A System for Analyzing and Indexing Human-Motion Databases

Guodong Liu, Jingdan Zhang, Wei Wang, Leonard McMillan
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175, USA
{liug, zhangjd, weiwang, mcmillan}@cs.unc.edu

ABSTRACT
We demonstrate a data-driven approach for representing, compressing, and indexing human-motion databases. Our modeling approach is based on piecewise-linear components that are determined via a divisive clustering method. Selection of the appropriate linear model is determined automatically via a classifier using a subspace of the most significant, or principle features (markers). We show that, after offline training, our model can accurately estimate and classify human motions. We can also construct indexing structures for motion sequences according to their transition trajectories through these linear components. Our method not only provides indices for whole and/or partial motion sequences, but also serves as a compressed representation for the entire motion database. Our method also tends to be immune to temporal variations, and thus avoids the expense of time-warping.

Categories and Subject Descriptors
I.2.10 [Vision and Scene Understanding]: Motion; H.3.1 [Content Analysis and Indexing]: Indexing Methods.

Keywords
Motion capture, piecewise linear modeling, motion compression, motion database indexing.

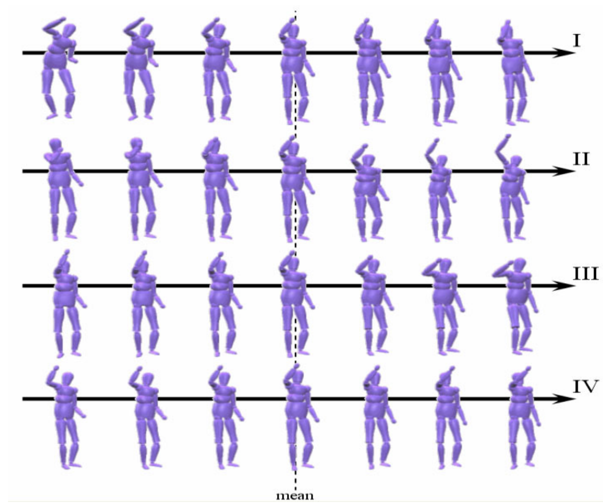

Figure 1. *Above we illustrate the range of motions typical of a single pose cluster, or local linear model. A mean pose, shown as the center of each progression and four orthogonal axes of variation specify this model. The range of motion along each axis is illustrated in both directions from the mean.*

1. INTRODUCTION
Motion capture, or mocap, is an important new technique for capturing and analyzing human articulations. At present, mocap is widely used to animate computer graphics figures in motion pictures and video games. Mocap is also used for analyzing and perfecting the sequencing mechanics of premiere athletes, as well as monitoring the recovery progress of physical therapies. Huge collections of mocap data are rapidly coming available and there is an immediate need for tools that index, query, compress, annotate, and organize these datasets.

Our approach to this problem is to build an implicit model of every distinct body pose seen in a motion database, and to cluster these poses into groups that can be effectively interpolated using simple linear models. Motion sequences can then be classified and indexed by the trajectories that they take through the set of pose clusters. The linear models also significantly reduce the storage requirements of the database.

A common form of motion capture uses optical sensing of strategically placed targets, known as *markers*. These markers are placed at specific positions usually at, or near, boney regions. Mocap uses triangulation from multiple cameras to estimate the 3D position of each marker. Raw marker data exhibits considerable redundancy that is attributable to a variety of real-world constraints, including kinematic, dynamic, and neurophysical limitations. Our approach exploits this redundancy by using simple models to describe groups of similar poses.

Our modeling approach partitions all pose instances in the motion database into a hierarchy of low-dimensional local linear models. We represent data sequences by their transitions through these local linear models, which we call a *cluster transition signature*. This signature acts as a simplified representation of the entire sequence, and it is used for inter-sequence comparisons and sequence indexing. The resulting indexing structure also supports variable error tolerances in the matching process, which provides more flexibility and control.

SIGMOD 2005 June 14-16, 2005, Baltimore, Maryland, USA

2. PREVIOUS WORK

Many methods have been applied to human-motion modeling [7][6][2][9]. Motion indexing [5] is an important step in creating motion databases, supporting queries and other motion database manipulations.

Piecewise linear regression [1][4][8][10] has been used for approximating models to within a specified error tolerance using a collection of local linear models. Our modeling approach is of this class.

Principal Feature Analysis (PFA) [3] is a method to select the most informative subset of features. In our work, we use PFA to identify principal markers, and employ these markers as parameters in our model.

3. PRINCIPAL MARKER SELECTION

The first step of our model construction process determines the subset of database features used to construct the local linear model classifier. PFA offers a way to systematically determine a subset of features that characterize the remaining features to within a specified tolerance [3]. We call this subset the *principal features*.

A body pose is specified by a set of marker positions, each with three spatial coordinates (*x, y, z*). In our case, the set is composed of 41 markers representing each pose. A motion sequence is a time-series of such poses. We normalize each pose by translating a common marker to the origin, and rotate the model so that each pose has its left-to-right and head-to-toe orientations along specific axes. We treat each normalized pose as a single high dimension vector. In this first stage, we identify a subset of markers that best explain the variability seen in the database. We have adapted PFA to select this principal marker set as follows: We treat motion-capture data as an *f-by-N* matrix, $\boldsymbol{D}$, where f is the number of features (markers times 3) and N is the number of poses. Let $\widetilde{\mathbf{D}} = \mathbf{A}_q\mathbf{W}$ represent an approximation of $\boldsymbol{D}$ where $\boldsymbol{A}_q$ is the matrix of the first q principal components deduced from Principal Component Analysis (PCA), and $\boldsymbol{W}$ is the weight matrix of the data points in the principal component space. Let $\boldsymbol{V}_i \in R^q$, $i=1:f$ be the rows of $\boldsymbol{A}_q$. If two features i and j are highly correlated, they will have similar absolute weights, i.e. $|V_i| \approx |V_j|$.

We first select a sufficiently small set of leading eigenvectors sufficient to satisfy a desired reconstruction error tolerance. Next, K-means clustering is performed on the set of absolute-weight vectors with K slightly greater than the number of the leading eigenvectors. Finally, we apply a heuristic to weight the importance of the three features from each marker and select a minimal set of the most important markers satisfying a cover of all clusters generated in the K-means clustering step described above.

4. PIECEWISE-LINEAR MODELING

We next construct a piecewise-linear model using a divisive clustering approach that, at each level, attempts to obtain a best-fit linear model of a user-specified dimension, d, and error tolerance, ε. This best-fit model is constructed via successive applications of PCA. If, at any level in the hierarchy, the variance of all data points in a cluster can be explained by a d-dimensional linear model to within ε, the cluster is considered a leaf in the modeling hierarchy and a local-linear mapping function is computed for the cluster. Otherwise, the cluster is split into two children clusters by performing a K-means clustering, with K = 2, based on the subset of principal features found in Section 3. The centroids found during this process are saved for later use as a decision tree classifier that maps input poses to their appropriate local linear model. We adopt a fuzzy partitioning scheme that distributes data points near the decision boundary to both child clusters, thus improving the reconstruction continuity at cluster transitions. This is particularly important for human motion modeling because we are very sensitive to these discontinuities, which are seen as jerkiness. The splitting process continues until all clusters satisfy the desired dimensionality constraint and error tolerance. Given a feature vector of principal markers $\boldsymbol{s}$, a classifier is used to find the appropriate cluster. Then the vector of the non-principal marker features $\boldsymbol{S}$ is estimated by a linear-mapping function

$$\boldsymbol{S} = \boldsymbol{T}\,\boldsymbol{s} \qquad (1)$$

where $\boldsymbol{T}$ is a least-squares mapping function computed using the training set.

5. APPLICATIONS

Our model is useful for many human motion related modeling and searching applications. In the following subsections, we will describe its utility in motion compression, estimating human motion from a reduced marker set, and motion database indexing.

5.1 Motion Compression

Our piecewise-linear modeling approach provides a compact model of the entire database accurate to within a defined error tolerance. Given $\boldsymbol{N}$ poses, each with $\boldsymbol{f}$ features, we construct a model with $\boldsymbol{k}$-dimensional piecewise-linear components. The original data points in each cluster are accurately represented by the cluster mean from one of $\boldsymbol{C}$ clusters, and the projection of each pose onto its most significant $\boldsymbol{k}$ eigenvectors. The resulting size of the entire reduced dataset would be $\boldsymbol{Cfk} + \boldsymbol{Nk}$, which represents a considerable compression of the original database size, $\boldsymbol{Nf}$. Typical values for $\boldsymbol{C}$, $\boldsymbol{f}$, $\boldsymbol{k}$, and $\boldsymbol{N}$, are 8,000, 62, 4, and 2.5M respectively.

5.2 Estimating Human Motions from a Reduced Marker Set

Motions that were not a part of the training set can also be estimated with our modeling approach. Moreover, we require only a reduced marker set, the principal markers, to estimate the unseen pose. Given a configuration of principal markers, we first find the most appropriate locally linear model, i.e. cluster, using the decision tree constructed during the model's construction. We estimate the remaining markers using the linear mapping function (Equation 1) associated with the cluster.

5.3 Motion Database Indexing

Our model also provides effective indexing of a motion database. Different motion sequences typically reside in distinct subsets of the clusters and transit among those clusters with distinctive trajectories. We treat the cluster transition trajectory of each motion as its signature and use it for indexing and searching the motion database. We represent the cluster transition trajectory of a motion sequence by the IDs of the clusters that it transits through.

Given a list of pose-cluster indices for a motion sequence, we first divide the index list into a number of segments, with each segment sharing the same cluster index. We then collapse runs of the same cluster to form the cluster transition signature for all motion sequences in the database.

In order to query the motion database for the k-nearest subsequences resembling a specified query motion sequence, we first compute the query sequence's cluster transition signature and search for exact and approximate string matches between cluster transition signatures.

6. DEMONSTRATION

We demonstrate the utility of our modeling approach for motion compression, modeling unseen motions, and motion database indexing. The motion data used in our experiments comes from Carnegie Mellon University's Graphics Lab motion-capture database available at http://mocap.cs.cmu.edu.

Four data sets are used in the demonstration. The first two are of single-subjects undergoing varied complex motions. The third set is composed of multiple subjects all engaged in various walking motions. The first single-subject data set is composed of eight motion sequences— including walking, punching, running, bending, washing, and three sword-play sequences. The second single-subject dataset has 57 motion sequences with many lower-body movements, such as walking, running, and jumping. The multi-subject data set has 57 different, and widely varying, walking sequences from 7 different subjects. We use all three data set to build a model for demonstrating motion compression and motion database indexing. However, in demonstrating motion estimation, we used 80% of data for training and held out the remaining 20% of the data for testing. The fourth data set includes all the data from CMU's Graphics Lab motion-capture database, with totally over 2.5M frames from 1,600 motion sequences. This dataset is used for motion database indexing demonstration.

After training, each frame is compressed by retaining only the projections of each pose onto the eigenvectors of their assigned clusters. These can be used to recover an approximation to any pose in the original database, and, the reconstruction error is to within the preset error tolerance used when specifying the model. Our demonstration shows that the recovered motions from the compressed data alongside the original motions.

When given new motion sequences, not used in the training set, we are able to estimate accurate full-body poses using only the set of principal markers. We demonstrate our model's ability to synthesize these new motions by displaying estimated motions alongside the original sequences. Figure 2 shows a snapshot from our motion viewer estimating an unseen sequence. The reconstructed motion is both metrically and visually accurate.

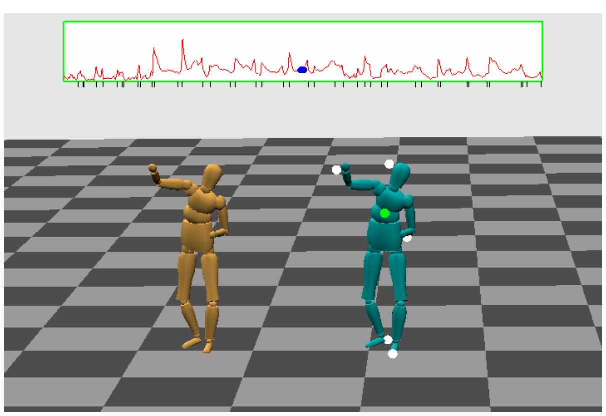

Figure 2. *The gold figure represents actual pose data. The cyan figure shows our model's pose estimate based on the principal markers, which are shown as white disks. The green disk indicates the origin marker. An RMS error meter for the entire marker set appears above with a full-scale value of 50mm/marker. The black tick marks denote cluster transitions.*

To demonstrate motion indexing, we use queries composed of motion subsequences not in the database. Our demonstration shows the query motion sequences alongside with the k most similar motion sequences returned. The response time of the query is interactive and the results are accurate, thus illustrating that our model is an effective index of the motion database.

7. CONCLUSIONS

We have presented a piecewise-linear modeling approach to human motion data parameterized by a subset of principal markers. Our method depends on a database of motion capture data, from which it automatically determines a set of principal markers, and then constructs a hierarchical model. The processes of model construction and data query are efficient and scale well with data size and dimensionality. We have demonstrated our model's ability to predict the complete configuration of a human model based on a subset of motion capture information as well as compressing and indexing motion databases.

8. ACKNOWLEDGMENTS

Our thanks to Carnegie Mellon University's Graphics Lab for allowing us to use their Motion-capture database, which was supported with funding from NSF EIA-0196217.

REFERENCES

[1] Atkeson C., Moore A., and Schaal S. Locally weighted learning. *Artificial Intelligence Review*, 11(4):76–113, 1997.

[2] Brand M., and Herzmann A. Style machines. *In Proceedings of ACM SIGGRAPH00*, 183–92, 2000.

[3] Cohen I., Tian Q., Zhou X. S., and Huang T. S. Feature selection using principal feature analysis, *ICIP'02*.

[4] Ferrari-Trecate G., and Muselli M. A new learning method for piecewise linear regression. *ICANN*, 444-449, 2002.

[5] Keogh E., Palpanas T., Zordan V. B., Gunopulos D, and CARDLE M. Indexing large human-motion databases, *Proceedings of the 30th VLDB Conference*, pp. 780-791, Toronto, Canada, 2004.

[6] Li Y., Wang T., and Shum H. Y. Motion texture: a two-level statistical model for character motion synthesis. *ACM Transactions on Graphics,* 21(3):465-472, July 2002.

[7] Rosales R., and Sclaroff S. Learning and synthesizing human body motion and posture, *In The 4th International Conf. on Automatic Face and Gesture Recognition,* 506-511, grenoble, France, March 2000.

[8] Torgo L., and Da Costa J. Clustered partial linear regression. *Machine Learning, 50(3):* 303-319, 2003.

[9] Semwal S., Highttower S., Highttower R., and Stansfield S. Mapping algorithms for real-time control of an avatar using eight sensors, *Presence 7*, 1, 1-21.

[10] Vijaykumar S., and Schaal S. Locally weighted projection regression: incremental real time learning in high dimensional space. *ICML,* 1079-1086, 2000.