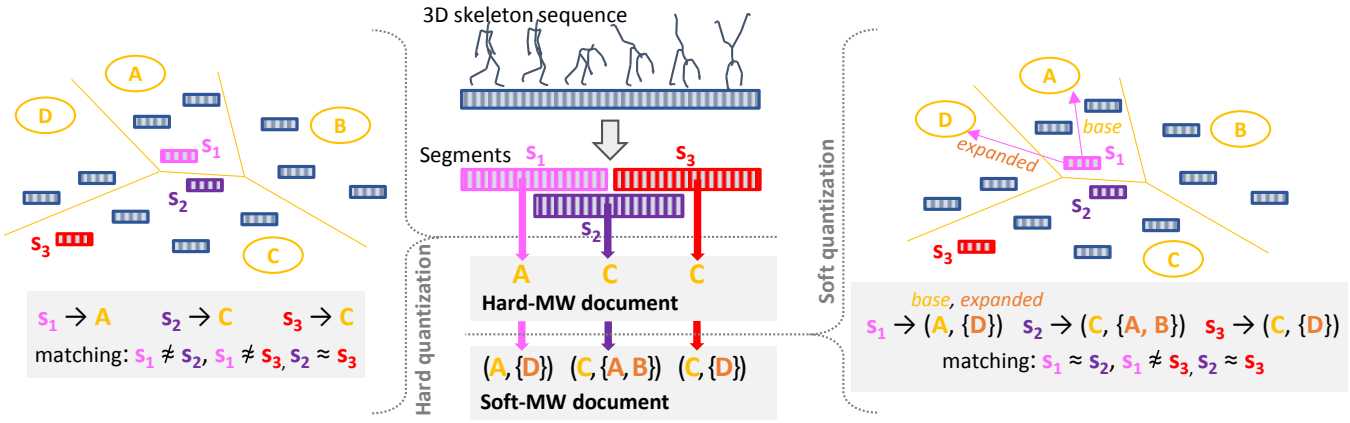

Figure 2: Transformation of 3D skeleton sequence into the hard-MW and soft-MW document. In the hard quantization, similar segments s_1 and s_2 do not match (the border problem appears), while in the soft quantization s_1 and s_2 do match.

respect to a *query document* $Q = (q_1, \ldots, q_n)$, where q_i and d_j correspond to individual soft motion words. This requires designing a suitable indexing structure and a scalable retrieval algorithm that can identify relevant documents while accessing only a fraction of the data collection.

2.3 Construction of Motion Documents

The transformation of a skeleton sequence into a motion document consists of the following three steps [20]: (i) each input skeleton sequence is cut into a sequence of short and overlapping segments; (ii) a similarity-based partitioning of the segment space is found; and (iii) each segment is replaced by a motion word, which characterizes the segment's position within the segment space using identifiers of near partitions. Noticeably, the whole transformation is completely unsupervised, which makes it widely applicable.

The motion words can be constructed in several ways, differing in their internal structure and the definition of a *MW matching function*. This Boolean-valued function determines whether two MWs are sufficiently similar to be considered mutually relevant in the course of motion retrieval: $match^{MW} : MW \times MW \rightarrow \{0, 1\}$. The objective of the MW approach is to provide a *similarity-preserving* transformation from segments to MWs, so that similar segment pairs are with a high probability mapped to matching MWs and dissimilar segment pairs to non-matching MWs.

Hard MWs. The hard motion words are the simplest implementation of the MW principle. Each hard MW is formed by a single identifier of the partition where the respective segment lies (see Figure 2-left). The *hard MW matching function* is trivial – it returns 1 only if the two MWs are identical. Using the hard MWs, it is possible to transform a skeleton sequence into a sequence of one-dimensional identifiers that can be readily processed by standard text-retrieval techniques. However, the hard quantization of the segment space suffers quite significantly from a so-called *border problem*: segments that are close in the segment space may be assigned to different partitions and thus considered non-matching in the MW space (segments s_1 and s_2 in Figure 2-left). This negatively affects the quality of MW-based similarity searching.

Soft MWs. To reduce the border problem, the soft MWs keep more information about the position of each segment in the segment space, which allows them to better preserve the segment-similarity relationships. The soft MW $q = (x, X)$ for segment s is composed of multiple *MW elements* that identify the partitions of the segment space that are in a close neighborhood of s: the *base element* x identifies the partition where s belongs, and $X = \{x_1, \ldots, x_n\}$ is the set of *expansion elements* corresponding to other partitions that are near to s (see Figure 2-right). The set X is bounded by two parameters: the maximum number of the closest partitions to be included, and the maximum distance of partition x_i from s. The *soft MW matching function* is defined as follows: $q_i = (x, X)$ and $d_j = (y, Y)$ are matching if the base elements x and y are equal, or x appears among the elements in Y, or y appears among the elements in X. As depicted in Figure 2-right, the soft-MW matching allows to overcome the border problem in case of segments s_1 and s_2.

The similarity between two MW documents of variable lengths is measured in [20] by the Dynamic Time Warping (DTW), where the hard-/soft-MW matching function is used inside the DTW.

2.4 Dataset and Evaluation Methodology

Throughout this paper, we verify the usefulness of the proposed technologies in experiments, using a prototype implementation and real-world motion data. Our experimental data come from the PKU-MMD dataset [11], which provides nearly 20 K single-subject skeleton sequences captured by the Microsoft Kinect technology at a 30 frames-per-second rate and a body model consisting of 25 joints. The sequences are categorized into 43 classes of daily activities (e.g., "drink", "wave hand", "stand-up") and significantly vary in length, having 118 frames (≈ 4 s) on average with the maximum of 759 frames (≈ 25 s). The total collection size is more than 20 hours, which makes it one of the largest research motion datasets.

Following the methodology provided in Section 2.3 (and described in [20] in more detail), we transformed each skeleton sequence into a sequence of soft MWs – the motion document. Specifically, we cut each skeleton sequence synthetically into a series of overlapping segments. Analogous to [2, 20], we fixed the segment

length to 20 frames and the segment overlap to 16 frames, so the segments are shifted by 4 frames. In total, this generated nearly 500 K segments whose space was quantized into 1,500 clusters. Each soft MW consists of up to 20 MW elements (i.e., identifiers of clusters), which resulted in the total number of 372 K unique soft MWs identified in all the dataset documents. The average size of a single motion document is 25 soft MWs.

We study the effectiveness and efficiency of different retrieval methods in the context of a 20-nearest-neighbor search evaluated over 129 query documents (for each out of 43 document classes, three different instances are randomly selected). We quantify effectiveness as the average *precision* of the 129 queries, where the query precision is computed as a ratio of correctly retrieved data documents. A document is considered as correctly identified if it belongs to the same class as the query document. We measure efficiency as the average time (in milliseconds) needed to evaluate a single query on the collection of 20 K data documents.

2.5 Baseline Evaluation

To put our results into perspective, we evaluate three baseline approaches: (i) a sequential scan with the DTW alignment over the original 3D skeleton data, and sequential scans with DTW applied to the documents of either (ii) hard, or (iii) soft MWs. The average precision and costs of all the three baselines are provided in Table 2 (rows 1–3). We can observe that the soft MW variant clearly achieves the best precision but requires higher search costs compared to the hard MW variant.

It should be noted that the DTW distance implementation used in our experiments is very basic. We are aware of the more advanced techniques that can be used to speed-up the DTW computation and significantly decrease the query processing costs [17]. However, even with the enhancements the sequential-scan approach is not scalable to very large datasets.

3 INDEXING SOFT-MW DOCUMENTS

The motion documents provide a compact skeleton-data representation that significantly reduces the data size and also decreases query processing costs by several orders of magnitude, even when the costly DTW distance is used for comparison. However, to support scalable data management, the MW representation has to be combined with efficient indexing and search strategies which have not been proposed yet. In this section, we gradually build a scalable indexing scheme suitable for soft-MW documents.

3.1 Applicability of the Bag-of-Words Model

A basic strategy in the text-search domain is to treat text documents as bags of words, represent them by (typically weighted) vectors, and apply the Cosine measure to compute the similarity between a query vector and the data vectors [3]. The top-ranking documents are directly presented to users or further examined using additional measures of document relevance. For motion documents, we assume that the two-phase retrieval will be necessary, combining fast identification of promising candidates with more thorough similarity evaluations in the re-ranking phase. Our first objective is to check whether the bag-of-words model and the Cosine measure are sufficient for identifying relevant candidate motion documents.

Table 1: 20NN search precision and processing costs of different retrieval schemes over soft MWs. Both candidate search and re-ranking are evaluated by sequential scanning of the dataset/candidate set.

Candidate retrieval	# of cand.	Re-rank	Precision	Costs [ms]	
				Cand. r.	Re-rank
DTW	20	–	60.60 %	648	–
Cosine	20	–	37.09 %	430	–
Cosine	50	DTW	55.08 %	430	3
Cosine	100	DTW	59.61 %	430	6
Cosine	150	DTW	59.92 %	430	8
Cosine	200	DTW	60.04 %	430	10
Cosine	300	DTW	60.16 %	430	15

The soft-MW documents can be straightforwardly represented by bags-of-words, but the Cosine computation needs to be adjusted to provide a meaningful measure of similarity between two soft-MW bags. The standard Cosine score of a document D with respect to query Q is computed as the (weighted and normalized) number of words that appear in both D and Q. However, the soft-MW representation works with the concept of matching, where two non-identical MWs can be considered mutually relevant. Clearly, the MW-matching needs to be incorporated into the Cosine similarity evaluation. In an ideal case, the Cosine similarity of soft-MW Q and D should take into account the number of words from $Q = (q_1, \ldots, q_n)$ that have some match in $D = (d_1, \ldots, d_m)$, and the number of words from D having match in Q (a single q_i can be matched by multiple different MWs in D, and a single d_i may match multiple different MWs in Q). However, the evaluation of matches is computationally expensive, and we also need to take into account the indexability of the Cosine computation. As discussed later, the number of matched query MWs can be determined using a specific type of query expansion over the inverted file index, whereas the computation of matches for all document MWs would be problematic. Therefore, we only consider the number of matched query MWs in the soft-MW Cosine similarity.

Following the best-practices of text retrieval, we also consider the possibility of applying frequency-based weighting to determine the importance of individual words for the Cosine computation. Standard IR typically uses a combination of the *text frequency (TF)* and *inverted document frequency (IDF)* of individual words. Accordingly, we attempt to utilize the same principles adjusted to the needs of soft-MW matching. In particular, we compute TF as usual, but the IDF of a MW d_i is computed as the ratio of documents that contain any MW that matches d_i.

Experimental evaluation. We first evaluate the precision and costs of a single-phase Cosine-based retrieval, then we combine it with the DTW alignment in a two-phase search model with different sizes of the candidate set. Table 1 compares the achieved results to the baseline DTW alignment on soft MWs. We can observe that the result of a simple Cosine-based retrieval is far worse than the baseline, but the two-phase search model with 100 or more candidate objects achieves a satisfactory precision. The extra costs for the refinement phase are negligible if a reasonably small candidate

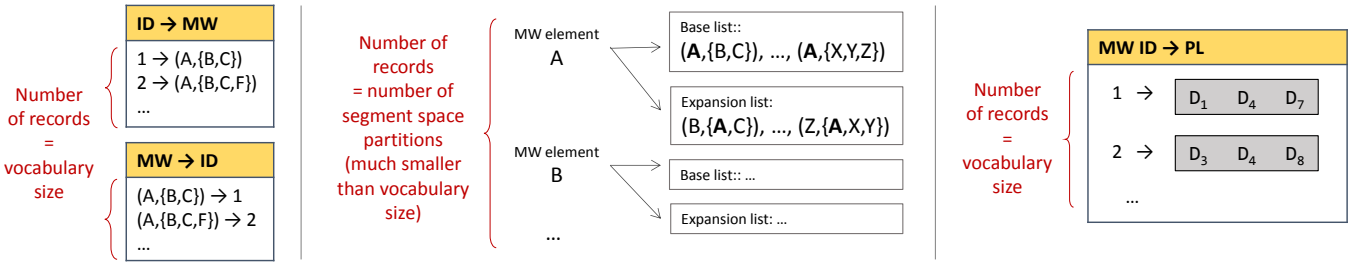

Figure 3: Components of the index: MW-ID translation tables (left), MW matching index (middle), and basic PLs (right).

set is used, but the improvement in result quality is significant. Noticeably, the Cosine-based candidate retrieval is rather slow; this is caused by the need to search for matching pairs of MWs between the query and a candidate document.

Regarding document and query weighting, our experiments show that the best results are achieved without any weights. In this aspect, the motion documents are markedly different from text documents, for which the TF-IDF weighting is beneficial. We have identified two types of reasons for this behavior: the general properties of the motion-text, and the properties of our particular dataset. The motion-text has the following specifics: (i) the motion words correspond to artificially cut motion segments that have no direct semantic meaning; (ii) the MW frequencies only roughly correspond to segment frequencies due to quantization; and (iii) MWs representing e.g. a simple standing can be repeated many times in a row, which does not happen with text words. In addition to this, our dataset is composed of short documents, each of which describes one semantic action. In such dataset, we expect a low occurrence of stopwords and a low number of MW repetitions within a single document, as opposed to the text databases for which the frequency-based weighting was developed. Altogether, we conclude that it is not possible to directly adopt TF-IDF weighting from text retrieval, and some further research into the possibilities of MW weighting might be interesting. In the rest of this work, we focus on the efficient retrieval of unweighted bag-of-words.

3.2 Extended Inverted-File Index

The Cosine-based retrieval shows promising potential for identifying candidate motion documents, so our next objective is to design an efficient indexing scheme for fast evaluation of Cosine queries over soft MWs. In text retrieval, the Cosine search is typically computed over the *inverted file index*, which provides for each vocabulary term t a *posting list* $PL(t) = (D_1, \ldots, D_n)$ of all documents that contain the term t. Similarly, we can define $PL(q)$ as a list of all motion documents that contain the exact soft MW q. However, the Cosine-based similarity query over soft MWs cannot be directly answered by merging the posting lists $PL(q_i)$ of all query words $q_1, \ldots, q_n$, because this wouldn't take into account the soft MW-matching function. Let's consider again the illustration of Figure 2: motion words $(A, \{D\})$ and $(C, \{A, B\})$ represent highly similar segments and are considered to be matching, so for a query containing the word $(A, \{D\})$, documents with $(A, \{D\})$ as well as $(C, \{A, B\})$ should be checked. However, a standard PL for $(A, \{D\})$ would only contain documents where $(A, \{D\})$ occurs.

To deal with this problem, we extend the standard inverted-file approach by introducing an *expanded posting list* $PL^+(q)$, which consists of the IDs of all data documents that contain q or any MW that matches q. Merging such expanded posting lists for all query MWs now provides the correct Cosine scores for all motion documents with non-zero similarity to Q, but the management of the expanded lists is more complicated. Let $matchingMWs(q)$ be the set of all MWs that match a given MW q. The expanded list $PL^+(q)$ can either be created during data indexing, or constructed during query evaluation by merging $PL(r)$ of all $r \in matchingMWs(q)$. The first solution would not require any changes of the query processing, but the expanded lists PL^+ would occupy much more memory and their updates would be rather expensive. Therefore, we choose the second option, where only the relatively short basic posting lists are stored. This is significantly less demanding in terms of memory occupation, but induces additional computational costs during retrieval when the basic posting lists are merged into the expanded lists. However, we are able to upper-bound these costs by using approximate searching, as discussed in the following section.

The complete architecture of our index is illustrated in Figure 3. Each MW from the dataset vocabulary is assigned an ID, and two hash tables are kept for translating the MWs to IDs and back. The vocabulary items are further organized in a MW matching index to support fast identification of $matchingMWs(q)$. In particular, two lists are maintained for each MW element e: (i) a *base list* of all MWs where e is the base element, and (ii) an *expansion list* of all MWs where e is among the expansion elements. The matches of MW $q = (A, \{B, C\})$ are then found by uniting the expansion list for A with the base lists for B and C. Finally, for each MW we keep the basic posting list of documents that contain that exact MW.

For our dataset of 20K data documents and 372K soft MWs, the sizes of the index structures are the following: the vocabulary translation tables need 50 MB, the MW matching index requires 30 MB, and the posting lists occupy 4MB of memory. For comparison, the original skeleton data size is 1.5 GB. The space needed for vocabulary manipulations seems rather large, but it is important to realize that we are using a very large vocabulary that should be sufficient for much larger data. When the dataset grows, the only growing memory structure would be the very compact posting lists.

Experimental evaluation. In rows 4 and 5 of Table 2, we can see how the processing costs of the Cosine-based retrieval change when the indexing is applied. To better understand the complexity of our index-based processing, we also decompose the costs into individual query processing phases. The candidate retrieval costs decrease

from 430 ms to 317 ms. The most expensive part of the index-based searching is the construction of the expanded lists $PL^+(q)$, which requires 276 ms on average. This is caused by the need to combine a high number of short basic PLs; in particular, each of our MWs has 12,436 matches on average, so that many PLs have to be merged.

3.3 Scalable Processing of MW Posting Lists

The extended inverted file index allows us to skip documents with zero overlap with the query, but the above-described construction and merging of the expanded lists PL^+ does not scale well to large datasets. We can observe in the experimental results that even for our rather small dataset with short basic PLs, the construction of expanded lists PL^+ is a performance bottleneck due to the high number of matching MWs that need to be considered for each query. When the dataset grows and the vocabulary remains unchanged, the basic posting lists are bound to grow as well, further increasing the processing time of PL^+ construction and also PL^+ merging. Alternatively, we can design the MW vocabulary in such way that it grows with the database size, gradually adding more MW expansion elements to the soft MWs. This would keep the basic PLs small but even more PLs would need to be merged for each query MW. The key to scalable retrieval thus lies in limiting both the number and size of PLs that need to be processed.

In this section, we design an approximate retrieval algorithm that limits the number of processed PL records without significantly worsening the result quality. The algorithm is based on a new non-binary similarity measure over the soft MWs, which can be used to order the matching MWs of each query MW and limit the number of PLs used for construction of each PL^+. We also show that this new MW-similarity measure can be used for weighting documents in the expanded posting lists, which increases the quality of the Cosine search results.

Non-binary similarity of soft MWs. Let us remember that each soft MW represents a short segment of the original skeleton sequence, and the structure of the soft MW reflects the position of the respective segment in the segment space. In particular, the base element references the partition where the segment lies, and the expansion elements list other partitions that are sufficiently near to the segment. This information can be used to define a more fine-grained measure of MW similarity than the binary matching function $match^{MW}$ defined in [20]. Intuitively, two segments that lie very close to each other are likely to have many of the same close partitions, whereas segments that are further apart will have different sets of close partitions. Therefore, we propose to combine the original MW matching function with the Jaccard similarity over the sets of all elements of the two MWs. More precisely, let $q = (x, X)$ and $r = (y, Y)$ be the two soft MWs, and let $X' = X \cup \{x\}$ and $Y' = Y \cup \{y\}$ be the sets of all elements of q and r, respectively. Then, we quantify the similarity between q and r using the following *sim* function:

$$sim(q,r) = \begin{cases} \dfrac{|X' \cap Y'|}{|X' \cup Y'|} & \text{if } match^{MW}(q,r) = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

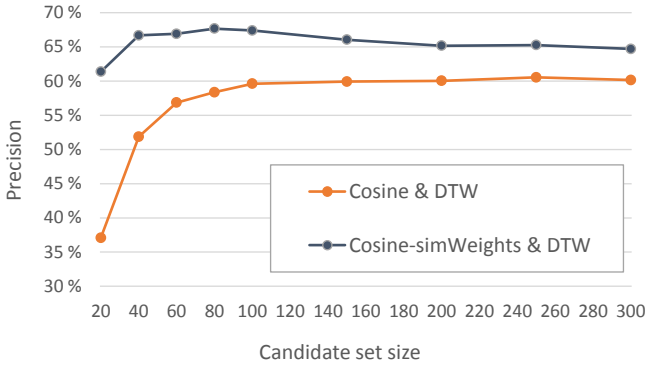

Figure 4: Influence of the *sim* function on the precision of two-phase searching.

For two non-matching MWs, the *sim* function still returns the constant 0, because the non-matching words are not interesting for retrieval; however, the similarity of matching MWs is changed from 1 to the interval (0;1], which allows us to distinguish between closer and less close matches of a given MW.

To test the semantic usefulness of the *sim* function, we apply it to determine the weights of individual documents in the expanded lists PL^+ (see Figure 5 for illustration). Let $Q = (q_1, \ldots, q_n)$ be the query and D a data document that appears in $PL^+(q_i)$ for some $i \in \{1, \ldots, n\}$. Then, D has to appear in one or more basic lists $PL(r_1), \ldots, PL(r_k)$, where $r_1, \ldots, r_k \in matchingMWs(q_i)$. We take $max(sim(q_i, r_1), \ldots, sim(q_i, r_k))$ as the weight of D in $PL^+(q_i)$, and compute the overall score of D with respect to Q as a sum of weights assigned to D in $PL^+(q_1), \ldots, PL^+(q_n)$.

The effect of MW-similarity weighting on the quality of the two-phase retrieval is shown in Figure 4. We can see that the precision of the Cosine-based candidate selection is significantly improved, especially for small candidate sets. It is also interesting that the overall precision does not gradually grow with the candidate set sizes but achieves its maximum for candidate set size of 100 items. This is caused by the fact that the DTW distance of many of the candidates is the same and the ties are broken arbitrarily.

Approximate searching with reduced expanded lists. The *sim* function improves the search quality, which is a nice bonus, but the main motivation for its construction was to reduce the size of expanded lists PL^+ so that they can be efficiently processed. For each query word q_i, we thus order its matching words by their descending similarity, and build $PL^+(q_i)$ using only a limited number of the best matches. There are several ways of selecting the matches to be used: (i) by introducing a minimum similarity score of matches that will be considered for building the PL^+; (ii) by limiting the number of the most similar matches to be used; (iii) by limiting the size of the PL^+ and using as many top-ranked MW matches as necessary to achieve this size. The last option is the most suitable for large-scale searching: it ensures that some minimum number of documents is considered for each query MW, and it guarantees a constant PL^+ processing time.

As illustrated in Figure 5, the same set of basic PLs can be used to build a variety of expanded lists PL^+ during the query processing,

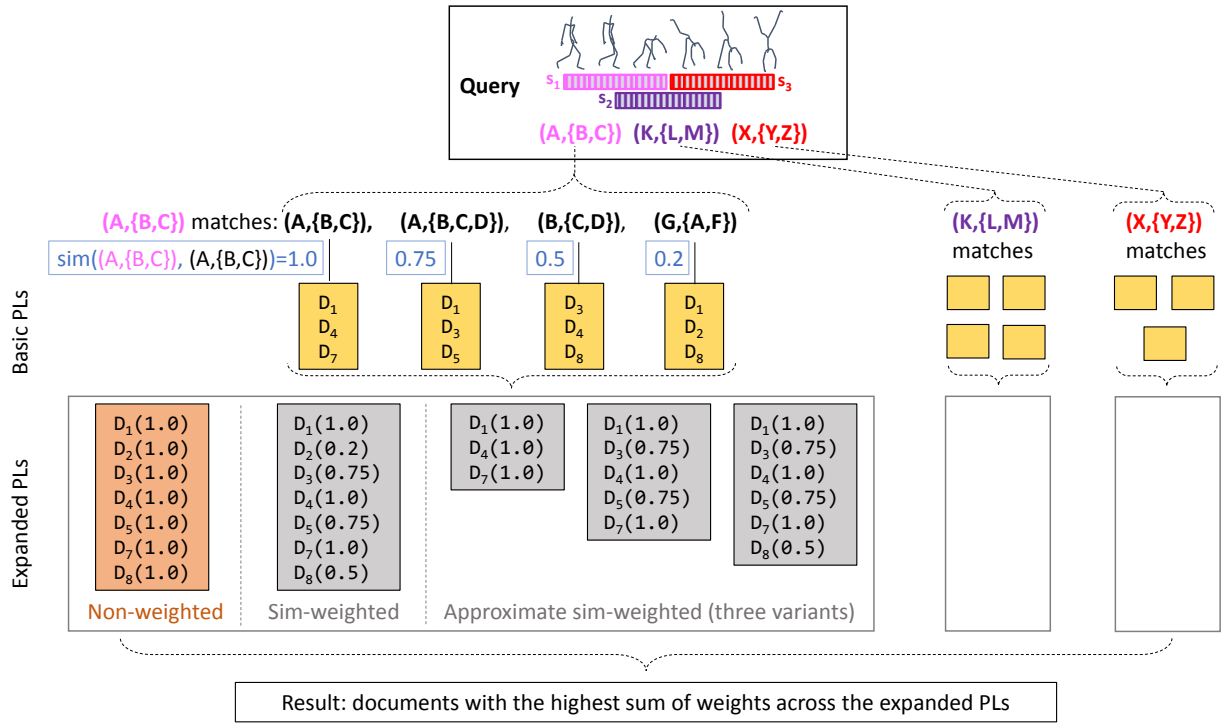

Figure 5: Construction of PL^+ with different weights and approximations.

depending on the preferences of a given application or user. The size limit for PL^+ can be set according to the requirements on retrieval precision and costs. Data collections of arbitrary sizes can then be searched in near-constant time – the time needed for identification of matching words and their ordering is logarithmic to the size of vocabulary, which may grow with the collection size but at a slower pace. In case the collection grows so much that even the basic PLs become larger than the given size limit, the index has to be re-build using soft MWs with a higher number of expanded MW elements.

Experimental evaluation. Table 2 provides an overview of the retrieval quality and costs of all the studied search approaches. It is worth noticing that the precision of the *sim*-weighted Cosine search with the DTW refinement overcomes even the baseline precision of the DTW-based alignment of soft-MW documents over the whole collection. Actually, the Cosine and DTW similarity are based on opposing principles: the Cosine score reflects the number of matches between two motion documents, whereas the DTW distance highlights the differences between the two sequences. It appears that a suitably weighted combination of these two principles performs better than any one of them individually.

The last three rows of Table 2 analyze the ability of the *sim* function to correctly identify the most relevant posting lists. In particular, we compare the results of Cosine retrieval with fully expanded posting lists (with the average size of 2,251 items) to approximate retrieval with the PL^+ size limited to 1,000, 500, and 100 top-ranked candidates. The results confirm that the result quality remains high even in the most restricted setting when less than 5 % of the PL^+ is used.

In terms of processing costs, the *sim*-weighted Cosine distance is more expensive to compute than the standard Cosine. We need to translate MW IDs into the actual motion words and compute their similarities, which is rather expensive when tens of thousands of matching MWs are processed. However, this part of the processing costs is fixed for a given vocabulary and does not grow with an increasing size of the dataset. Furthermore, we can see that the additional costs of *sim* computation are outweighed by the reduced processing times of the approximate searching even for our small dataset. It is also important to emphasize that the processing times are measured using our own, non-optimized prototype implementation, which does not employ any existing text-search engine. Moreover, all the computations are performed on one CPU only. Therefore, the actual costs are quite high even for the 20 K test dataset. Still, we can observe the important decreasing trend in processing costs when the approximate Cosine computation is used. In future, we plan to further decrease the query evaluation costs by employing parallel processing for creating and merging the expanded posting lists.

4 DISCUSSION

Large-scale searching in motion data is a challenging task with many open areas of research, which can be approached from diverse directions. For a rough categorization of existing works, let us separate the two principal subtasks of motion retrieval: (i) feature extraction and metric learning, and (ii) organizing and searching the features. For feature extraction, we need to separate the two orthogonal approaches of supervised or unsupervised learning; the

Table 2: 20NN search precision and processing costs using different retrieval methods. Apart from Baseline 1 and Baseline 2, all methods work with documents represented by soft MWs.

Candidate retrieval method	Re-rank	Precision	Time [ms]					
			find MW matches	compute *sim*	construct PL^+	merge PL^+	re-rank candid.	overall costs
Baseline 1: DTW on skeletons	–	54.19 %	*sequential scan*				–	92,589
Baseline 2: DTW on hard MWs	–	51.82 %	*sequential scan*				–	415
Baseline 3: DTW on soft MWs	–	60.60 %	*sequential scan*				–	648
Cosine100	DTW	59.61 %	*sequential scan*				6	436
Cosine100	DTW	59.61 %	*18*	–	*276*	*23*	*6*	323
Cosine100-simWeights	DTW	67.40 %	*18*	*114*	*276*	*23*	*6*	437
Cosine100-simWeights, $\|PL^+(q)\| \leq 1,000$	DTW	66.94 %	*18*	*114*	*173*	*14*	*6*	325
Cosine100-simWeights, $\|PL^+(q)\| \leq 500$	DTW	66.16 %	*18*	*114*	*96*	*9*	*6*	243
Cosine100-simWeights, $\|PL^+(q)\| \leq 100$	DTW	64.34 %	*18*	*114*	*49*	*3*	*6*	190

supervised methods generally achieve higher precision, but require labeled training data that limit their applicability. The feature organization can have different forms and properties, ranging from fast linear scan of compact features to sublinear index-based retrieval.

In our research, we aim at widely-applicable large-scale motion retrieval. Therefore, our approach is based on features prepared in an unsupervised way, and organizes them in an index structure that allows efficient and scalable retrieval with near-constant time complexity. To the best of our knowledge, such approach to motion retrieval has not been proposed before. There are several works that study unsupervised feature extraction [2, 20], but these do not attempt to index the data. On the other hand, those works that study efficient motion retrieval use supervised features [7–9, 12, 22, 25]. Our approach is therefore unique and innovative, but this makes it difficult to compare our results to state-of-the-art motion retrieval methods in terms of efficiency and effectiveness.

We deal with this situation in the following way. We compare our approach to state-of-the-art research in those aspects that can be fairly evaluated, i.e., we compare the efficiency and scalability of our approach to other indexing-based methods, and discuss the precision of our approach in the context of other unsupervised approaches. Furthermore, we make our selected queries publicly available[1] to allow other researchers to perform direct comparisons with the proposed approach.

In terms of efficiency, supervised-feature-indexing systems report the average search times of tens [7, 8, 14, 22] or hundreds [9, 25] of milliseconds per query on data collections that are equal or smaller in magnitude than ours. Our non-optimized single-thread implementation achieves the query response times in order of hundreds of milliseconds as well. Considering scalability, the indexing structures of state-of-the-art methods promise a linear [14] or sub-linear [7, 8, 22] growth of query processing times for growing datasets. However, the sub-linear solutions typically use high-dimensional feature space indexing [9, 22], which is known to be problematic on large scales due to the curse of dimensionality. On the other hand, our index utilizes compact low-dimensional motion features, can scale gracefully with the dataset size using the adaptations of the vocabulary size, and the retrieval costs can be upper-bounded by a user-provided parameter.

In terms of precision the comparison to state-of-the-art methods is even more complicated, since the few works that study unsupervised features [2, 20] use diverse datasets, some of which are not even publicly available [2]. Therefore, we use the standard baseline DTW on normalized skeleton data as the common precision baseline, and furthermore compare our approach to the DTW alignment of soft MWs used in [20]. We significantly increase the retrieval accuracy from 54 % to 67 % compared to the unsupervised DTW alignment on normalized skeleton data, and also overcome the DTW-based alignment of soft MWs that achieved the precision of 61 % on our data. We also believe that the precision of the MW-based processing can be further increased by incorporating state-of-the-art unsupervised feature learning into the MW construction process. In particular, we plan to integrate unsupervised NN models into our solution and use them instead of DTW for all skeleton-level similarity measurements.

5 CONCLUSIONS

This paper introduces an efficient indexing and retrieval scheme for large collections of spatio-temporal human motion data. We apply the motion-word concept to transform skeleton sequences into structured text-like motion documents, and index these by extended inverted files. A new similarity-based ranking of soft motion words is used to significantly reduce the number of candidate documents in posting lists while maintaining high retrieval quality. The proposed approach should scale well to large motion collections as the top-ranked pruning of expanded lists guarantees a near-constant query processing time. To the best of our knowledge, the proposed solution is the first unsupervised scalable motion retrieval method. Since the principle of soft quantization, originally introduced for human skeleton data, is quite generic, the proposed indexing approach may also be applicable to other data domains.

ACKNOWLEDGMENTS

This research is supported by the Czech Science Foundation project No. GA19-02033S.

[1] http://disa.fi.muni.cz/research-directions/motion-data/data/

REFERENCES

[1] Fakhreddine Ababsa, Hicham Hadj-Abdelkader, and Marouane Boui. 2019. 3D Human Tracking with Catadioptric Omnidirectional Camera. In *International Conference on Multimedia Retrieval (ICMR)*. ACM, New York, NY, USA, 73–77. https://doi.org/10.1145/3323873.3325027

[2] Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Deep Motifs and Motion Signatures. *ACM Transactions on Graphics* 37, 6 (2018), 187:1–187:13. https://doi.org/10.1145/3272127.3275038

[3] Ricardo Baeza-Yates and Berthier A. Ribeiro-Neto. 2011. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England.

[4] Christian Beecks and Alexander Grass. 2018. Efficient Point-Based Pattern Search in 3D Motion Capture Databases. In *6th IEEE International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE Computer Society, 230–235. https://doi.org/10.1109/FiCloud.2018.00041

[5] Petr Byvshev, Pascal Mettes, and Yu Xiao. 2020. Heterogeneous Non-Local Fusion for Multimodal Activity Recognition. In *International Conference on Multimedia Retrieval (ICMR)*. ACM, 63–72. https://doi.org/10.1145/3372278.3390675

[6] Shuning Chang, Li Yuan, Xuecheng Nie, Ziyuan Huang, Yichen Zhou, Yupeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Towards Accurate Human Pose Estimation in Videos of Crowded Scenes. In *28th ACM International Conference on Multimedia (MM)*. ACM, 4630–4634. https://doi.org/10.1145/3394171.3416299

[7] Myung Geol Choi and Taesoo Kwon. 2019. Motion rank: applying page rank to motion data search. *The Visual Computer* 35, 2 (2019), 289–300. https://doi.org/10.1007/s00371-018-1498-6

[8] Mubbasir Kapadia, I-Kao Chiang, Tiju Thomas, Norman I. Badler, and Joseph T. Kider Jr. 2013. Efficient motion retrieval in large motion databases. In *Symposium on Interactive 3D Graphics and Games (I3D)*. ACM, 19–28. https://doi.org/10.1145/2448196.2448199

[9] Björn Krüger, Jochen Tautges, Andreas Weber, and Arno Zinke. 2010. Fast Local and Global Similarity Searches in Large Motion Capture Databases. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation (SCA)*. Eurographics Association, 1–10. https://doi.org/10.2312/SCA/SCA10/001-010

[10] Jianan Li, Xuemei Xie, Qingzhe Pan, Yuhan Cao, Zhifu Zhao, and Guangming Shi. 2020. SGM-Net: Skeleton-guided multimodal network for action recognition. *Pattern Recognition* 104 (2020), 1–38. https://doi.org/10.1016/j.patcog.2020.107356

[11] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. 2017. PKU-MMD: A Large Scale Benchmark for Skeleton-Based Human Action Understanding. In *Workshop on Visual Analysis in Smart and Connected Communities (VSCC@MM)*. ACM, 1–8. https://doi.org/10.1145/3132734.3132739

[12] Na Lv, Ying Wang, Zhiquan Feng, and Jingliang Peng. 2021. Deep Hashing for Motion Capture Data Retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2215–2219. https://doi.org/10.1109/ICASSP39728.2021.9413505

[13] Vladimir Mic, David Novak, and Pavel Zezula. 2019. Binary Sketches for Secondary Filtering. *ACM Transactions on Information Systems* 37, 1 (2019), 1:1–1:28. https://doi.org/10.1145/3231936

[14] Meinard Müller, Tido Röder, and Michael Clausen. 2005. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics* 24, 3 (2005), 677–685. https://doi.org/10.1145/1073204.1073247

[15] Konstantinos Papadopoulos, Enjie Ghorbel, Renato Baptista, Djamila Aouada, and Björn E. Ottersten. 2019. Two-Stage RGB-Based Action Detection Using Augmented 3D Poses. In *18th International Conference on Computer Analysis of Images and Patterns (CAIP)*, Vol. 11678. Springer, 26–35. https://doi.org/10.1007/978-3-030-29888-3_3

[16] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society. https://doi.org/10.1109/CVPR.2008.4587635

[17] Thanawin Rakthanmanon, Bilson J. L. Campana, Abdullah Mueen, Gustavo E. A. P. A. Batista, M. Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn J. Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 262–270. https://doi.org/10.1145/2339530.2339576

[18] Cheng Ren, Xiaoyong Lei, and Guofeng Zhang. 2011. Motion Data Retrieval from Very Large Motion Databases. In *International Conference on Virtual Reality and Visualization*. IEEE, 70–77. https://doi.org/10.1109/ICVRV.2011.50

[19] Tingxin Ren, Wei Li, Zifei Jiang, Xueqing Li, Yan Huang, and Jingliang Peng. 2020. Video-Based Human Motion Capture Data Retrieval via MotionSet Network. *IEEE Access* 8 (2020), 186212–186221. https://doi.org/10.1109/ACCESS.2020.3030258

[20] Jan Sedmidubsky, Petra Budikova, Vlastislav Dohnal, and Pavel Zezula. 2020. Motion Words: A Text-like Representation of 3D Skeleton Sequences. In *42nd European Conference on Information Retrieval (ECIR)*. Springer, 527–541. https://doi.org/10.1007/978-3-030-45439-5_35

[21] Jan Sedmidubsky, Petr Elias, Petra Budikova, and Pavel Zezula. 2021. Content-Based Management of Human Motion Data: Survey and Challenges. *IEEE Access* 9 (2021), 64241–64255. https://doi.org/10.1109/ACCESS.2021.3075766

[22] Jan Sedmidubsky, Petr Elias, and Pavel Zezula. 2019. Searching for variable-speed motions in long sequences of motion capture data. *Information Systems* 80 (2019), 148–158. https://doi.org/10.1016/j.is.2018.04.002

[23] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2018. Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection. *IEEE Transactions on Image Processing* 27, 7 (2018), 3459–3471. https://doi.org/10.1109/TIP.2018.2818328

[24] Yingying Wang and Michael Neff. 2015. Deep signatures for indexing and retrieval in large motion databases. In *8th ACM SIGGRAPH Conference on Motion in Games (MIG)*. ACM, 37–45. https://doi.org/10.1145/2822013.2822024

[25] Shuangyuan Wu, Zhaoqi Wang, and Shihong Xia. 2009. Indexing and retrieval of human motion data by a hierarchical tree. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*. ACM, 207–214. https://doi.org/10.1145/1643928.1643974