FMDistance: A fast and effective distance function for motion capture data

Kensuke Onuma[1], Christos Faloutsos[2] and Jessica K. Hodgins[2]

[1]Sony Corporation, Tokyo, Japan [2]Carnegie Mellon University, Pittsburgh, PA, U.S.A

Abstract

Given several motion capture sequences, of similar (but not identical) length, what is a good distance function? We want to find similar sequences, to spot outliers, to create clusters, and to visualize the (large) set of motion capture sequences at our disposal. We propose a set of new features for motion capture sequences. We experiment with numerous variations (112 feature-sets in total, using variations of weights, logarithms, dimensionality reduction), and we show that the appropriate combination leads to near-perfect classification on a database of 226 actions with twelve different categories, and it enables visualization of the whole database as well as outlier detection.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Graphics data structures and data types I.3.5 [Computer Graphics]: Physically based modeling H.2.8 [Database Management]: Data mining

1. Introduction

Motion capture data is often used to create human animations for video games, movies and other applications. Large databases of motion now exist both on the web (see http://mocap.cs.cmu.edu/, for example) and as proprietary resources within entertainment companies. These databases are not easily searchable. In this paper, we present a distance function that represents the characteristics of the actions in a motion capture sequence. This distance metric is suitable for classifying motions, searching for similar motions, and for detecting some classes of outlying motions.

We would like a distance function that will satisfy two requirements: *speed* and *effectiveness*. The distance function should be fast to compute, even on long motion capture sequences. Ideally, it should be independent ($O(1)$) on the length N_{frame} of the sequences. The distance function should be also meaningful, so that it is useful for clustering, classification, and anomaly detection (see Figure 1), where the usual distance functions for motion capture sequences do not work well.

Specifically, we propose *FMDistance*, a method which is *independent* on the sequence length, and only depends on K, the number of joint angles we track. Our idea is to calculate the approximation of the total kinetic energy of each joint as a preprocessing step, thus compressing each motion capture sequence of $K \times N_{frame}$ numbers into K numbers (and possibly, even fewer, if we do dimensionality reduction). The proposed method is also effective, as we illustrate in Section 4.

The rest of the paper is organized as follows: we first re-

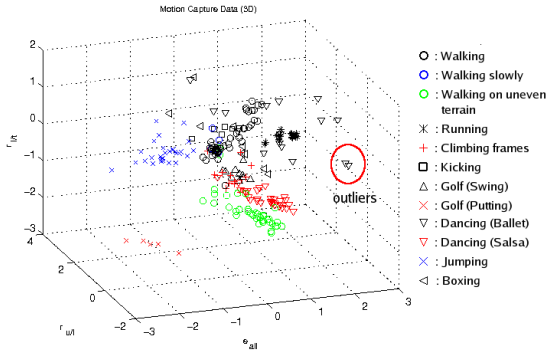


Figure 1: *Visualization of classification by three new features. Notice the* homophily*: similar motions cluster together; also notice that we can visually spot outliers, like ballet dancing which has noisy capture frames, in the red circle.*

view the related work in Section 2; the proposed strategies are presented in Section 3; the experimental results are presented in Section 4; and we conclude the paper in Section 5.

2. Related work

To our knowledge, there is little work in computer graphics that focuses on a distance function for expressing the characteristics of a motion capture sequence. The work of Ren et al. [RPE*05], who explored methods for verifying the naturalness of a motion capture sequence, is close to our problem. Troje [Tro02] investigated gender classification of walking motions by analyzing motion capture data. Researchers in computer vision also proposed distance functions for human activity classification [HJB*05].

Many distance functions have been proposed for finding candidates of "frame to frame" transition [KGP02] or indexing for segmentation [KPZ*04]. However, the distance function in [KGP02] requires that the two motion capture sequences are *exactly* the same length, and even then, both require $K \times N_{frame}$ computations; Our proposed method does not depend on N_{frame} and is much simpler and faster ($O(K)$).

3. Proposed method

In this section, we describe details of *FMDistance* for the classification of motion capture data.

3.1. Preprocessing step : Transformation from motion capture data to kinetic energy-based parameters

We assume that motion capture data have K DOF in total and a series of motion capture data for one action has N_{frame} frames. Specifically, every frame has joint angles, the root orientation and the root positions (coordinates of the root). A set of motion capture data is denoted by $\{\vec{x}_i | i = 1, ..., N_{frame}\}$. Each frame $\vec{x}_i = \left[x_{i,1}, x_{i,2}, ..., x_{i,K}\right]^T$ represents a point in K dimensional space.

The main idea is to use the average of the approximate kinetic energy of each joint angle (or carefully selected groups of joints), as features. This way, the $K \times N_{frame}$ numbers of a motion capture sequence are condensed into at most K values as a preprocessing step, achieving our first goal, speed. As we show in Section 4, the second goal, effectiveness, is also achieved.

We compute the approximate kinetic energy as the sum of squares of velocities. Specifically, the velocity of the root and the angular velocity of each joint, $\vec{v}_i = \left[v_{i,1}, v_{i,2}, ..., v_{i,K-1}\right]^T$ are calculated by the first derivatives: $\vec{v}_i = \frac{\vec{x}_{i+1} - \vec{x}_i}{t_{frame}}$ where t_{frame} is the period between frames. For the root velocity, we calculate the velocity and the energy across the plane from the x + z position. Thus, the dimensionality of $\vec{v}_i$ decreases by 1, to $K-1$. We note that the vertical velocity are included as one of parameters.

To compute the kinetic energy, we also need to consider the moment of inertia m_j of each joint, (and body mass, for each position-coordinate). The kinetic energy $E_{i,j}$ of joint j at time i is

$$E_{i,j} = m_j \times v_{i,j}^2 \tag{1}$$

Although the moment of inertia varies depending on the body part, we assume that they are constant with respect to time.

3.2. Distance function for classification

We calculate the mean of the kinetic energy at each dimension (joint angle/position/orientation), j:

$$\overline{E}_j = \frac{1}{N_{frame}} \sum_{i=1}^{N_{frame}} E_{i,j} \tag{2}$$

The kinetic energy is bursty: some joints have a high kinetic energy while others do not. We propose to treat the burstiness, by taking logarithms, specifically $\log(x+1)$ (to handle the joints of zero energy). Thus:

$$e_j = \log\left(\overline{E}_j + 1\right) \tag{3}$$

The vector $\vec{e} = [e_1, e_2, ..., e_{K-1}]^T$ is our proposed feature vector. Then, the distance between two motion capture sequences N and M is the Euclidean distance of their feature vectors $\vec{e_N}$ and $\vec{e_M}$.

3.3. Dimensionality reduction and visualization

The features described in the Section 3.2 have more than three dimensionality. For visualization, we have to reduce the dimensionality to three. Although it is common to use principal component analysis (PCA) ([Jol86]) in order to reduce dimensionality for classification, we propose a more intuitive method.

The intuition is that different motions will exercise different body parts: for example, "walking" will have a balance between upper body and lower body, while "golf swing" will have more energy on the upper body. We propose to capture these differences with new features, namely, the ratio r of the approximate kinetic energy of groups of body parts.

First we sum up the average of the approximate kinetic energy of the joint rotation in order to estimate the kinetic energy of the whole body, upper body, lower body, limbs and trunk.

$$\overline{E}_{parts} = \sum_{j \in parts} \overline{E}_j \tag{4}$$

Then we take the ratio of the kinetic energy between symmetrical body parts. As in Equation (3), we also use the $\log(x+1)$ transform:

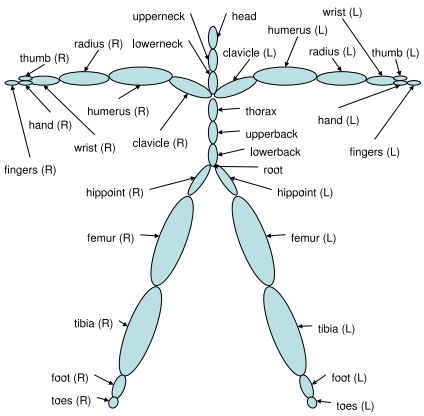
$$e_{all} = \log\left(\overline{E}_{total}+1\right),\ r_{u/l} = \log\left(\frac{\overline{E}_{upper}+1}{\overline{E}_{lower}+1}\right),\ r_{l/t} = \log\left(\frac{\overline{E}_{limbs}+1}{\overline{E}_{trunk}+1}\right) \tag{5}$$

We propose to use the 3-d vector $\vec{\beta} = [e_{all}, r_{u/l}, r_{l/t}]^T$ as a feature vector of a motion capture sequence, and, again, we use the corresponding Euclidean distance between two such feature vectors β_{Ni}, β_{Mi}.

4. Experimental results

In this section, we evaluate the effectiveness of our approach.

The motion capture data we use for the experiments is http://mocap.cs.cmu.edu/. Figure 2 shows the human figure, the number of DOF, and the value of m_j for each joint. The value of m_j tries to reflect the moment of inertia: hip joints get high values, shoulders get a bit smaller, knees are next, elbows are next, etc.



bone	DOF	mj
root (pos)	3	1
root (rotation)	3	5
hippoint	0	0
lowerback	3	4
upperback	3	4
thorax	3	4
lowerneck	3	3
upperneck	3	3
head	3	0
clavicle	2	3
humerus	3	3
radius	1	2
wrist	1	1
hand	2	0
fingers	1	0
thumb	2	0
femur	3	3
tibia	1	3
foot	2	0
toes	1	0

Figure 2: *Human figure model, DOFs and the distribution of m_j. It has 29 joints, 56 joint angles, three angles for the root orientation and three position coordinates for the root*

We used 226 sequences of motion capture data and they are categorized into twelve actions described in Figure 3. We also carefully examined 112 feature-sets, from the cross products of [{log, lin} $\times$ {original data, transformed data from quaternion} $\times$ {weighted m_j, constant m_j} $\times$ {normalized, unnormalized} $\times$ {seven feature-sets}].

Implementation details: The distance function for each feature-set was the Euclidean distance. Before calculating velocities, we followed standard practice and removed noise with a linear low-pass filter spanning five frames. Finally, when we normalized the data in order that they have zero mean and unit standard deviation.

Effectiveness measure: To measure the effectiveness of our feature-sets, we use the classification accuracy, and specifically, a 1-nearest neighbor (1-NN) classifier. We chose this classifier because its accuracy is directly related to the effectiveness of the feature-set, and it needs no training. Moreover, as we show next, it gives excellent classification accuracy.

4.1. Accuracy of various feature-sets

We use the feature-sets described in Section 3, as well as some other, *simpler* feature-sets, for comparison. The nomenclature for a feature set is as follows: For example, *61-LOG-cons* stands for 61 features, with the log transform, and constant values for the moments/weights m_j. Similarly, *61-LIN-est* stands for the same 61 features, without the log transform, and with the estimated values of the m_j weights, as shown in Figure 2.

61-LOG Log of the mean of the approximate kinetic energy for each joint angle. See Equation (3).

61-LIN (for comparison) Mean of the approximate kinetic energy. See Equation (2).

31-LOG (for comparison) This feature-set contains the root's energy in the horizontal and vertical direction, as well as and the approximate kinetic energy of each joint.

62-POS (for comparison) This feature-set consists of the mean of $\vec{x}_i$ with respect to time.

Table 1 shows the most interesting results of our experiments.

features	m_j	% error
61-LOG-cons	constant	**2.21%**
61-LOG-est	estimated	2.65%
61-LIN-cons	constant	3.98%
61-LIN-est	estimated	4.42%
31-LOG-cons	constant	3.10%
31-LOG-est	estimated	3.10%
62-POS	N/A	5.75%

Table 1: *Classification accuracy for several feature sets. The winner, 61-LOG-cons, is in bold. Results of other methods are worse, and omitted for space.*

Table 1 shows that 61-LOG-cons is the best feature-set for classifying motion capture data. The full table has 112 methods in total, but we omit the lower-performing methods, for space (see [OFH07]). We report the conclusions and observations:

- Taking the logarithms ($\log(x+1)$) always improves performance. For example, see 61-LOG vs. 61-LIN.
- There is no clear winner with respect to the sets of m_j weights. Both competitors ("cons", and "est") perform about the same.

Figure 3 shows the *confusion matrix* for our best performer, the *61-LOG-cons* feature-set: Columns correspond to the predicted labels, and rows to the actual label. In a perfect classifier, the matrix would be diagonal. Notice that *61-LOG-cons* gives a near-diagonal matrix. The sequences it confused were all "Walking" sequences, ("Walking", "Walking slowly", and "Walking on uneven terrain").

We also examine the effect of dimensionality reduction by PCA. We did PCA on the **61-LOG-cons** feature-set, and we observed that the classification accuracy was preserved, as long as we retain the first 18 components (or more). This

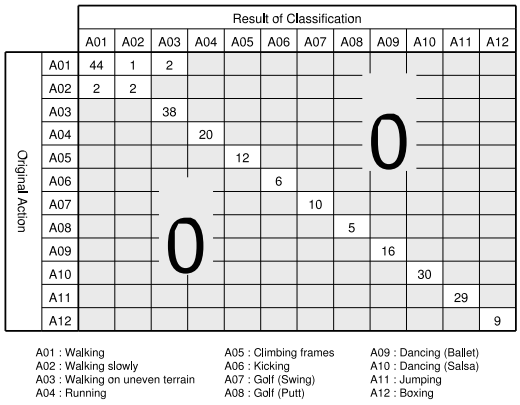


Figure 3: *Confusion Matrix of condition 61-LOG-cons in Table 1. Notice that (a) it is near-diagonal and (b) the confused motions are very similar (A*01*-A*03*), all being variations of walking motions.*

is a pleasant surprise - we were expecting a small loss of accuracy, after such a drastic dimensionality reduction (61 to 18).

4.2. Visualization

Here we show that, even with 3-d feature-sets, we can still have a useful visualization. We use the following features for experiments: the three features that we manually derived, as shown in Equation (5) (**3-MAN**), and the first three principal components of PCA from the *61-LOG-cons* feature-set (**3-PCA**).

3-MAN led to 11.50% classification error, outperforming *3-PCA* (with 15.49%). This result was another pleasant surprise: the human intuition behind *3-MAN* won over PCA, which is mathematically optimal under the L2 norm.

Thus, we use the *3-MAN* feature set for visualization. Figure 1 shows the scatter-plot of motion capture sequences in this 3-d space. The scatter-plot leads to observations that agree with our intuition, underlining the effectiveness of our chosen feature-sets: (a) The trunk energy during *walking on uneven terrain* is higher than during a normal walk. (b) *Frame-climbing* requires roughly the same energy of all body parts. (c) Trunk energy during *walking on eneven terrain* is higher than during a normal walk. (d)*Running* and *walking* have similar proportions of energy (upper body vs. lower body and limbs vs. trunk).

Moreover, the scatter-plot can help us spot outlier motions. For example, the points inside the red circle of Figure 1, correspond to actions with high energy; closer inspection shows that they are noisier, and the first derivatives skyrocket. This is clear in their time-plots, which are omitted for space (see [OFH07]). The offenders correspond to *ballet dancing* motions, labeled as #5-6 and #5-8 in `http://mocap.cs.cmu.edu/`.

5. Conclusions

The goal of this paper is to find an effective and fast-to-compute distance function between two unequal-length motion capture sequences. Our main contribution is that we proposed a low-dimensionality feature-set for each motion capture sequence. After extensive experiments on 112 possible variations, on a large real motion capture dataset, we propose two methods, 61-LOG-cons (most accurate) and 3-MAN (best for visualization). In all variations, the idea is to consider the total approximate kinetic energy expended, by each of the approximately 70 angles in the data. The resulting feature sets achieve the original design goals:

- *Speed:* Our proposed distance function is *fast* to compute, independent of the duration of the motion capture sequences.
- *Effectiveness* 61-LOG-cons gives excellent classification accuracy, and provides an excellent starting point for dimensionality reduction (with PCA, or 3-MAN), for visualization, clustering, and outlier detection (see Figure 1).

A promising direction for future work is to extend this approach to subsequence search. Another direction is to also consider the potential energy, in addition to the kinetic one.

References

[HJB*05] HAMID R., JOHNSON A., BATTA S., BOBICK A., ISBELL C., COLEMAN G.: Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. In *Proc. of CVPR* (2005), vol. 1, pp. 1031–1038.

[Jol86] JOLLIFFE I. T.: *Principal Component Analysis*. Springer, 1986.

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Trans. Graph. 21*, 3 (2002), 473–482.

[KPZ*04] KEOGH E., PALPANAS T., ZORDAN V. B., GUNOPULOS D., CARDLE M.: Indexing large human-motion databases. In *Proc. of VLDB* (2004), pp. 780–791.

[OFH07] ONUMA K., FALOUTSOS C., HODGINS J. K.: FMDistance: a fast and effective distance function for motion capture data. *CMU SCS Technical Report* (2007), CMU–CS–07–164.

[RPE*05] REN L., PATRICK A., EFROS A. A., HODGINS J. K., REHG J. M.: A data-driven approach to quantifying natural human motion. *ACM Trans. Graph. 24*, 3 (Aug. 2005), 1090–1097.

[Tro02] TROJE N. F.: Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision 2*, 5 (2002), 371–387.