Distance matrices as invariant features for classifying MoCap data

Antonio W. Vieira, William Schwartz, Mario Campos and Thomas Lewiner

MAT. 07/12
Distance matrices as invariant features for classifying MoCap data

Antonio W. Vieira, William Schwartz, Mario Campos and Thomas Lewiner


April 7th, 2012


Matemática Pura
Matemática Aplicada

Nacional
Internacional

Modelagem Geométrica e Computação Gráfica

Aproximação invariante de curvaturas Euclideana, afim e projetiva
Distance matrices as invariant features for classifying MoCap data

ANTONIO W. VIEIRA\textsuperscript{1}, WILLIAM SCHWARTZ\textsuperscript{1}, MARIO CAMPOS\textsuperscript{1} AND THOMAS LEWINER\textsuperscript{2}

\textsuperscript{1} Department of Computer Science — UFMG — Belo Horizonte — Brazil
\textsuperscript{2} Department of Mathematics — Pontifícia Universidade Católica — Rio de Janeiro — Brazil

Abstract. This work introduces a new representation for Motion Capture data (MoCap) that is invariant under rigid transformation and robust for classification and annotation of MoCap data. This representation relies on distance matrices that fully characterize the class of identical postures up to the body position or orientation. This high dimensional feature descriptor is tailored using PCA and incorporated into an action graph based classification scheme. Classification experiments on publicly available data show the accuracy and robustness of the proposed MoCap representation.

Keywords: Distance Matrix. Invariant descriptor. MoCap data. Action Graph. 3d video. Motion classification.

![Figure 1: Examples of skeleton postures and their respective distance matrices.](image)

1 Introduction

Human motion has been an active research area due to its many applications in computer vision, animation, biometrics and sports. The standard technique for generating natural-looking motion sequences is human 3D motion capture (MoCap), and a large amount of such data is now available. However, the acquisition and processing of such data is still costly, which emphasizes the growing need for re-using previously recorded data. This motivates the development of automatic methods for comparison, classification and retrieval of Mocap.

Although comparing two motion sequences is an easy task for a person, automatic comparison is hard due to enormous numerical differences between two similar motion sequences. Spatial variations are mostly due to almost rigid transformations among similar postures. Temporal variations are due to non-linear differences in the dynamic of an action when performed by different subjects or even by two different performances of the same subject. Thus, methods for motion classification need a suitable spatial representations for comparing postures and a scheme for temporal alignment of sequences with the spatially identified posture.

Increasing the robustness of the spatial representation can improve on both spatial and temporal comparisons, and this is the main purpose of the present work.

Preprint MAT. 07/12, communicated on April 7\textsuperscript{th}, 2012 to the Department of Mathematics, Pontifícia Universidade Católica — Rio de Janeiro, Brazil.
Related Work  Most of early literature on the recognition of human motion focuses on 2D video sequences, while the amount of work on 3D data is comparatively limited, mainly due to the difficulty of 3D data acquisition. We refer to the excellent survey of Weinland et al. [10] on action representation, segmentation and recognition. They classify spatial data into local, global and parametric representations. This work focuses on Motion Capture (MoCap) data, which is a parametric representation widely used and with several publicly available datasets [1][11].

For application purposes, MoCap data is generally described in a view-invariant manner. For content-based human motion retrieval applications, Chiu et al. [2] proposed a posture descriptor where each skeletal segment is represented by the local spherical coordinate relative to the root orientation. Kovar and Gleicher [4] proposed a technique to overcome the lack of robustness in spatial comparisons, in particular joint orientations and angular velocities. For pose-to-pose distance calculations, Forbes and Fiume [3] proposed a weighted PCA-based pose representation, reducing from the list of quaternions corresponding to each joint’s angular position. Geometric relations between body key points of a pose was presented by Müller et al. [7] and for annotation [8]. In a MT, a motion sequence defines a matrix where each column have boolean features to represent a posture and Dynamic Time Warping is used for time alignment among different sequences. A class of motion is then represented as a weighted sum of pre-aligned individual sequences. Raptis et al. [9] developed a real-time system for classifying dance gestures using angular representation of the skeleton in a reference frame. The classifier relies on the strong assumption that the input motion adheres to a known musical beat to assert canonical time-base alignment.

Contributions  While most methods usually consider some reference frame and compute features using different measures, generally angles, we prove that distances among joint positions, concatenated into our distance matrices, are enough to completely represent a posture up to its global position and orientation. Equivalent postures are mapped to a single point in distance matrices-space, and robustness is ensured since close-by points in the feature space correspond to close-by poses. Therefore, spatial variations among similar postures can be efficiently addressed in a clustering strategy in the low dimensional feature space and temporal variations are addressed using an action graph strategy [5].

2 Methodology

This section details the construction of the distance matrix as invariant features, dimension reduction and the action graph based classification method.

(a) Distance Matrix as Invariant Features

A MoCap skeleton posture \( S \) is an \( n \)-tuple of 3D points \( S = \{p_1, p_2, \ldots, p_n\} \) describing the body joints of a posture. A motion \( M \) is a sequence of \( m \) skeletons ordered in time: \( M = \{S_1, S_2, \ldots, S_m\} \) (see Figure 3). Two given postures with the same semantic may have very different sequence of coordinates depending on the body position, orientation and point of view. This turns the comparison of two given postures a very difficult task. In order to address this issue, we show that using joints distance matrix leads to an unambiguous representation for postures that is invariant to rigid transformations.

Let \( H \) denote the set of all possible postures. We define an equivalence relation \( \mathcal{R} \) over \( H \), saying that two postures \( (S, S') \in H^2 \) are equivalent if there exists a rigid transformation \( T \in \mathbb{R}^3 \) such that \( T(S_i) = S_j \). The quotient \( H/\mathcal{R} \) is a set of posture classes, i.e., each \( \overline{S} \in H/\mathcal{R} \) contains postures equivalent by rigid transformations. Those classes refer to a posture independently of the global position and orientation. Hence, once we obtain a single descriptor for all the elements of a class \( \overline{S} \in H/\mathcal{R} \), we have a posture descriptor that is invariant to rigid transformations.

For a posture \( S \), we define its \( n \times n \) matrix \( d(S) = [||p_j - p_i||]_{i,j} \) of distances among all joints. Figure 1 shows an example of three skeleton postures and corresponding distance matrices. Since rigid transformations preserve distances, \( d \) is a posture descriptor that is invariant to rigid transformation: if two postures \( S, S' \) are equivalent, they have the same matrix of distances: \( d(S) = d(S') \). The main question is the converse, i.e. whether those distance matrices characterize postures.

Indeed, a useful property of distance matrices is that, in the same way triangles with equal edge lengths are congruent, postures with equal distance matrices are equivalent, that is, a class of equivalent postures is completely described by its distance matrix.

Theorem 1. Distance matrices fully characterize a class of equivalent postures, that is, \( \forall S, S' \in H, d(S) = d(S') \Rightarrow \exists T \text{ rigid }, T(S) = S' \).

This theorem can be geometrically understood with the following two steps construction. First, \( T \) is defined as the rigid transformation mapping \( \{p_1, p_2, p_3, p_4\} \) to \( \{p'_1, p'_2, p'_3, p'_4\} \). Then, for each \( i > 4 \), point \( T(p_i) \) is the

Example of a sequence of skeletons from the motion “JumpingJack”.

Figure 2: Example of a sequence of skeletons from the motion “JumpingJack”.
(non-empty) intersection of the 4 spheres of centers \( p_1, p_2, p_3, \) and \( p_4 \), and radii \( \| p_1 p_i \|, \| p_2 p_i \|, \| p_3 p_i \|, \) and \( \| p_4 p_i \| \). However, to cope with the degenerate case, we provide a more straightforward demonstration.

**Proof.** Given two postures \( S = \{ p_1, p_2, \ldots, p_n \} \) and \( S = \{ p_1', p_2', \ldots, p_n' \} \) with the same distance matrix, define two sequences of \((n - 1)\) vectors: \( v_i = p_i - p_0 \) and \( v'_i = p_i' - p_0 \). Since triangle \( p_0 p_i p_1 \) is congruent to triangle \( p_0' p_i' p_1' \), their internal angles are pairwise equals. In particular: \( \forall i, j, v_i \cdot v_j = v_i' \cdot v_j' \) (even if \( i = j \)), i.e., the Gram matrix of \( \{ v_i \} \) equals the Gram matrix of \( \{ v_i' \} \). Therefore, there exist (eventually more than one) linear rigid motion \( R \) that maps the sequence of vectors: \( \forall i, R(v_i) = R(p_i - p_0) = v'_i = v_i' - p_0 \). Defining the rigid transformation \( T \) as the composition of \( R \) with the translation of vector \( p_0' - R(p_0) \), we get \( T(p_i) = R(p_i) + (p_0' - R(p_0)) = R(p_i - p_0) + p_0' = (p_i' - p_0') + p_0 = p_i' \).

Observe that two close-by postures are associated to close-by distance matrices, which is not the case for angle-based descriptors (due to the discontinuity around \( 2\pi \)). This turns the distance matrix descriptor more suitable for dimension reduction techniques. Moreover, a random symmetric, positive and diagonal-free matrix, is in general not a distance matrix, adding robustness to numerical error and thresholds to our descriptor.

The results above allow us, without loss of generality, to refer to a motion as a sequence of distance matrices: From now on, each posture will be referred to as a point \( d \in \mathbb{R}^{n \times n} \), discarding its global position or orientation. A role motion then describes a continuous curve in \( \mathbb{R}^{n \times n} \).

(b) Dimension Reduction

Distance matrices are symmetric with null diagonal elements. As stated in the proof of Theorem 1, distance matrices are actually characterized with only 4 lines, showing a strong correlation between its elements. This suggests that dimension reduction will be effective for devising low dimensional features descriptors for posture class.

We refer to the \( n \times n \) elements of a distance matrix \( d \) as a feature vector \( f \in \mathbb{R}^{n^2} \). Considering a training set of motion sequences \( \{ M_1, M_2, \ldots, M_l \} \) to be used in the classification process, we concatenate all the feature vectors \( f_i \) from all postures in all motion sequences into matrix \( X = (f_1, f_2, \ldots, f_N) \). We then use Principal Component Analysis (PCA) on \( X \) to obtain a lower dimensional feature vector for each posture. For each feature vector \( f_i \), we project it into the subspace spanned by the first \( k \) principal components. The reduced invariant descriptor is the resulting vector \( e_i \in \mathbb{R}^k \). Applying PCA allows for both dimension reduction and noise suppression, since small variations are discarded with the components greater than \( k \).

In practice, only few principal components are enough to represent postures in a discriminative way. Figure 3 shows an example of low dimensional features from three different motions, from the HDM dataset, projected onto \( k = 3 \) principal components. Points in red are from 13 performances of JumpingJack, points in green are from 13 performances of ElbowToKnee and, points in blue are from 15 performances of SitDownKneelTieShoes. Notice that there is a common start posture shared by the three different motion classes.

All motions in Figure 4 were performed by five different actors. In order to illustrate the similarity of curves of a motion when performed by different actors, Figure 5 shows points from the motion JumpingJack with different colors for each actor.

(c) Action Graph

To build our classification scheme with time alignment, we use a grammar based approach, where an observation and a transition model are learned to describe the dynamics of motions in an Action Graph 5.

An Action Graph uses a set of salient postures shared among all motion classes to explicitly model the dynamics of human motion. To construct the action graph, we use an unsupervised phase to cluster feature vectors in a set of salient postures. The salient postures will be used as nodes for the graph and each action modeled as a path in the graph.

Our motion recognition system is composed of a set \( A \) of \( h \) trained motion classes, a set with \( t \) salient postures \( V = \{ v_1, v_2, \ldots, v_t \} \), a set \( \Sigma(e) = \{ p(e|v_1), p(e|v_2), \ldots, p(e|v_t) \} \) with observation model of a feature vector \( e \) with respect to salient postures \( v_i, i = 1, \ldots, t \) and a set of \( h \) graphs representing the transition probability between salient postures.
Given a test motion sequence \( M \), we obtain a sequence of feature vectors \( e(M) = \{ e_1, e_2, \ldots, e_r \} \), and compute the probability \( p(e(M)|a) \) of occurrence of \( M \) with respect to each trained motion class \( a \in A \). The resulting class \( \bar{a} \) is the one that maximizes this probability. The decoding process to compute \( \bar{a} \) uses a dynamic programing scheme to save computational time. More details can be found in Li et al.’s work [5].

3 Experiments

In this section we provide experiments with public data to validate our features for motion classification. Our experiments were performed using a Core i5 CPU with 4 GB RAM. The action graph were implemented in C/C++ and compiled with gcc for Ubuntu Linux. Publicly available MoCap data from HDM05 database [11] were used in our experiments.

We consider 10 different motion classes, performed by five different subjects. A total of 156 motion sequences were used in a 10-fold cross-validation. Table 1 reports on the 10 motion classes used in this experiment, and the confusion matrix is shown in Figure 5.

<table>
<thead>
<tr>
<th>Action Class</th>
<th>(ns)</th>
<th>(nf)</th>
<th>(%)</th>
<th>(std)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClapAboveHead</td>
<td>14</td>
<td>6102</td>
<td>85.71</td>
<td>4.33</td>
</tr>
<tr>
<td>SitDownChair</td>
<td>14</td>
<td>4502</td>
<td>87.89</td>
<td>3.10</td>
</tr>
<tr>
<td>SitDownFloor</td>
<td>15</td>
<td>5679</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>PunchRFront</td>
<td>14</td>
<td>6450</td>
<td>90.01</td>
<td>2.21</td>
</tr>
<tr>
<td>KickLFront</td>
<td>11</td>
<td>5102</td>
<td>88.11</td>
<td>2.71</td>
</tr>
<tr>
<td>JumpingJack</td>
<td>13</td>
<td>5589</td>
<td>88.89</td>
<td>1.78</td>
</tr>
<tr>
<td>ElbowToKnee</td>
<td>13</td>
<td>5711</td>
<td>94.13</td>
<td>1.52</td>
</tr>
<tr>
<td>Squat</td>
<td>12</td>
<td>7619</td>
<td>89.09</td>
<td>2.33</td>
</tr>
<tr>
<td>StandUpKnee</td>
<td>13</td>
<td>2371</td>
<td>93.33</td>
<td>1.10</td>
</tr>
<tr>
<td>SitDownKnee</td>
<td>13</td>
<td>9010</td>
<td>91.44</td>
<td>2.55</td>
</tr>
</tbody>
</table>

Table 1: Motion classes used in our experiments: (ns) number of sequences, (nf) number of frames, (%) mean accuracy, (std) standard deviation.

4 Conclusions

We presented a distance matrix based feature for representing MoCap data and demonstrated that such representation leads to unambiguous view invariant descriptor for postures. By using a dimension reduction scheme and an action graph based classifier, we showed that such representation is suitable for classifying motion sequences. Future works will focus on identifying sub-matrices associated with specific motion classes in order to allow for classification with occlusion of joints not related to an action class.

Acknowledgements

This work is partially financed by CNPq, FAPEMIG, FINEP, CEMIG.

References