

Indexing 3-D Human Motion Repositories for Content-Based Retrieval

Gaurav N. Pradhan and Balakrishnan Prabhakaran, *Member, IEEE*

Abstract—Content-based retrieval of the similar motions for the human joints has significant impact in the fields of physical medicine, biomedicine, rehabilitation, and motion therapy. In this paper, we propose an efficient indexing approach for 3-D human motion capture data, supporting queries involving both subbody motions as well as whole-body motions.

Index Terms—Content-based retrieval, indexing, motion capture, singular value decomposition (SVD).

I. INTRODUCTION

HUMAN motion events can be represented in the form of 3-D positional and rotational information of joints in space over time. Such information can be used to better analyze and quantify the complex human body motions for gait analysis and several orthopedic applications, such as joint mechanics, prosthetic designs, and sports medicines. Physical medicine and rehabilitation [1], biomechanics [2], and physiology are the applications that require a large repository of 3-D human motions wherein similarity searches and comparisons can be made on whole human body or specifically on some part of body like arm, leg, etc.

Fig. 1 shows the retroreflective markers on the participant's body in the 3-D space. As the participant keeps moving, the motion capture technology tracks the movement of these markers (for instance, the comparison of the movement of the foot segment for a similar motion performed by two different participants in 3-D space is shown in the figure) and gives the exact position of the joints/segments in the 3-D space. Hence, every motion is represented by a matrix called as “*whole-body matrix*,” which contains the 3-D positional information for all joints. The 3-D positional information specific to a joint (i.e., “*joint motion*”) in “*whole-body matrix*” is given by a three-column matrix called as “*joint matrix*.” The rows of these matrices represent the number of frames captured.

In general, human body can be considered to have 18 major joints. Fig. 2(a) and (b) shows 54 trajectories of 18 joints in all three dimensions while performing walking motions of two different subjects. Since human motions are mostly natu-

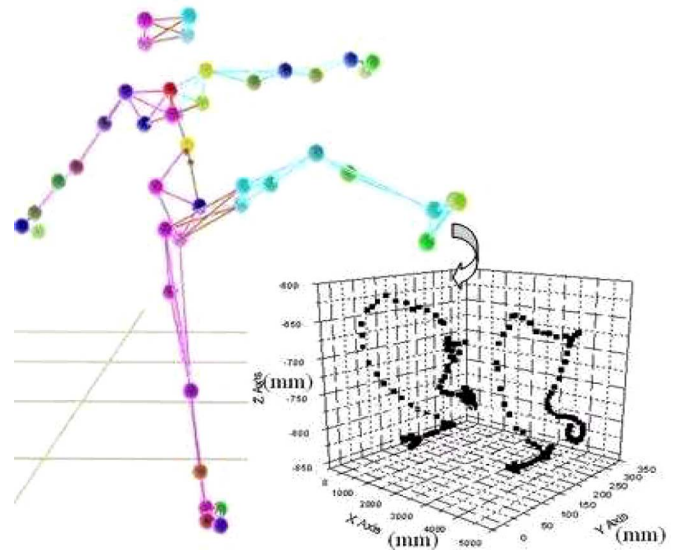


Fig. 1. Reflectors mapping in 3-D space for the same motion.

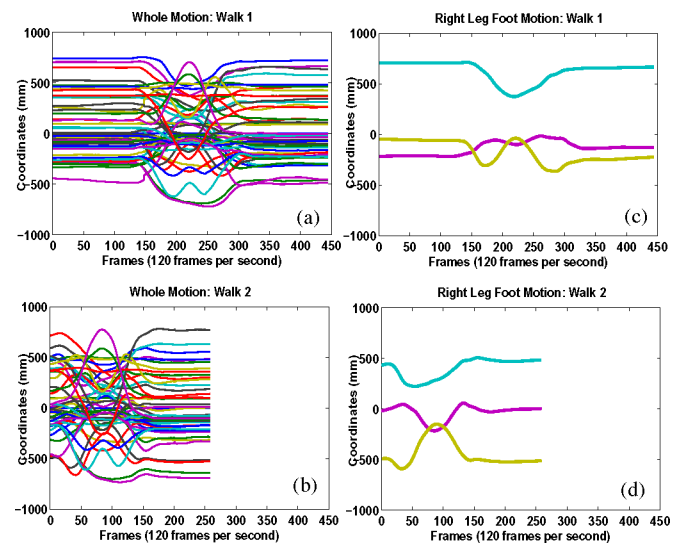


Fig. 2. (a) and (b) X , Y , Z trajectories of all segments for two similar walking motions with variation in time and speed. (c) and (d) Corresponding trajectories of only foot segments.

Manuscript received August 26, 2007; revised July 16, 2008, November 21, 2008, and March 18, 2009. First published April 28, 2009; current version published September 2, 2009.

G. N. Pradhan is with the Department of Computer Science, Arizona State University, Tempe, AZ 85287 USA (e-mail: gaurav.pradhan@asu.edu).

B. Prabhakaran is with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080-3021 USA (e-mail: praba@utdallas.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITB.2009.2021262

ral activities, semantically similar motions such as walking can have large variations. For example, when a person is walking, he/she can wave his/her hands, swing his/her arms, or just have his/her arms crossed. Now, let us consider a query by a physical medicine and rehabilitation specialist to *find similar foot*

motions while walking. Resolution of this type of subbody queries can be affected if we consider the different possibilities for arm motions. To focus only on foot motions, we can extract only positional information of the foot joint, as shown in Fig. 2(c) and (d).

Thus, the main objective of this paper is to find similar 3-D human motions by constructing the indexing structure that supports queries on subbody motions in addition to whole-body motions. We focus on content-based retrieval for the subbody queries such as *find similar shoulder motions*, *find similar leg motions*, etc., or more regular query on whole body such as *find similar walking human motions*. Some of the major challenges in indexing large 3-D human motion databases are as follows.

- 1) 3-D motions are multidimensional, multiattribute, and correlated in nature; associated segments of one subbody (e.g., hand) should be processed always together along every dimension.
- 2) The result/ranking of similar patterns to subbody query is influenced by movements of other subbody parts.
- 3) It is difficult to come up with a similarity measure/metric on 3-D human motions due to unexpected variations in performances for similar motions (such as walking) and also due to different behaviors of segments depending on the hierarchical structure of the human body.
- 4) Human motions exhibit huge variations in speed for similar motions as well as in directionality.

In our approach, we index the positional information of different human body joints in a motion as the feature points. Using these indexed feature points, a composite index structure for 3-D human motions comprising five index trees (corresponding to subbody parts body, left hand, right hand, left leg, and right leg) is constructed. Each level of the index tree is designated to a joint of the corresponding subbody part depending on the hierarchical structure of the human body joints. Each level prunes the irrelevant motions for the given query with respect to the associated joint. Finally, each index tree with respect to the corresponding subbody part gives the relevant motions that are ranked using a similarity measure. For the whole motion query, the outputs from all index trees are merged, and then ranked to get the most relevant motions.

This paper extends our previous work [3] on hierarchical indexing structure for 3-D human motions. As compared to [3], in this paper, we have achieved significant improvements in indexing function, thus solving the problem of false dismissals due to feature-space partitioning. We have extensive experimental simulations to evaluate more performance measures than [3]. Also, in this paper, we have augmented the comparisons with other works.

To summarize, the salient contributions of our research in this paper include the following.

- 1) A composite index structure that can handle both subbody queries as well as whole-body queries.
- 2) This index structure is *generic* in the sense that it can work with different indexing functions used for projecting 3-D human motions in the feature space.
- 3) A prototype implementation handling approximately 1000 3-D human motions with 97% pruning effi-

ciency while retrieving similar motions with recall ratio of 90%.

II. RELATED WORK

In the field of biomedicine, classification and recognition of everyday activities using wearable sensors are analyzed in [4] to increase awareness of daily activity and achieve active lifestyle. A multidimensional time series motion data are decomposed into segments in [5] using the parameters of dynamic systems to make comparisons or assessments between human movements easier. An interactive framework was designed for classifying biomedical pattern data using artificial neural networks [6].

In recent years, some approaches have been proposed on motion retrievals from motion database. Qualitative features describing geometric relations between specified body points of a pose are constructed in [7], and these features are used to induce a time segmentation of motion capture data streams for motion indexing. For each query, a user has to select suitable features in order to obtain high-quality retrieval results. The authors in [8] cluster motion poses using piecewise-linear models and construct indexing structures for motion sequences according to the cluster transition trajectories through these linear components. This method is more susceptible to misclassification as similar motions may have different cluster transition signatures due to variation in performances. In addition, exact/approximate string matches may create false dismissals. Similarly, posture features of each motion frame are extracted and mapped into a multidimensional vector in [9] and [10] for motion indexing. These methods are more posture specific, and matching in two motions is carried by indexing first and last frames that may not be same for most of the similar motions, thus affecting similarity results. The authors in [11] and [12] used hierarchical motion description for a posture, and then clustering-based key-frame extraction for retrieving and compressing the motions, respectively. To extract key frames, they need to find similarity between each consecutive frame, which is time-consuming. Due to space constraints, we address few other recent approaches on motion retrievals from motion database proposed in [13]–[15]. Indexing of whole-body motions was considered in [16] using singular value decomposition (SVD), and only the first right singular vectors were considered to be feature vectors. All these approaches lack in finding similar subbody motions in varied whole-body motions.

Also, lot of work is proposed on retrieving nearest neighbors for the queries in multiattribute data repository. Due to space constraint, we address only few of them. Dynamic time warping (DTW) and longest common subsequence (LCSS) are extended for similarity measures of multiattribute data in [17]. A pivot-based index structure is proposed in [18] for combination of feature vectors to address the problem for searching in multiple feature datasets. An aggregate nearest-neighbors retrieval algorithm is proposed in [19] for the multiple query points. The iDistance [20] is a distance-based index structure; here, dataset is partitioned into clusters and transformed into lower dimension using similarity with respect to the reference point of a cluster.

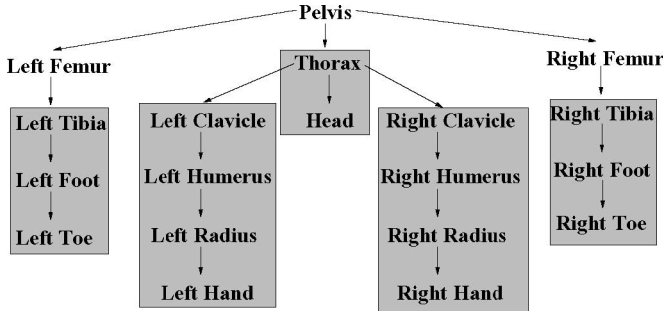


Fig. 3. Hierarchical tree structure of human body segments.

III. 3-D MOTION INDEX STRUCTURE DESIGN

A. Model for Index Structure and Query Region

The similarity of the retrieved motions with respect to the query motion depends on whether the query is resolved using “whole-body matrix” or any particular “joint motion matrix.” However, movements of the segments associated with a subbody have correlation while performing action. In other words, for leg segments, movement of the toe joint is constrained or related to the foot joint, and the movement of foot joint is constrained to the tibia joint. This implies that for retrieving similar leg motions, we first retrieve all the similar tibia motions for the given leg query. These retrieved motions are considered for finding similar foot motions. Finally, the retrieved motions that are similar to tibia and foot are considered for finding toe similarity. The final retrieved motions are similar, specifically to the leg-motion query.

This motivates us to design the index tree for each subbody part depending on the hierarchical structure of the body segments (see Fig. 3), so that we can resolve both subbody as well as whole-body motion queries efficiently. Depending on the index tree, we need to index each 3-D joint associated with the corresponding subbody. Using weighted “SVD” [21] as the indexing function, we separately index each 3-D joint matrix data for all motions into the “feature vectors.” The computed feature components for the corresponding 3-D joint matrix are the weighted aggregation of the three right singular vectors weighted by their associated normalized singular values. These feature vectors are represented/mapped as the “3-D feature points” in the “3-D feature space” for the corresponding joints, as shown in Fig. 4.

Most indexing functions in literature such as SVD [21], independent component analysis (ICA) [22], discrete Fourier transform (DFT) [23], and discrete Haar wavelet transform (DWT) [24] provide only similarity measures, i.e., they are not metrics. Due to the nonmetric characteristic of the available indexing functions, it is difficult to strictly rank similar motions for a given query. Hence, it is better to retrieve the set of motions that lie within a threshold distance from the query motion’s feature point. We refer to it as “query region.”

To get appropriate threshold distances, it becomes important to capture the uncertainty of differences in similar motions for a joint in different feature dimensions. Now, in a training database of M motions with E sets of predetermined similar motions, the maximum difference between a set of similar motions for

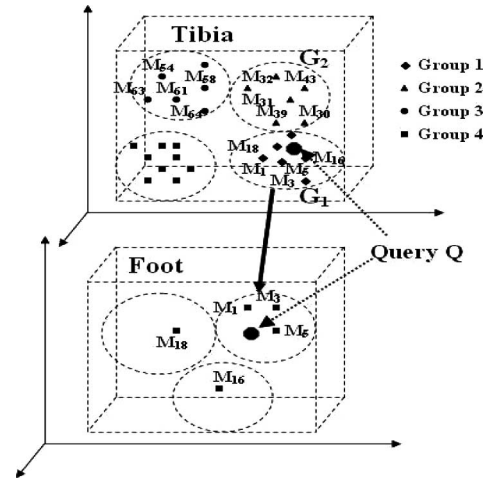


Fig. 4. Grouping of mapped feature points in feature space for body joints and single-path query resolution.

each feature component gives the variability range for the corresponding set of similar motions in a feature space. The standard deviation of these maximum differences across all sets of similar motions gives a good measure of uncertainty for each feature component. For instance, if $simDev_c$ is one such standard deviation for the c th feature component, then we get threshold δ_c for the c th dimension of the feature space as follows:

$$simTolerance_c = \delta_c = \epsilon * simDev_c \tag{1}$$

where ϵ is an input parameter used to adjust the threshold. $simTolerance_c$ (δ_c) is ultimately a threshold distance used to retrieve the set of motions that lie within this distance along the c th dimension from the query motion’s feature point. Based upon experiments with motion capture data, the practical range for ϵ is 0–1. The larger ϵ increases the threshold, and a query region is enlarged to involve more feature points, which decreases pruning efficiency.

B. Overcoming Feature-Space Partitioning Problem

When we consider the entire set of 3-D motions in the database, the crisp grouping on the feature points of one body joint gives the set of groups. Each of these groups indicates similar set of motions with respect to the corresponding joint. Several approaches in the literature can be applied for grouping the data such as hierarchical, self-organizing clustering [25], BIRCH [26], k -means via principal component analysis [27], pCluster [28], CURE [29], etc. The grouping for some tibia feature points from actual motion database is shown in Fig. 4. For a query Q of the leg segment, if a query’s tibia feature point falls in group G_1 , then the patterns present in G_1 are considered similar. According to the model of index structure discussed in Section III-A, these patterns in G_1 are further used to find similar motions to the query’s foot segment. This approach to retrieve similar motions for the given motion query has feature-space partitioning problem as follows.

- 1) Due to different variations in performing similar motions, the corresponding mapped points may fall into different groups after clustering. Hence, space partitioning of the feature points creates a problem of false dismissals in resulting output of similar motions for a query.
- 2) According to Section III-A, we retrieve the set of similar motions that lie in the query region. What happens when a query region overlaps with more than one groups?
- 3) What happens when the query point lies in between the groups?

One solution is to traverse the multiple groups that overlap the query region in order to retrieve all similar motions without any false dismissals. This multipath traversal technique is not only time-consuming but also decreases the precision level of the retrieved motions. To have a fast, accurate, no false dismissals solution using single-path traversal, we need to reorganize the memberships of some motions present in the groups after clustering, which means duplicating the similar motions from the neighboring groups in respective groups. This is equivalent to virtually extending the bounding region of a group (not physically perturbing) along each dimension and appending the original list of pattern identifiers present in a group with the list of pattern identifiers from the neighboring groups that lie within the extended region.

The virtual extension of the boundaries for a group along all dimensions to accumulate the similar motions from a neighboring group must be dependent on the threshold distances (as seen in Section III-A) along respective dimensions. Thus, to avoid false dismissals, we incorporate following rules, where d_i is the distance between the two groups and δ_i is the threshold distance along i th dimension.

Rule 1: If $\exists i : \delta_i \leq d_i < 2 * \delta_i$ and $\forall i : 0 \leq d_i < 2 * \delta_i$, then $\forall i$: extend the bounding region by $(d_i + \delta_i)$ along dimension i .

Rule 2: If $\forall i : 0 \leq d_i < \delta_i$, then $\forall i$: extend the bounding region by $(d_i + 2 * \delta_i)$ along dimension i .

Rule 3: If for any i , $d_i < 0$, then extend the bounding region by $2 * \delta_i$ along dimension i .

For instance, if we consider the group G_1 in Fig. 5 having the distance differences of $d_1 \geq \delta_1$ and $d_2 \geq \delta_2$ with the group G_2 along feature dimensions 1 and 2, then according to rule 1, we get the “overlapping region” C in group G_2 . The pattern identifiers present in region C are added in group G_1 . Similarly, for another neighboring group G_3 of the group G_1 (for simplifying the figure, we consider same d_1 and d_2), we get E as the “overlapping region” for group G_1 . This procedure is followed for all groups to get “overlapping regions” from their neighboring groups.

After collecting overlapping regions for all groups, we get no false dismissals during single-path traversal. The reason being that for any query region that overlaps with neighboring groups, for instance, as shown in Fig. 5, such query regions will always remain completely within the (shaded) overlapping regions (B and C) of both groups G_1 and G_2 . Hence, as the motions from these overlapping regions are duplicated among these groups, the nearest group to the query point or any group (in case of equidistant) selected during single-path traversal will have all

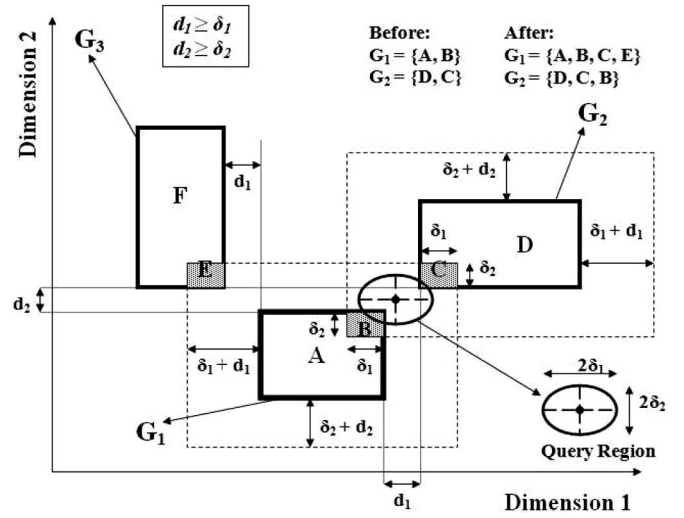


Fig. 5. Example for rule 1: the boundary extension of groups to collect similar motions from neighboring groups.

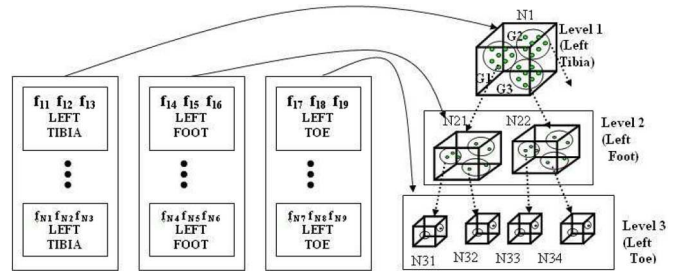


Fig. 6. Construction of hierarchical index tree for leg segments.

motions from the query region. Due to lack of space, the examples and discussions for the other rules are not discussed but they can be followed easily.

C. Construction of Subbody Index Trees

Section III-A and III-B leads us to our index structure in the form of index trees corresponding to each subbody. Fig. 6 shows the index tree for leg part of the body. Here, we follow the human body segment hierarchy to associate level 1 (i.e., the root node of the index tree) with the tibia segment while level 2 is associated with the foot segment and level 3 with the toe segment.

The number of nodes constructed in nonroot level, say level j , is equal to the total number of groups present inside the nodes of immediate parent level (i.e., level $j - 1$). Thus, each node has a parent in the form of a group in the immediate higher level. In each node of level j , joint feature vectors associated to that level are mapped in 3-D indexed space. In each nonroot node, corresponding joint feature vectors of the motions present only in the parent group are considered. In a root node, corresponding joint feature vectors of all motions are considered. A node of

the index tree has the following structure:

$$N : (G_1, G_2, \dots, G_e)$$

$$G_i : (R, S, C, \text{child} - \text{pointer}). \quad (2)$$

A node N consists of e groups G_1, \dots, G_e of mapped points formed by grouping the feature space. Each entry G_i consists of bounding region R . The variable S is a set of n pattern identifiers whose mapped feature points are present in G_i , $\text{child} - \text{pointer}$ is a pointer to the node in the next level of an index tree, and C is the centroid of a group G_i .

Similarly, other index trees are constructed to build the whole indexing structure. The (left/right) hand index tree has level 1 associated with the clavicle segment to level 4 associated with the hand segment. The possible index tree operations such as pattern insertion and search can be followed from the discussions of Section III-A–C. However, it is important to find the most relevant motions among the retrieved motions for the given motion query.

Ranking similar motions: After traversing the index tree for finding relevant motions to the query, the majority of irrelevant motions should have been pruned. To find out the most similar motions out of the retrieved relevant motions, we need to rank them in the order of similarity. According to [30], to capture the similarity of query matrix and the matrix corresponding to retrieved patterns, SVD is applied on the motion matrices. A similarity measure (Ψ) [30] can be used to compute the similarity of the query with all returned motions, and the motions with highest similarity has the highest rank or they are most similar to the query.

D. Handling Whole-Body Queries

In some cases, queries on the whole-body motions would be more meaningful than queries on subbody motions. For example, if we want to find motions similar to certain baseball pitching, motions of the whole body considered together would be more useful.

For resolving such queries, each of the five index trees returns the respective similar motions for the subbody parts. To get similar whole-body motion, we need to merge these outputs by taking the intersection of all returned pattern sets. The common patterns in all output sets from index trees will form the answer for the whole-body query. The ranking of similarity patterns is then decided by similarity measure proposed in [30].

IV. PERFORMANCE ANALYSIS

In this section, we present the performance of the indexing tree structure with respect to pruning irrelevant motions for a given motion query. We also analyze the average traversing time for the given queries through the index tree, and accuracy of indexing structure along with average recall and precision. Let N_{pr} be the number of irrelevant motions pruned for a given query by the index tree and N_{ir} be the total number of irrelevant motions in the database. We define the pruning efficiency \mathcal{P} as

$$\mathcal{P} = \frac{N_{pr}}{N_{ir}} \times 100\%. \quad (3)$$

TABLE I
PERFORMANCE PARAMETERS

| Symbol | Definition |
|------------|---|
| α | Heterogeneity Scaling Factor |
| δ_c | Threshold on c^{th} feature component |
| ϵ | Input Parameter for scaling δ_c |

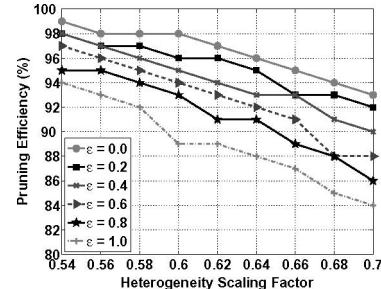


Fig. 7. Pruning efficiency for different heterogeneity scaling factors.

Our test bed consists of 200 different types of human motions performed by 20 participants in 3-D motion capture laboratory that consists of 16 high-resolution Vicon cameras recording motions at 120 frames per second. The different types of motions were included from the categories of karate, athletic movements, sports' actions, dances, etc., and each action was repeated five times, resulting into 1000 motion capture data matrices of 54 columns for each full-body motions.

A. Performance Parameters

The heterogeneity value for any body segment indicates the total variation in the corresponding feature points mapped in indexed space. α is the heterogeneity scaling factor in the range 0.5–0.9 deciding the heterogeneity threshold. For inserting patterns in different groups, the input parameter ϵ is varied between 0.2 and 1. Table I summarizes the parameters discussed in this section.

B. Pruning Efficiency

During the construction of the index trees, we used the hierarchical, self-organizing clustering approach inside nodes of all levels. We tested the pruning efficiencies by conducting a variety of experiments on different configurations of trees by varying heterogeneity scaling factor α . We also varied an input parameter ϵ while inserting motions in the index trees. All these experiments were carried out on the same test bed of 1000 motion capture data matrices. For each experiment, we issued 1000 queries to calculate the average pruning efficiency for the indexing tree, as shown in Figs. 7 and 8.

For a fixed ϵ , as we go on increasing the heterogeneity scaling factor (α), the heterogeneity threshold goes on increasing, and as a result, more pattern-corresponding feature points get accumulated in the same partition. This reduces the pruning power (as shown in Fig. 7) due to the inclusion of some irrelevant motions.

The effect on pruning efficiency was also studied by keeping heterogeneity scaling factor constant and varying the input parameter ϵ . The results shown in Fig. 8 convey that, as we

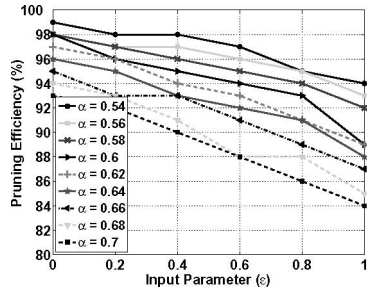
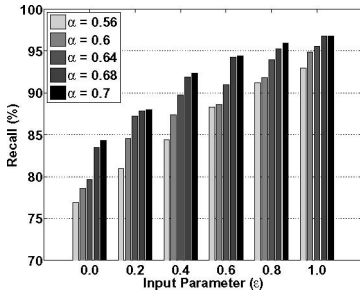
Fig. 8. Pruning efficiency for different input parameters (ϵ).

Fig. 10. Recall for different configurations of the index tree.

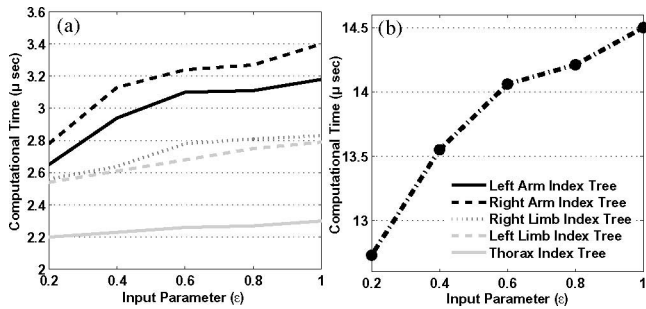


Fig. 9. (a) Computational time for all index trees. (b) Total time required for index structure for processing a query.

increase ϵ , the $simTolerance(\delta)$ for all components increases. As a result, increased extension of bounding regions for the groups gives larger “overlapping regions” in respective neighboring groups. Hence, more number of pattern identifiers from neighboring groups are duplicated inside corresponding group to decrease pruning efficiency. For a whole-body motion query, the average pruning efficiency achieved is 98% as more motions get pruned while performing intersection.

C. Computational Time for Searching Query

We tested the average computational time required for 1000 queries to traverse through the index tree and to prune the irrelevant motions. All experiments are performed on a computer with 3.00 GHz Linux Intel(R) Xeon(TM) CPU. Fig. 9(a) shows the computational time for all index trees when tested with varying ϵ . As we have pattern identifiers of all similar motions in one group, the query traversal is along only one branch of the index tree. For the whole motion query, we need to traverse all the five index trees. Fig. 9(b) shows the aggregate time required for the whole index structure to prune the irrelevant motions for the whole-body motion query.

D. Recall and Precision

When a motion query is given to an index tree, ideally, it should return all similar motions. However, in practice, resolution of some queries may have some irrelevant motions due to the use of nonmetric nature of the similarity measures (as discussed in Section III-A) as well as the variations in performing similar motions. Fig. 10 shows the average recall of 1000 motion queries for each different configurations of the index tree. When ϵ is in range of 0.6–1.0, about 95% of relevant motions are re-

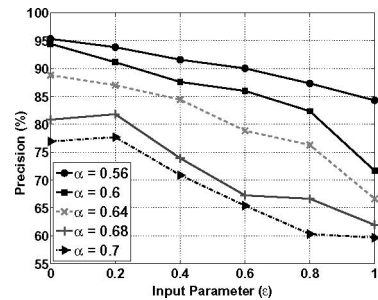


Fig. 11. Precision for different configurations of the index tree.

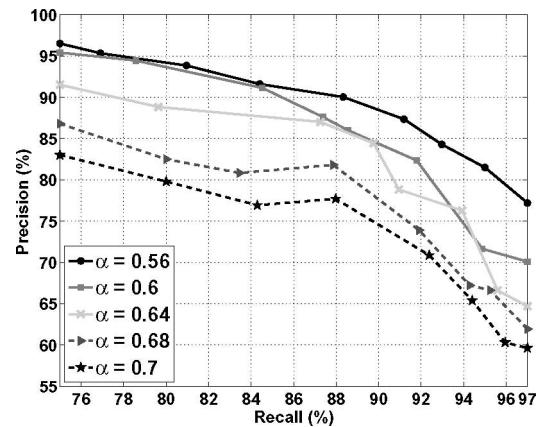


Fig. 12. Precision as a function of recall, parameterized by the heterogeneity factor.

turned for the motion query for greater values of heterogeneity scaling factors.

Precision: Fig. 11 shows the results of the precision of retrieval motions for different configurations of the index structure. We issued 1000 queries for every configuration of the index structure. For $\epsilon = 0.2$, the precision of retrieval achieved was 94%. As heterogeneity scaling factor increases, the amount of retrieved patterns increases, and as a result, there are more false hits in the retrieval that lowers precision.

Fig. 12 shows the precision versus recall curves parameterized by the different heterogeneity scaling factors. It is clear that, for a given recall value, the precision of the indexing structure for lower heterogeneity scaling factors is better than that for higher scaling factors.

E. Comparisons

We compared our approach with:

- 1) indexing time series technique applied to motion capture data in [31];
- 2) multilevel distance-based index structure proposed in [32].

A technique for indexing time series with invariance to uniform or global scaling based on bounding envelopes was introduced in [31]. It was applied on motion capture database to search the closest match for the query motion with pruning efficiency of 90%–95%. The limitation of this approach is that query can only be in the form of single time series and the candidates too are univariate time subsequences with fixed length. To find similar motion for the given query, we need to apply searching algorithm sequentially to every degree of freedom and at the end evaluate the best match by measuring the weighted sum of Euclidean distance of the sequences, which is time-consuming. Hence, the fastest average running time per query achieved in [31] was 10–20 ms, which is significantly lower than our approach, typically around 15 μ s.

The multilevel distance-based index structure for extended frobenius norm (MUSE) indexing structure works on indexing synthetic hand gesture motions generated using an instrumented device called CyberGlove [32]. Since hand gesture motions have multiattributes and different variations just like captured 3-D human motions, MUSE seems to be the most suitable indexing structure published so far for multiattribute motion data. For performance comparison, we applied MUSE on our database of 3-D human motions. The MUSE structure was constructed using three levels. By querying 1000 3-D motions, the pruning efficiency achieved was 5.5%, as compared to 97% of our index tree structure. Also, the average computational time required per query was 0.3 s. The reason MUSE has a poor performance on 3-D motions is that the lower bound defined by MUSE for pruning irrelevant motions [32] is not tight enough for 3-D human motions.

V. CONCLUSION AND DISCUSSIONS

In this paper, we considered content-based motion querying on a repository of multiattribute 3-D motion capture data. Our objective was to develop an indexing structure that can efficiently handle queries on subbody motions as well as whole-body motions. We observed that the similarity measure of a motion of a body subpart (such as leg motion) might be heavily influenced by the motion of another part (such as hand), if the entire human body motion is considered as a whole. Hence, we argued that the resolution of subbody motion queries could be handled better by comparing the isolated motions of the relevant human body joints. Based on these arguments, we proposed a composite index structure that maps on to the hierarchical segment structure of the human body.

In our prototype implementation, we used SVD as the technique to map the multidimensional 3-D motion data onto the feature space. However, the proposed index structure can easily work with any other appropriate similarity matching approaches as well. We also used the hierarchical, self-organizing approach for grouping similar motions in the prototype. Again,

any other appropriate grouping/clustering approach can be incorporated into the implementation. We tested our prototype using a database of 1000 human motions captured in our Motion Capture Laboratory. Our experiments show that up to 96%–97% irrelevant motions can be pruned for any kind of motion query while retrieving all similar motions, and one traversal of the index structure through all index trees takes on an average 15 μ s.

This research will be used to develop a user-friendly query interface that can assist user to give specific queries for subbody motions as well as generic queries on whole-body motions in order to retrieve similar motions from the motion database depending on the query. Since the proposed indexing approach helps in analyzing the different body segments/joints together or separately for different motions, this approach can be utilized rigorously for biomedical applications. Also, physical medicines and rehabilitation specialists can use this method for diagnosis purpose.

Though our proposed indexing structure is based on the joint/bone structure of the human motion data, it would be applicable to other forms of multidimensional medical data with hierarchical relations among the attributes.

REFERENCES

- [1] C. G. Song, J. Y. Kim, and N. G. Kim, "A new postural balance control system for rehabilitation training based on virtual cycling," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 2, pp. 200–207, Jun. 2004.
- [2] S. Hermann, "Exploring sitting posture and discomfort using nonlinear analysis methods," *IEEE Trans. Inf. Technol. Biomed.*, vol. 9, no. 3, pp. 392–401, Sep. 2005.
- [3] G. N. Pradhan, C. Li, and B. Prabhakaran, "Hierarchical indexing structure for 3D human motions," in *Proc. 2007 ACM Multimedia Model. Int. Conf.*, Jan., pp. 386–396.
- [4] J. Parkka, M. Ermes, P. Korpiainen, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 119–128, Jan. 2006.
- [5] C. Lu and N. J. Ferrier, "Automated analysis of repetitive joint motion," *IEEE Trans. Inf. Technol. Biomed.*, vol. 7, no. 4, pp. 263–273, Dec. 2003.
- [6] H. Wang, F. Azaue, and N. Black, "An integrative and interactive framework for improving biomedical pattern discovery and visualization," *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 1, pp. 16–27, Mar. 2004.
- [7] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Trans. Graph. (TOG)*, vol. 24, pp. 677–685, 2005.
- [8] G. Liu, J. Zhang, W. Wang, and L. McMillan, "A system for analyzing and indexing human-motion databases," in *Proc. 2005 ACM SIGMOD Int. Conf. Manage. Data*, pp. 924–926.
- [9] S.-P. Chao, C.-Y. Chiu, J.-H. Chao, Y.-C. Ruan, and S.-N. Yang, "Motion retrieval and synthesis based on posture features indexing," in *Proc. 5th Int. Conf. Comput. Intell. Multimedia Appl.*, Sep. 2003, pp. 266–271.
- [10] C.-Y. Chiu, S.-P. Chao, M.-Y. Wu, S.-N. Yang, and H.-C. Lin, "Content-based retrieval for human motion data," *J. Vis. Commun. Image Representation*, vol. 15, pp. 446–466, Apr. 2004.
- [11] F. Liu, Y. Zhuang, F. Wu, and Y. Pan, "3D motion retrieval with motion index tree," *Comput. Vis. Image Understanding*, vol. 92, pp. 265–284, Jun. 2003.
- [12] Q. Gu, J. Peng, and Z. Deng, "Compression of human motion capture data using motion pattern indexing," *Comput. Graph. Forum*, vol. 28, no. 1, pp. 1–12, 2009.
- [13] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.
- [14] K. Tang, H. Leung, T. Komura, and H. P. H. Shum, "Finding repetitive patterns in 3D human motion captured data," in *Proc. 2nd Int. Conf. Ubiquitous Inf. Manage. Commun.*, Suwon, Korea, 2008, pp. 396–403.

- [15] X. Wang, Z. Yu, and H. Wong, "3D motion sequence retrieval based on data distribution," in *Proc. 2008 IEEE Int. Conf. Multimedia Expo.*, 2009, pp. 1229–1232.
- [16] C. Li, G. Pradhan, S. Zheng, and B. Prabhakaran, "Indexing of variable length multi-attribute motion data," in *Proc. ACM MMDB*, Washington, DC, Nov. 2004, pp. 75–84.
- [17] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," in *Proc. SIGMOD*, Aug. 2003, pp. 216–225.
- [18] B. Bustos, D. Keim, and T. Schreck, "A pivot-based index structure for combination of feature vectors," in *Proc. SAC 2005*, New York, pp. 1180–1184.
- [19] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui, "Aggregate nearest neighbor queries in spatial databases," *ACM Trans. Database Syst.*, vol. 30, no. 2, pp. 529–576, 2005.
- [20] C. Yu, B. C. Ooi, K.-L. Tan, and H. V. Jagadish, "Indexing the distance: An efficient method to KNN processing," in *Proc. VLDB*, San Francisco, CA, 2001, pp. 421–430.
- [21] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
- [22] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Hoboken, NJ: Wiley, 2001.
- [23] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," in *Proc. 4th Conf. Found. Data Org. Algorithms*, Oct. 1993, pp. 69–84.
- [24] K.-P. Chan and A. W.-C. Fu, "Efficient time series matching by wavelets," in *Proc. 15th ICDE*, Mar. 1999, pp. 126–133.
- [25] F. Luo, L. Khan, F. Bastani, I.-L. Yen, and J. Zhou, "A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles," *Bioinformatics*, vol. 20, pp. 2605–2617, May 2004.
- [26] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD*, New York, 1996, pp. 103–114.
- [27] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. 21st ICML*, New York, 2004, pp. 225–232.
- [28] H. Wang, W. Wang, J. Yang, and P. S. Yu, "Clustering by pattern similarity in large data sets," in *Proc. ACM SIGMOD*, New York, 2002, pp. 394–405.
- [29] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," in *Proc. ACM SIGMOD*, New York, 1998, pp. 73–84.
- [30] C. Li and B. Prabhakaran, "A similarity measure for motion stream segmentation and recognition," in *Proc. 6th Int. Workshop Multimedia Data Mining*, Chicago, IL, Aug. 2005, pp. 89–94.
- [31] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle, "Indexing large human-motion databases," in *Proc. 30th VLDB Conf.*, Toronto, ON, Canada, 2004, pp. 780–791.
- [32] K. Yang and C. Shahabi, "Multilevel distance-based index structure for multivariate time series," in *Proc. 12th Int. Symp. on Temporal Representation and Reasoning*, Washington, DC, Jun. 2005, pp. 65–73.



Gaurav N. Pradhan received the Ph.D. degree in computer science from the University of Texas at Dallas, Richardson, in 2008.

He is currently a Postdoctoral Research Associate in the Research Center of Cognitive Ubiquitous Computing, Arizona State University, Tempe. His current research interests include data analysis and mining in biomedical sensor databases, indexing, pattern recognition, classification, content-based similarity retrieval, clustering analysis in multiple, high-dimensional databases, and ubiquitous computing in assistive and rehabilitative environments.



Balakrishnan Prabhakaran (M'99) received the Ph.D. degree in computer science from the Indian Institute of Technology Madras, Chennai, India.

He is currently an Associate Professor with the Department of Computer Science, University of Texas at Dallas, Richardson. He has been involved in the area of multimedia systems: animation and multimedia databases, authoring and presentation, resource management, and scalable Web-based multimedia presentation servers. He was a Visiting Research Faculty with the Department of Computer Science, University of Maryland, College Park. He was also a faculty member in the Department of Computer Science, National University of Singapore, as well as in the Indian Institute of Technology Madras.