Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

Fast Local and Global Similarity Searches in Large Motion Capture Databases

Björn Krüger[1], Jochen Tautges[1], Andreas Weber[1] and Arno Zinke[2]

[1] Bonn University, Institute of Computer Science II
[2] GfaR mbH, Bonn

Abstract

Fast searching of content in large motion databases is essential for efficient motion analysis and synthesis. In this work we demonstrate that identifying locally similar regions in human motion data can be practical even for huge databases, if medium-dimensional (15–90 dimensional) feature sets are used for kd-tree-based nearest-neighbor-searches. On the basis of kd-tree-based local neighborhood searches we devise a novel fast method for global *similarity searches. We show that knn-searches can be used efficiently within the problems of (a) "numerical and logical similarity searches", (b) reconstruction of motions from sparse marker sets, and (c) building so called "fat graphs", tasks for which previously algorithms with preprocessing time quadratic in the size of the database and thus only applicable to small collections of motions had been presented. We test our techniques on the two largest freely available motion capture databases, the CMU and HDM05 motion databases comprising more than 750 min of motion capture data proving that our approach is not only theoretically applicable but also solves the problem of fast similarity searches in huge motion databases in practice.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—[Animation] Information Storage and Retrieval [H.3]: Information Search and Retrieval—

1. Introduction

Searching for similar motion segments is of central importance for data driven approaches of motion synthesis and content-based retrieval of motion data. Whereas efficient indexing techniques being linear in the size of the motion database have been described, for the problem of finding logically similar motions, methods such as neighbor graphs or similarity matrices have been used for tasks requiring numerically similar motions. These however require a preprocessing time quadratic in the size of the motion capture database in use and are therefore impractical for larger databases.

Due to the dimensionality of motion capture data and the "curse of dimensionality" of search structures such as BSP-trees or kd-trees [BBK01] these had not been applied for similarity searches of motions, as was succinctly expressed by [KG04] (format of references adapted to our references):

> One challenge in finding matches is that individual frames are high-dimensional objects with non-Euclidean distance metrics [KGP02,LCR*02]. As a result, traditional methods for organizing the data into a spatial hierarchy (such as a BSP-tree) can not be directly applied [BBK01].

In contrast to kd-trees, which speed up searches using Euclidean distance metrics, R-trees, which efficiently speed up searches in L_1 norms, have already been used in the context of motion data by [KPZ*04].

1.1. Our contributions

Devising feature sets for fast similarity searches. In this paper we describe and analyze medium dimensional feature sets for human motions (in general 15 to 90 dimensional ones). These can be used with naturally occurring Euclidean distance measures in standard spatial data structures—specifically kd-trees—to perform fast exact and approximate similarity searches in large motion capture databases for various purposes.

Analyzing different distance measures. We systematically compare previously-described distance measures with each other and with those induced by our feature sets. This comparison is done locally, i.e. on single frames, as well as globally, i.e. on motion segments, on the basis of the CMU [Car04] and HDM05 [MRC*07] databases.

Expanding pose matching to motion matching. On the basis of the fast kd-tree-based pose matching and local motion matching we devise a novel fast method for global motion matching. For a motion database of size n and a query sequence Q consisting of m frames using local k-nearest-neighbor-searches the overall complexity of the global similarity search is $O(km \log n)$, with $m \ll n$ and $k \ll n$.

Moreover, we show that distances on neighboring motion segments (parameterized by a local distance measure) induced by our novel technique are in general equivalent to the similarity measures computed by *dynamic time warping* (DTW) parameterized by the same local distance measure. Thus our method can be used as a fast alternative to subsequence DTW-based alignment.

Demonstrating the usability of fast similarity searches for different applications. We apply fast similarity searches of time complexity $O(n \log n)$ in the size of a database n to the problems of "numerical similarity searches", reconstruction of motions from sparse marker sets, and building so called "fat graphs", tasks for which previous algorithms with quadratic preprocessing time have been proposed.

2. Related Work

Nearest-neighbor-search for human motion. Chai and Hodgins [CH05] use a neighbor graph in a preprocessing step on a motion database allowing fast nearest-neighbor-search. However, the preprocessing step requires time quadratic to the size of the database and thus does not scale well to larger motion databases.

Kovar and Gleicher [KG04] perform numerical and "logical" similarity searches on collections of motion capture data. They build so called "match webs" on dense distance matrices, thus requiring a preprocessing time quadratic in the size of motion capture data.

The problem of finding short motion segments that are similar to a given one is also of central importance when synthetic transitions between motions are generated. Here the concept of "motion graphs" [KGP02,SO06,HG07,SH07, MP07] has become a central tool. However, in all these approaches the generation of the various variants requires an effort quadratic to the size of the motion database and thus cannot be used for large collections of motions.

Müller et al. [MRC05] use binary geometric features and index structures to address the problem of content-based retrieval on large motion databases. Whereas the binary geometric features are well suited for defining notions of logical similarity of motions and for coming up with "motion templates" [MR06], they are not suitable in contexts requiring close numerical similarity of motions.

The use of spatial search structures is well established for multi-media databases [BBK01]. Also a "generic multimedia indexing approach" (GEMINI) [Fal96] has been widely used for multi-media applications for more than a decade. However, the crucial step is to have suitable low-dimensional feature sets that can be used with an efficient spatial access method. In the context of motion data, its use and the use of R-trees are abstractly discussed in [FHP07], and Keogh et al. [KPZ*04] use R-trees for searching lower and upper bounds, which naturally yield L_1 norms, efficiently. However, prior to our own work presented here we are not aware of any practical attempts to define low- or medium dimensional feature sets for human motion data and using them both for efficient spatial access methods for Euclidean distance measures and for fast similarity searches in large motion databases.

The techniques of locality sensitive hashing (LSH) [AI08] for fast approximate nearest-neighbor-search in high dimensions has recently been applied to the problem of mining "motion motifs" from medium-sized collections of motion data (of about 32 000 frames) [MYHW08].

Low and medium dimensional feature sets for human motion. For small databases it is well known that human motions have very good 7–10 dimensional approximations [SHP04, EMMT04, CH05], which can be obtained by simple techniques like PCA (principal component analysis) on the angular skeleton representation. However, for large heterogenous databases such low-dimensional approximations are less accurate [CH05] and higher dimensional feature sets are required. Beaudoin et al. [BCvdPP08] use 18 dimensional PCA approximations of joint angle data.

The suitability of our medium dimensional geometric features for describing human poses is closely related to the well known analysis of the inverse kinematics problem for anthropomorphic limbs [TGB00]. An evaluation of different distance metrics for blending purposes is given in [vBE09].

3. Feature Sets for Fast Similarity Searches of Human Motions

In order to compare our newly-devised feature sets with existing ones, we will first review various distance measures for human motions and features sets (with induced distance measures) that have been described in the literature. Specifically we will fix the notations for them.

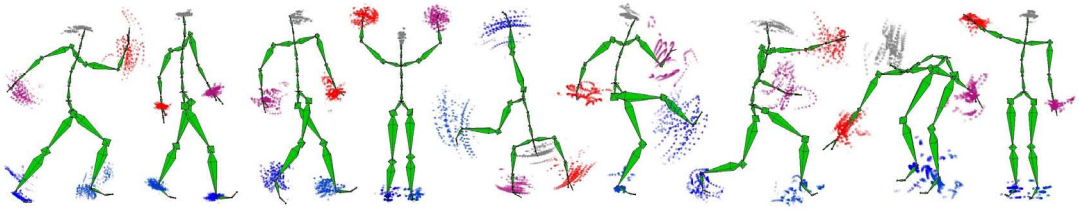

Figure 1: *Visualizations of the 256 nearest neighbors for 9 exemplary poses. All results were computed using feature set $\mathcal{F}_E^{15}$ on the union of CMU and HDM05 motion database. The positions of the wrist and ankle joints as well as head joints are visualized for the nearest neighbors (with color fading out with increasing distance). In all cases the nearest neighbors have been computed in a few milliseconds within databases containing more than 750 min of motion capture data.*

3.1. Previously described distance measures and feature sets for human motions

There are purely pose-based distance measures such as the one measuring distances on joint angles [CH05]. As the distance measure depends on the encoding of the joint angles, e.g. whether quaternion-based representations or Euler angle-based representations are used, we denote the former one by $\mathcal{D}_{\text{quat}}$ and the latter one by $\mathcal{D}_{\text{euler}}$. More specifically, the Euler angles in the standard asf/amc representation of the mocap data will be used. PCA-based compression of pose-based feature sets [SHP04, CH05, BCvdPP08] will be denoted by $\mathcal{F}_{\text{pca}}^{pn}$. Here, n means the number of principal components on joint positions in body frame—pre-computed on a fixed database, which will be chosen to be the entire HDM05 database in all our experiments (n dimensions).

In order to describe not only the properties of a pose statically but also to encode the kinematic properties of a motion sequence in the feature set of a frame, Kovar and Gleicher [KG04] introduced a point cloud distance measure on a normalized window of the previous and subsequent $n/2$ poses. In the following this distance measure will be denoted by $\mathcal{D}_{\text{pc}n}$.

In [LCR*02] the authors describe a cost function to determine transition points in motion streams, defining the distance between two frames as the sum of weighted differences of joint angles as well as of joint velocities. Whereas Lee et al. [LCR*02] propose a set of weights containing one and zero only—setting the weights to one for the shoulders, elbows, hips, knees, pelvis, and spine and setting all others to zero—Wang and Bodenheimer [WB03, WB08] use a refined cost metric. By using motion capture data to determine optimal values for all weights that modify the transition costs, they reason that only certain joints are considered important and thus are associated with non-zero weights–right and left hip, right and left knee, right and left shoulder, right and left elbow. The resulting distance measure, which is based on the optimized weights, will also be investigated and in the following will be denoted as $\mathcal{D}_{\text{wb}}$.

3.2. Devising novel medium dimensional feature sets

We will devise several medium dimensional feature sets of increasing dimensionality: we define frame-based geometric feature sets that can be extended to local feature sets on frame windows.

Frame-based feature sets For our geometric features we use normalized root positions and orientations, as is the standard technique for features of human motion data [KG04, CH05, AFO03]. Our primary feature set

$\mathcal{F}_{\mathbf{E}}^{\mathbf{15}}$ consists of the positions of 4 end-effectors and head.

This 15-dimensional feature set is motivated by the following considerations:

- As is well known the geometry of anthropomorphic limbs is fully determined by the positions of the end-effectors, their orientation, and one single additional scalar quantity—the so called "swivel angle" [TGB00]. Moreover, the corresponding inverse kinematics problem can be solved very efficiently using analytic solutions [TGB00, HJBC05].
- For typical human motions the orientations of the end-effectors are statistically quite dependent on the end-effector positions, as are the values of the swivel angles, so that the positions of the arms and legs should be well determined.
- Given the positions of the legs, the arms, and the head as well as the position of the root (due to normalization) there should be little variability in body positions.

For the sake of comparison (and for statistically validating the claims made above), we also use the following two pose-based geometric feature sets:

$\mathcal{F}_{\mathbf{E}}^{\mathbf{30}}$ Positions of 4 end-effectors, and head, as well as the 5 positions of the elbows, knees and one chest joint (30 dimensions).

$\mathcal{F}_{\mathbf{E}}^{\mathbf{39}}$ All features of $\mathcal{F}_{\text{E}}^{30}$; in addition position of the shoulders and one lower-back joint (39 dimensions).

Feature sets on windows of frames Purely pose-based feature sets such as $\mathcal{F}_{\text{E}}^{15}$ give no information about the temporal

evolution of a motion. In contrast, feature sets including several frames on a small window represent the local evolution in time. Based on this observation it is possible to extend every frame-based distance measure $\mathcal{F}^n$ of dimension n to one on a window of l frames $\mathcal{F}^{n \times l}$ of dimension nl. We will sample the windows sparsely, using only 3 or 5 frames on a window of fixed length 0.3 sec—a value commonly used in the literature for the size of local motion windows. The resulting feature sets will be denoted by $\mathcal{F}_{\mathrm{E}}^{15\times 3}$, $\mathcal{F}_{\mathrm{E}}^{15\times 5}$, $\mathcal{F}_{\mathrm{E}}^{30\times 3}$, and $\mathcal{F}_{\mathrm{E}}^{39\times 3}$.

3.3. Comparing feature sets

Pose based comparisons Our comparisons will be focused on feature sets designed to identify neighborhoods of a pose, because they are of main concern in the applications. As the search using kd-trees can be done efficiently for all of our feature sets, cf. Table 1, it is possible to do such comparisons systematically for large motion capture databases. For all of our experiments k-nearest-neighbor-searches were performed using the ANN library [MA06].

In Table 1 the computation times for the previously defined feature sets searching for the nearest 16 resp. 256 nearest neighbors on the HDM05 database (380 813 frames at 30 Hz) and CMU database (1 038 388 frames at 30 Hz) are given for exact ($\varepsilon = 0$) nearest-neighbor-searches.

Our frame-based (15 to 39 dimensional) feature sets allow very fast nearest-neighbor-searches and show the expected good scaling from a database consisting of 380 813 to one consisting of 1 038 388 frames.

The running times of the previously described features sets had the expected behavior according to their dimensionalities.

For the windowed feature sets (of dimension 45 to 90) the search times are about one order of magnitude higher than for the ones based on single frames—thus being much better than worst case theoretical considerations predict. Thus even if these feature sets do not in general fulfill hard real time requirements on current PCs for large motion databases, they are nevertheless practical for many applications.

However, as will be shown below the use of higher-dimensional feature sets, gives little or no advantage over the use of lower dimensional ones—specifically the simple feature set $\mathcal{F}_{\mathrm{E}}^{15}$.

Pose-based comparisons on a small sample database In Fig. 2 the correlations between the previously-described feature sets and distance measures are given. In order to compare the distance measures we use Spearman's rank correlation coefficient ρ [MW03], which is a robust measure with respect to commonly used slight but non-linear variations of the distance measures. This overall comparison is based on a small sample database as for the high-dimensional

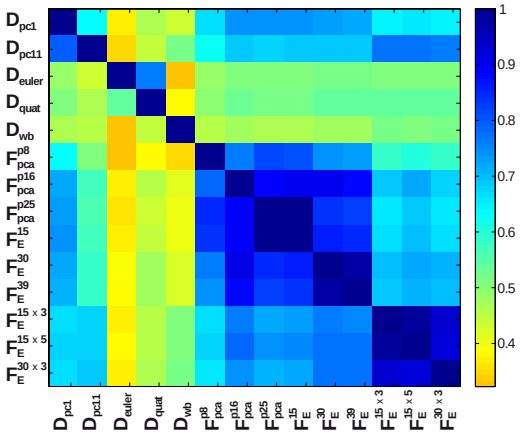

Figure 2: *Rank correlations between various distance measures and feature sets on an example database based on 76 motion clips taken from the HDM05 database: Average values for 1024 random samples choosing 256 nearest neighbors according to the feature set given in the vertical axis with the distance measure given in the horizontal axis.*

distance measures we do not have a fast nearest-neighbor-search method.

Notice that the matrices are *not symmetric*, as we perform the correlations on the nearest neighbors according to the feature set given on the vertical axis. This asymmetric behavior is especially prominent for $\mathcal{D}_{\mathrm{quat}}$ in comparison to $\mathcal{D}_{\mathrm{euler}}$ and the frame-based feature sets with their counterparts involving 3 or 5 frames. These observations are easily explainable: if the distance on angular representations given by Euler angles are similar, so are the ones given by quaternions, whereas similar quaternion-based distances might result in bigger differences in Euler angles (especially in "near gimbal lock" configurations). And if the distances according to a feature set involving l frames are similar, then so are the ones involving single frames, whereas vice versa, the similarities in one frame, involving static information only might result in fewer similar measures based on l frames.

Nevertheless, there is also a rather high rank correlation between $\mathcal{D}_{\mathrm{pc1}}$ and $\mathcal{D}_{\mathrm{pc11}}$, and frame-based feature sets and their counterparts involving 3 or 5 frames. Moreover, the distance measures based on $\mathcal{F}_{\mathrm{E}}^{15\times 3}$ and $\mathcal{F}_{\mathrm{E}}^{15\times 5}$ are *very highly correlated*: using 5 frame samples instead of 3 gives little additional information, so that the lower dimensional feature set can be used yielding lower computation times.

The distance measures based on $\mathcal{F}_{\mathrm{E}}^{30}$ and $\mathcal{F}_{\mathrm{E}}^{39}$ are *very highly correlated*: using the normalized root position, the head position and the ones of arms and legs there is almost no additional statistical variety in the body positions. Also $\mathcal{F}_{\mathrm{E}}^{15}$ and $\mathcal{F}_{\mathrm{E}}^{30}$ are highly correlated: there is already a very high statistical determination of the arm and leg positions from their end-effector positions.

Table 1: *Average computation times (in milliseconds) for searching 16 (256) nearest neighbors using various feature sets on motions of HDM05 database (380 813 frames at 30 Hz) and CMU database (1 038 388 frames at 30 Hz).*

database	# NN	$\mathcal{F}_{\text{pca}}^{\text{p8}}$	$\mathcal{F}_{\text{pca}}^{\text{p16}}$	$\mathcal{F}_{\text{pca}}^{\text{p25}}$	$\mathcal{F}_{\text{E}}^{15}$	$\mathcal{F}_{\text{E}}^{30}$	$\mathcal{F}_{\text{E}}^{39}$	$\mathcal{F}_{\text{E}}^{15\times 3}$	$\mathcal{F}_{\text{E}}^{15\times 5}$	$\mathcal{F}_{\text{E}}^{30\times 3}$
HDM05	16	0.19	1.33	2.86	1.11	3.87	5.27	7.95	14.01	18.98
	256	1.03	4.99	8.68	4.53	12.50	16.37	23.70	36.22	46.71
CMU	16	0.25	2.15	5.37	1.65	6.37	8.71	20.62	37.08	55.59
	256	1.35	8.97	18.10	7.36	24.12	31.68	60.23	96.55	136.29

Whereas $\mathcal{F}_{\text{pca}}^{\text{p16}}$ is comparable in general to other feature sets, the use of only 8 dimensions in $\mathcal{F}_{\text{pca}}^{\text{p8}}$ is connected with a strong loss in correlation. The feature sets $\mathcal{F}_{\text{pca}}^{\text{p16}}$ and $\mathcal{F}_{\text{pca}}^{\text{p25}}$ are very highly correlated indicating that most information on the motions is already contained in the first 16 principal components.

Pose-based comparisons on large databases Using our fast similarity searches we can extend the correlation analysis to large motion capture databases for the cases in which the correlations are computed on nearest neighbors defined by one of the medium-dimensional feature sets.

In general the findings are similar to the ones on the small sample database described above. Especially, the correlations between $\mathcal{F}_{\text{E}}^{15\times 3}$, $\mathcal{F}_{\text{E}}^{15\times 5}$, and $\mathcal{F}_{\text{E}}^{30\times 3}$ are still very high, and these are highly correlated to $\mathcal{D}_{\text{pc11}}$. The single-frame-based feature sets $\mathcal{F}_{\text{E}}^{30}$ and $\mathcal{F}_{\text{E}}^{39}$ are still *very highly correlated*, and there is still a high correlation to $\mathcal{F}_{\text{E}}^{15}$. Also a rather high correlation to $\mathcal{D}_{\text{pc1}}$ exists.

On this larger database the correlations to the PCA-based feature sets are somewhat lower, and so are the correlations between the single frame based feature sets and their counterparts involving 3 or 5 frames.

A figure showing all rank correlations corresponding to pose-based comparisons on a large database is given in the supplementary material.

Motion segment based comparisons We now focus on comparisons between motion segments, based on several feature sets and distance measures. These comparisons show relationships between the definitions of similarity induced by the feature sets and distance measures.

As example dataset we used 76 hand-cut motion sequences from the HDM05 database. This example dataset contains nine motion classes where at least eight motions of every class were available.

For direct comparison of the motion sequences we performed a dynamic time warping between each pair of 76 motion sequences for every feature set and distance measure. Based on this DTW-distances (accumulated pose distances along the warping path) we performed a ranking. As a result we get a ranked list of motions for each motion of the example dataset. Since the use of similar feature sets or distance measures should give similar rankings of the motion sequences, we computed rank correlations for the first eight motions of this ranking.

This rather small number is motivated by the consideration that we are concerned with the distance metrics on similar motions and not on very different ones (say a walking and a grasping motion)—and we know that at least eight motions of every class are available in the database.

In general the correlations are similar to the point-wise evaluations. There is a higher correlation for $\mathcal{D}_{\text{wb}}$ to almost all other distance measures and feature sets than for the point wise evaluation, whereas the correlations for the PCA-feature sets decrease. The correlations between our feature sets $\mathcal{F}_{\text{E}}^{15}$, $\mathcal{F}_{\text{E}}^{30}$, and $\mathcal{F}_{\text{E}}^{39}$ in between but especially to their counterparts involving several frames increase. The latter observation can be explained by the fact that for a point-wise evaluation the frame-based feature sets on the level of single frames *do not* distinguish directions of the motion in contrast to $\mathcal{F}_{\text{E}}^{15\times 3}$, $\mathcal{F}_{\text{E}}^{15\times 5}$, or $\mathcal{F}_{\text{E}}^{30\times 3}$, but nevertheless the warping paths are quite similar, cf. Fig. 3.

Motion segment based comparisons on large databases We could also perform motion segment-based comparisons on large databases using the fast global motion-matching methods described in Section 4.

The main differences to the evaluation on a small database are the higher correlations to the PCA-based feature sets. This finding might be explained by the fact that the PCA-based features are computed on the entire database, so that these perform better on random samples than on specifically-selected samples.

Conclusions from the comparisons As expected the low-dimensional feature sets $\mathcal{F}_{\text{E}}^{15}$ and the PCA-based feature sets allow the fastest nearest-neighbor searches. As the correlation of $\mathcal{F}_{\text{E}}^{15}$ to the higher-dimensional ones and to $\mathcal{D}_{\text{pc1}}$ as well as to $\mathcal{D}_{\text{pc11}}$ is higher than for $\mathcal{F}_{\text{pca}}^{\text{p16}}$ the former one should be preferred.

As $\mathcal{F}_{\text{E}}^{15}$ as a purely frame-based feature set does not distinguish directions of the motion in contrast to $\mathcal{F}_{\text{E}}^{15\times 3}$, there will be certain applications for which $\mathcal{F}_{\text{E}}^{15\times 3}$ should be used. However, we will show in Section 4 and Section 5 that for several applications, for which a priori one would suspect that including temporal information in a feature set plays a major role, nevertheless using the feature set $\mathcal{F}_{\text{E}}^{15}$ in the algorithms gives almost the same results as $\mathcal{F}_{\text{E}}^{15\times 3}$—but requires much shorter computation times.

So the simple feature set $\mathcal{F}_E^{15}$ seems to be the one of choice—especially for real-time applications.

4. From Pose Matching to Motion Matching

In many applications regarding analysis and synthesis of motions the problem of retrieving motion sequences within large unstructured motion databases, that are similar to a given query, is of central importance [KGP02, AFO03, KG04,SO06,HG07,SH07,MP07,BCvdPP08]. In the context of computer animation this problem was previously tackled by applying either *subsequence DTW* [Mül07] or *match web* [KG04] heuristics. Unfortunately, the above methods are computationally very costly. The construction of match webs in $O(n^2)$ and subsequence DTW has a complexity of $O(nm)$ where m is the size of the query and n is the number of frames included in the database.

In this section we are going to present a novel method in $O(m \log n)$ that gives similar results to subsequence DTW in practical scenarios and is more general and robust than match webs, since no ad-hoc heuristics are used. Our new approach is especially suitable for identifying closest matches to a given query. Practical applications demonstrating the efficiency of the method are given in section 5.

4.1. A novel approach to fast global motion matching

In order to specify global motion-matching we follow the literature and define *a valid temporal alignment* of two motions as a continuous and monotonic mapping of poses [KG04]. For an optimal alignment of two motions we have to search for a sequence of consecutive frames (with ascending indices), which describes a discrete matching substitute to subsequence dynamic time warping.

To find similar motion segments included in a database (a sequential collection of motion clips indexed by frame) for a given query Q efficiently we propose to use a novel technique, based on a "lazy neighborhood graph". The novel method consists of four different key steps:

1. *preprocessing*, where a kd-tree is constructed,
2. *search*, for identifying local similarities of Q and motions included in the database,
3. *graph construction*, for creating a lazy neighborhood graph,
4. *path search*, for finding global optimal alignments by solving a shortest path problem for this neighborhood graph.

Preprocessing As a preprocessing step we build a kd-tree for a motion database D (of size n frames) with respect to the feature set $\mathcal{F}$ to be considered.

For each query sequence $Q = [q_1 \cdots q_m]$ consisting of m frames we proceed as follows:

Search Find nearest neighbors for each pose in Q using fixed radius k-nearest-neighbor-search. The radius is given by r and the maximum number of neighbors is limited by user defined parameter k. For each pose q_i of the query a set $S(q_i)$ of poses that are similar according to $\mathcal{F}$ is retrieved in $O(k \log n)$. Thus, in total km neighboring poses have to be stored, which requires km space.

Graph construction Build a weighted and directed graph based on the sets S by regarding each reported neighbor $h_j(q_i)$, $1 \leq j \leq k$ and $1 \leq i \leq m$, as a node and adding edges between nodes that form *valid continuations*. While many definitions of valid continuations are thinkable, we define them equivalent to the basic steps most commonly used in traversing DTW cost matrices, i.e. a diagonal, a horizontal and a vertical step. Formally spoken, this leads to edges between

- $h_j(q_i)$ and $h_l(q_{i+1})$ with $h_l(q_{i+1}) = h_j(q_i) + 1$ (corresponding to the diagonal step),
- $h_j(q_i)$ and $h_l(q_i)$ with $h_l(q_i) = h_j(q_i) + 1$ (corresponding to the vertical step), and
- $h_j(q_i)$ and $h_l(q_{i+1})$ with $h_l(q_{i+1}) = h_j(q_i)$ (corresponding to the horizontal step).

Associating each edge with costs defined by the distance d_j of the node they are pointing at—as reported by the kd-tree search—the task is now to find the paths with minimal costs that start in a node $h_j(q_1) \in S(q_1)$ and end up in a node $h_j(q_m) \in S(q_m)$.

Path search By adding one additional node to the graph and connecting it via edges to all $h_j(q_1) \in S(q_1)$ this task turns into a single-source shortest-path problem. Since the resulting graph is directed and acyclic and a topological ordering of its nodes—which means, whenever there is an edge from x to y, the ordering visits x before y—is directly given by construction, this problem can be solved in linear time [CLRS01] (chapter 24.2).

This algorithm is parameterized by an arbitrary feature set $\mathcal{F}$. The global accumulated costs along the path define a global distance between the query motion and the motion segments found in the database. Thus the retrieved motion segments can be ranked by their global distance according to the selected feature set $\mathcal{F}$. By this algorithm similar motion segments can be extracted in $O(km)$ time and the overall complexity for the similarity search is given by $O(km \log n)$.

The algorithm returns a best path for each match. These paths give a global optimal alignment between the query motion and the retrieved motion segments with respect to the local neighborhoods of each frame of the query motion.

The paths found by our method are equal to the paths found by subsequence dynamic time warping [Mül07] under the condition that all frames that are assigned to each other by subsequence DTW are in the neighborhood of the query motion. On tests on smaller databases such as the 76 cut motions this assumption was fulfilled in 100 %

of the cases. For the tests on the entire HDM05 database the rank correlation between the global distances based on our approach and the global distances computed by dynamic time warping on the same feature sets was bigger than 0.99 for all feature sets (but not 1.0). Thus, for larger databases the assumption that all frames that are assigned to each other by subsequence DTW are in the neighborhood of the query motion fails in in a few rare cases. In addition this issue becomes less relevant for applications in which only close matches are required. In such cases our approach can be seen to be an extremely efficient substitute for subsequence dynamic time warping.

Remark. Our algorithm was inspired by the trellis approach to extract motion motifs by Meng et al. [MYHW08]. The trellis approach as well as ours searches for local nearest neighbors parameterized by some distance measure. However, in the trellis approach only constant distances d_j independent of the actual local costs are used. For that reason the original method fails to provide a temporal alignment of motion segments as it is achieved by our method. Moreover, there is no ranking of retrieved subsequences meaning that closest matches cannot be identified. Another serious drawback of the method described in [MYHW08] is the use of greedy strategies that in general fail to find global optimal alignments.

4.2. Comparing the global motion matching for different feature sets

Based on the algorithm described in the previous section we compare the results of our global matching with respect to various feature sets. For the sake of simplicity only a small sample database containing six steps of a left-turning walking motion, starting with the right foot, is considered. A motion clip consisting of a two-step left-turning walking motion, starting with the right foot, is used as query.

As can be seen from the similarity matrices in Fig. 3 the pose-based feature sets exhibit forward as well as backward diagonal structures for similar frames of query and database, as these pose-based feature sets do not incorporate velocity information and do not distinguish directly between forward and backward in motions (e.g. walking and running). Note that the windowed feature sets do not suffer from this issue. Nevertheless, the global motion-matching algorithm is able to identify the expected walking motions regardless of the feature set that was used for alignment. Note that the motion segments found by our algorithm are similar to the segments found by subsequence DTW based on $\mathcal{D}_{\text{pc1}}$. As the structure of the similarity matrices suggests, the task of finding warping paths is less constrained for the frame-based feature sets and thus more costly, since more potential alignments need to be investigated.

We did a more complete evaluation on the entire HDM database using 128 random motions of length 1 sec and

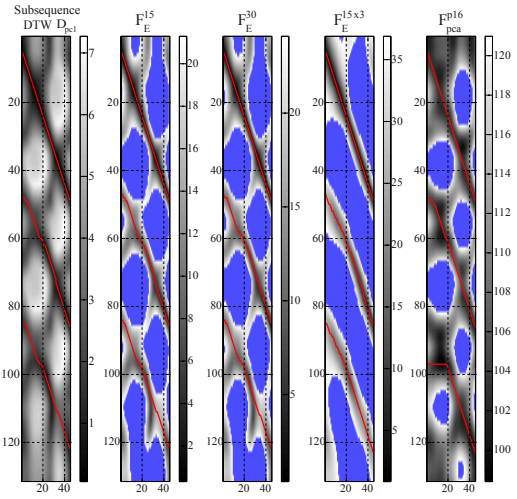

Figure 3: *Aligning a query motion segment to similar motions of a small database. The distance matrices implicitly computed by our approach are plotted for four different feature sets. A distance matrix based on $\mathcal{D}_{pc1}$ and warping paths computed by subsequence DTW are shown for comparison. Please note that only frames found during k-nearest-neighbor-search (indicated by shades of grey, the darker the more similar) need to be considered explicitly and that the blue regions do not have to be computed. The red lines represent the recovered warping paths.*

128 random motions of length 3 sec as queries. Searching alignments for windowed feature sets (excluding k-nearest-neighbor-search) was about 1.5 times faster than for frame-based ones. For the queries of length 1 sec (3-sec) it took on average 0.03 sec (0.08 sec.) to construct the alignments using exact nearest-neighbor-searches for the feature set $\mathcal{F}_{\text{pca}}^{\text{p16}}$. These experiments confirm that the asymptotic linear complexity in the length of the query of our algorithm also occurs in practice. Using approximate ($\varepsilon = 0.1$) instead of exact nearest-neighbor-searches gave exactly the same results—with computation times for the kd-tree searches being about 10 % lower. When increasing epsilon to 0.5 about 99 % of the original alignments for the window-based feature sets were found. For the frame-based feature sets this rate dropped to 90 %–95 % and the times for computing the alignment increased for approximate k-nearest-neighbor-search: 0.02–0.06 sec on average for the 1 sec queries (0.09–0.16 sec for the 3 sec queries). Notice that the total timings for global motion-matching are dominated by the costs for k-nearest-neighbor-search (90%) cf. Table 1.

5. Applications

5.1. Numerical and Logical Similarity Searches

Kovar and Gleicher [KG04] presented a technique that allows numerical as well as "logical" similarity searches

Table 2: *Results for numerical similarity searches on the entire HDM05 database. A motion clip semantically classified to the motion class in the top row was used as query. The rows show the number of correct hits and in brackets the number of wrong hits that were not identified by our reference feature set $\mathcal{F}_E^{15}$, the number of false hits and finally in brackets the number of false hits not identified by $\mathcal{F}_E^{15}$.*

HDM_bk_kickRSide1Reps_007

	$\mathcal{F}_{\mathrm{E}}^{15}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p8}}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p16}}$	$\mathcal{F}_{\mathrm{E}}^{30}$	$\mathcal{F}_{\mathrm{E}}^{15\times 3}$	$\mathcal{F}_{\mathrm{E}}^{30\times 3}$
correct hits	5(-)	1(0)	4(0)	5(0)	3(0)	4(0)
wrong hits	0(-)	0(0)	0(0)	0(0)	0(0)	0(0)

HDM_tr_punchRSide1Reps_023

	$\mathcal{F}_{\mathrm{E}}^{15}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p8}}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p16}}$	$\mathcal{F}_{\mathrm{E}}^{30}$	$\mathcal{F}_{\mathrm{E}}^{15\times 3}$	$\mathcal{F}_{\mathrm{E}}^{30\times 3}$
correct hits	6(-)	4(0)	6(0)	8(2)	6(3)	9(6)
wrong hits	0(-)	0(0)	0(0)	0(0)	0(0)	2(2)

HDM_bk_skier1RepsLStart_011

	$\mathcal{F}_{\mathrm{E}}^{15}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p8}}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p16}}$	$\mathcal{F}_{\mathrm{E}}^{30}$	$\mathcal{F}_{\mathrm{E}}^{15\times 3}$	$\mathcal{F}_{\mathrm{E}}^{30\times 3}$
correct hits	9(-)	8(0)	9(0)	9(0)	10(1)	10(1)
wrong hits	0(-)	0(0)	0(0)	0(0)	0(0)	0(0)

in motion databases. However, their technique does not scale to larger motion databases, as they have to compute a dense distance matrix of size $O(n^2)$, where n is the number of frames in the database. Using the novel algorithm described in Section 4.1 we have a direct substitute for a "numerical similarity" search that scales well to huge motion databases. This technique allows for a substantial speedup of "logical similarity searches" originally proposed by Kovar and Gleicher [KG04]: use the set of found similar motion segments as queries for new iterations of similarity searches.

Not only do we avoid the preprocessing step of quadratic complexity to the size of the database n, but also for each query the cost of our method is only logarithmic to the size of the database instead of being linear as is the one from Kovar and Gleicher [KG04].

The basic properties of our numerical similarity search approach and results on a very small data sample were already presented in Section 4.2. A comparison of search results on a large motion database for different feature sets is given in table 2.

As could be expected from the prototypical results on the very small database given in Fig. 3, the results using the simple feature set $\mathcal{F}_{\mathrm{E}}^{15}$ are the same or almost the same as for the higher dimensional feature sets $\mathcal{F}_{\mathrm{E}}^{30}$, $\mathcal{F}_{\mathrm{E}}^{39}$, $\mathcal{F}_{\mathrm{E}}^{15\times 3}$, $\mathcal{F}_{\mathrm{E}}^{15\times 5}$, and $\mathcal{F}_{\mathrm{E}}^{30\times 3}$ on the large database. Notice that there are examples for which the simple feature sets return more similar motions than the higher-dimensional ones, and there are examples for which the higher dimensional ones return more similar motions than $\mathcal{F}_{\mathrm{E}}^{15}$. Also the feature sets $\mathcal{F}_{\mathrm{pca}}^{\mathrm{p16}}$ and $\mathcal{F}_{\mathrm{pca}}^{\mathrm{p25}}$ yield similar search results, whereas $\mathcal{F}_{\mathrm{pca}}^{\mathrm{p8}}$ differs and also returns results not regarded as being similar by a human classification of the motions.

Table 3: *Average (max) reconstruction errors (in cm per joint) for test motions from CMU and HDM05 database.*

motion	#frames	HDM	CMU$^-$	CMU
CMU_86_01	1145	2.63 (6.47)	2.06 (4.11)	1.30 (3.42)
CMU_86_08	2302	3.06 (6.67)	2.67 (6.95)	1.96 (6.82)
CMU_86_15	1773	3.37 (6.41)	2.64 (8.01)	2.31 (5.88)
CMU_86_avg	29040	2.79 (7.99)	2.30 (5.58)	1.74 (5.05)
HDM_bk_01-01_01	2571	1.33 (5.13)	2.69 (5.04)	2.70 (5.43)
HDM_bk_02-01_01	912	2.17 (6.45)	3.23 (8.68)	3.22 (8.65)

We give the values for motions 86_01, 86_08, and 86_15 of CMU database, and for motions HDM_bk_01-01_01 and HDM_bk_02-01_01 of HDM05 database. The average error over the 15 motions in collection 86 of the CMU database is denoted as CMU_86_avg. We give the values for different databases for the pose priors. HDM: entire HDM05 database (possibly without test motion). CMU$^-$: CMU database without collection 86. CMU: entire CMU database (possibly without test motion). In all cases feature set $\mathcal{F}_{\mathrm{E}}^{15}$ is used for the nearest-neighbor-search.

5.2. Reconstructions of Motions from Few Markers

Reconstructing motions from only a few markers is a challenging task that was recently tackled by Chai and Hodgins [CH05]. For their approach, identifying poses in a database as being similar to a given medium-dimensional control signal (sparse marker position data) is of central importance. The necessary pose-based nearest-neighbor-search was implemented by using a neighborhood graph, which requires a preprocessing effort quadratic to the size of the underlying motion capture database. Replacing the nearest-neighbor-search in a static graph by our fast kd-tree-based search method (on various of our feature sets) even orders of magnitude larger collections of motions become practical. Moreover, with kd-trees we can search around arbitrary, i.e. newly synthesized poses not included in the original database directly. Hence, we do not have to approximate the nearest-neighbor-search by using nearest neighbors of previously synthesized frames, as has to be done in [CH05].

In order to have ground truths for the quality of the reconstruction we performed the reconstructions on synthetic data obtained from test motions from the databases: the positions of the 4 end effectors, the head, and the root are taken as "marker positions" randomly disturbed within a range of 1mm (simulating measurement errors of an optical marker tracking).

The results given in Table 3 indicate that using the basic technique of Chai and Hodgins [CH05] a motion can be reconstructed reliably with 6 markers only, if large databases can be used to infer local statistics on motions (e.g. pose priors based on local linear models).

The quality of the reconstructions increases, if more motions related to the one to be reconstructed from sparse marker data are available—a result which is certainly expected but nevertheless shows the need to have fast similarity searches for motions on huge motion databases, a

Table 4: *Average (max) reconstruction errors (in cm per joint) for a test motion using different feature sets for nearest-neighbor-searches. Entire CMU database without test motion was used for pose prior.*

motion	$\mathcal{F}_{\mathrm{E}}^{15}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p8}}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p16}}$	$\mathcal{F}_{\mathrm{pca}}^{\mathrm{p25}}$	$\mathcal{F}_{\mathrm{E}}^{30}$	$\mathcal{F}_{\mathrm{E}}^{39}$	$\mathcal{F}_{\mathrm{E}}^{15\times 3}$	$\mathcal{F}_{\mathrm{E}}^{15\times 5}$	$\mathcal{F}_{\mathrm{E}}^{30\times 3}$
CMU_86_03	1.66	1.71	1.49	1.41	1.35	1.34	1.38	1.40	1.30
	(4.73)	(5.09)	(5.25)	(4.80)	(4.55)	(4.92)	(4.80)	(4.79)	(4.35)

possibility opened by our method but not given by the original method of Chai and Hodgins [CH05].

As can be seen from the results shown in Table 4 the motion reconstruction procedure works quite well for all of the feature sets we have tested: it is enough to build a local linear model on a set of somewhat-related neighbors. Even those computed by a global 8 dimensional pca (as for feature $\mathcal{F}_{\mathrm{pca}}^{8}$), which per se do not approximate the current motion segment well, are sufficient as a basis for the used local linear model. On the other hand the windowed feature sets $\mathcal{F}_{\mathrm{E}}^{15\times 3}$, $\mathcal{F}_{\mathrm{E}}^{15\times 5}$, or $\mathcal{F}_{\mathrm{E}}^{30\times 3}$ give almost identical results, as do their frame-based counterparts. Also exact nearest-neighbor-searches can be substituted with approximate ones using $\varepsilon = 0.5$ without changing the reconstruction results notably.

5.3. Fast Fat Graphs

We can also come up with a method of substituting the quadratic preprocessing time in the construction of so-called "fat graphs" [SO06]—a method for motion synthesis—with one in $O(n \log n)$ by using kd-tree-based searches. The crucial step in building "fat graphs" is the computation of so-called "base poses", a clustering of motion capture data collections. For a collection of motion data $D = [d_1 \dots d_n]$, for which a "fat graph" is to be computed, proceed as follows:

1. Search nearest neighbors for each frame $f \in D$ in a fixed radius r; the maximum number of neighbors is limited to k. This search can be done in $O(kn \log n)$.
2. Find the pose with maximum number of neighbors and use it as "base pose". This step can be done in $O(n)$.

As $k \ll n$ is constant, the complexity of finding the base poses is only $O(n \log n)$ instead of $O(n^2)$, as is the method used in [SO06].

As we have to use a distance measure related to one of our feature sets, the search criterion will be different from the one used in the original construction of "fat graphs". However, in the experiments we performed, the synthesized motions differ only slightly from the ones generated using the original fat graph approach. Moreover, the visual quality of both results is comparable. We refer to the accompanying video for examples.

6. Conclusion and Future Work

In this work, efficient approaches for local and global motion matching, which are applicable even to huge databases, have been presented. Using these novel techniques we have reduced the time complexity of being quadratic to the size of the motion database n to one of at most $O(n \log n)$ for three very different applications in the realm of data-driven animation. From a practical point of view this means an enhanced applicability of these methods to large databases.

We presume that for other problems something similar can be achieved. Specifically, we also consider the adaption of our approaches for local and global motion matching to parametric motion graphs [HG07] and interpolated motion graphs [SH07]. Moreover, the technique described by Chai and Hodgins [CH07] to generate animations from user defined constraints, which uses a global preprocessing on a medium-sized database of human motions seems to be localizable by our technique and thus extendable to huge databases involving very different motions.

It turned out that fast kd-tree-based nearest-neighbor-searches together with viable medium-dimensional feature sets are highly practical even for amounts of motion capture data two orders of magnitude bigger than has been done with any previously applied techniques.

The kd-trees for even the largest currently-available motion capture databases still fit into main memory on current standard PCs, and the memory requirements for kd-trees are much lower than the one required when using locality sensitive hashing for the same data. Nevertheless, our techniques would not scale well if the kd-trees did not fit into the main memory. An adaptation to out-of-core-techniques, e.g. lazy kd-trees [Nar96, HMF07], will become a topic of future research if the available mocap data grow faster than the available main memory.

On the basis of our techniques, data-driven approaches requiring nearest-neighbor-searches on motion data can work efficiently on much larger collections of motion capture databases than are currently available.

Acknowledgements

We thank all anonymous referees for constructive and valuable comments.

This research is financially supported in part by *Deutsche Forschungsgemeinschaft* under grant WE 1945/5-1

References

[AFO03] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Motion synthesis from annotations. *ACM Transactions on Graphics 22*, 3 (2003), 402–408. 3, 6

[AI08] ANDONI A., INDYK P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM 51*, 1 (2008), 117–122. 2

[BBK01] BÖHM C., BERCHTOLD S., KEIM D. A.: Searching

in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv. 33*, 3 (2001), 322–373. 1, 2

[BCvdPP08] BEAUDOIN P., COROS S., VAN DE PANNE M., POULIN P.: Motion-motif graphs. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), Gross M., James D., (Eds.). 2, 3, 6

[Car04] CARNEGIE MELLON UNIVERSITY GRAPHICS LAB: CMU Motion Capture Database, 2004. mocap.cs.cmu.edu. 2

[CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. *ACM Trans. Graph. 24*, 3 (2005), 686–696. SIGGRAPH 2005. 2, 3, 8, 9

[CH07] CHAI J., HODGINS J. K.: Constraint-based motion optimization using a statistical dynamic model. *ACM Transactions on Graphics 26*, 3 (2007). SIGGRAPH 2007. 9

[CLRS01] CORMEN T. H., LEISERSON C. E., RIVEST R. L., STEIN C.: *Introduction to Algorithms*. MIT Press, 2001. 6

[EMMT04] EGGES A., MOLET T., MAGNENAT-THALMANN N.: Personalised real-time idle motion synthesis. In *12th Pacific Conference on Computer Graphics and Applications (PG 2004)* (2004), pp. 121–130. 2

[Fal96] FALOUTSOS C.: *Searching Multimedia Databases by Content*. Springer, 1996. 2

[FHP07] FALOUTSOS C., HODGINS J., POLLARD N.: Database techniques with motion capture. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (San Diego, California, 2007), ACM, p. 1. 2

[HG07] HECK R., GLEICHER M.: Parametric motion graphs. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2007), ACM Press, pp. 129–136. 2, 6, 9

[HJBC05] HILDENBRAND D., J. Z., BAYRO-CORROCHANO E.: Inverse kinematics computation in computer graphics and robotics using conformal geometric algebra. In *International Conference on Clifford Algebras and their Applications* (2005). 3

[HMF07] HUNT W., MARK W. R., FUSSELL D.: Fast and lazy build of acceleration structures from scene hierarchies. In *IEEE Symposium on Interactive Ray Tracing (RT '07)* (2007), pp. 47–54. 9

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics 23*, 3 (2004), 559–568. SIGGRAPH 2004. 1, 2, 3, 6, 8

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Transactions on Graphics 21*, 3 (2002), 473–482. SIGGRAPH 2002. 1, 2, 6

[KPZ*04] KEOGH E., PALPANAS T., ZORDAN V. B., GUNOPULOS D., CARDLE M.: Indexing large human-motion databases. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases* (2004), VLDB Endowment, pp. 780–791. 1, 2

[LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (San Antonio, Texas, 2002), ACM Press, pp. 491–500. 1, 3

[MA06] MOUNT D. M., ARYA S.: *ANN: A Library for Approximate Nearest Neighbor Searching*. Programming manual, Department of Computer Science, University of Maryland, College Park, Maryland, U.S.A., 2006. 4

[MP07] MCCANN J., POLLARD N.: Responsive characters from motion fragments. *ACM Transactions on Graphics 26*, 3 (2007). SIGGRAPH 2007. 2, 6

[MR06] MÜLLER M., RÖDER T.: Motion templates for automatic classification and retrieval of motion capture data. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), ACM Press, pp. 137–146. 2

[MRC05] MÜLLER M., RÖDER T., CLAUSEN M.: Efficient content-based retrieval of motion capture data. *ACM Trans. Graph. 24*, 3 (2005), 677–685. SIGGRAPH 2005. 2

[MRC*07] MÜLLER M., RÖDER T., CLAUSEN M., EBERHARDT B., KRÜGER B., WEBER A.: *Documentation: Mocap Database HDM05*. Computer Graphics Technical Report CG-2007-2, Universität Bonn, May 2007. www.mpi-inf.mpg.de/resources/HDM05. 2

[Mül07] MÜLLER M.: *Information Retrieval for Music and Motion*. Springer, 2007. 6

[MW03] MYERS J. L., WELL A. D.: *Research design and statistical analysis*. Mahwah, N.J. : Lawrence Erlbaum Associates, 2003. 4

[MYHW08] MENG J., YUAN J., HANS M., WU Y.: Mining motifs from human motion. In *Eurographics 2008 – Short Papers* (2008), Mania K., Reinhard E., (Eds.), pp. 71–74. 2, 7

[Nar96] NARDELLI E.: Distributed *k*-d trees. In *Proceedings 16th Conference of Chilean Computer Science Society (SCCC '96)* (1996), pp. 142–154. 9

[SH07] SAFONOVA A., HODGINS J. K.: Construction and optimal search of interpolated motion graphs. *ACM Transactions on Graphics 26*, 3 (2007). SIGGRAPH 2007. 2, 6, 9

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics 23*, 3 (2004), 514–521. SIGGRAPH 2004. 2, 3

[SO06] SHIN H. J., OH H. S.: Fat graphs: constructing an interactive character with continuous controls. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, 2006), Eurographics Association, pp. 291–298. 2, 6, 9

[TGB00] TOLANI D., GOSWAMI A., BADLER N. I.: Real-time inverse kinematics techniques for anthropomorphic limbs. *Graph. Models Image Process. 62*, 5 (2000), 353–388. 2, 3

[vBE09] VAN BASTEN B. J. H., EGGES A.: Evaluating distance metrics for animation blending. In *FDG '09: Proceedings of the 4th International Conference on Foundations of Digital Games* (New York, NY, USA, 2009), ACM, pp. 199–206. 2

[WB03] WANG J., BODENHEIMER B.: An evaluation of a cost metric for selecting transitions between motion segments. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 232–238. 3

[WB08] WANG J., BODENHEIMER B.: Synthesis and evaluation of linear motion transitions. *ACM Trans. Graph. 27*, 1 (2008), 1–15. 3