Humanized Robot Dancing: Humanoid Motion Retargeting Based in a Metrical Representation of Human Dance Styles

Paulo Sousa[1,3], João L. Oliveira[1,2,3], Luis Paulo Reis[1,3], and Fabien Gouyon[2]

[1] DEI/FEUP - Informatics Engineering Dep.,
Faculty of Engineering of the University of Porto, Portugal
[2] INESC Porto - Systems and Computers Engineering National Inst., Porto, Portugal
[3] LIACC - Artificial Intelligence and Computer Science Lab.,
Univ. of Porto, Portugal
{ei06047,joao.lobato.oliveira,lpreis}@fe.up.pt, fgouyon@ienscporto.pt

Abstract. Expressiveness and naturalness in robotic motions and behaviors can be replicated with the usage of captured human movements. Considering dance as a complex and expressive type of motion, in this paper we propose a method for generating humanoid dance motions transferred from human motion capture (MoCap) data. Motion data of samba dance was synchronized to samba music, manually annotated by experts, in order to build a spatiotemporal representation of the dance movement with variability, in relation to the respective musical temporal structure (musical meter). This enabled the determination and generation of variable dance key-poses according to the captured human body model. In order to retarget these key-poses from the original human model into the considered humanoid morphology, we propose methods for resizing and adapting the original trajectories to the robot joints, overcoming its varied kinematic constraints. Finally, a method for generating the angles for each robot joint is presented, enabling the reproduction of the desired poses in a simulated humanoid robot NAO. The achieved results validated our approach, suggesting that our method can generate poses from motion capture and reproduce them on a humanoid robot with a good degree of similarity.

Keywords: Humanoid Robot Motion Generation, Motion Retargeting, Robot Dance.

1 Introduction

Robotics applications grow daily, and the creation of realistic motion for humanoid robots increasingly plays a key role. Since motion can be regarded as a form of interaction and expression, that allows to enrich communication and interaction, improving humanoid robot motion expressiveness and realism, as a form to accomplish better and richer human-robot interaction. A form of achieving more humanized robotic motion is to feasibly reproduce, and imitate, the motions performed by humans. This would allow not only more expressiveness,

L. Antunes and H.S. Pinto (Eds.): EPIA 2011, LNAI 7026, pp. 392–406, 2011.

diversity and realism in the humanoid robot motion, but also a simple, less time-consuming and automatic form of creating and converting diverse human motion to robotics. Considering dance as a rich and expressive type of motion, constituting a form of non-verbal communication in social interactions, also transmitting emotion, it imposes a good case study of clear humanized motion.

This paper presents methods for generating humanoid robot dancing movement from human motion capture data. The presented methods aim to generate robot dance motion with a good degree of similarity and musical expressiveness according to the original dance. All the described processes are built upon [11]'s method of dance motion analysis. This method was applied to motion data of samba dance style synchronized to samba music (manually annotated by experts), for building a spatiotemporal representation of the original dance movement, with variability, in relation to the respective music temporal structure (musical meter), allowing the determination of the fundamental key-poses of the dance style. Using this representation as starting point, we present methods for resizing the body segments and retargeting the joint trajectories towards different humanoid body morphologies. As such, we firstly synthesize stochastic variations of the determined key-poses, then resize these according to the targeted segment lengths and finally process the necessary adjustments to retarget the generated joint trajectories onto the considered humanoid morphology. The method was tested on a humanoid robot NAO [2], simulated on the SimSpark simulation environment [1] and using the FC Portugal agent [13] [5], in order to generate and reproduce the original samba dance.

The remainder of this paper is structured as follows: Section 2 presents a review of related work on human dance motion analysis and humanoid dance motion generation. Section 3 presents the proposed methods and the developed work. In Section 4, the main results are presented and discussed and a evaluation of the similarity between the original human motion and the generated humanoid motion is performed. Finally, in Section 5 the conclusion and future work are presented.

2 Related Work

Nowadays many attempts have been made to achieve realistic humanoid motion based on human motion. The techniques referred below use dance motion capture data and synthesize new motion from it. The first step in this process typically consists on determining the most important key-poses from the motion capture data. This choice impacts the overall aspect of the final motion, and must be accurate at determining the key-poses that best represent the original dance style. From these key-poses, the motion is then transfered to the robot, by trying to achieve the greater similarity possible with the original motion capture data.

2.1 Motion Analysis

One of the traditional methods to generate dance motion in computer animation and in robotics, is to interpolate transition motion between defined key-poses.

As such, key-poses most show representative instances of the motion. The techniques to analyze and determine the appropriate key-poses, in dance motion, are based on a spatial analysis of the performed body motion [6] [4] or by analyzing the dance music [11], and, in some cases, a simultaneous analysis of both aspects [18] [19] [17] [10]. Working on Latin dances [6], uses information about the main characteristics of Merengue dance style, focusing only on components of the rotation of shoulders and hips. By analyzing Japanese folk dance, [8] and [7] segmented the dance MoCap data in key-poses, in terms of minimum velocity of the end-effectors' (hands and feet). Then these key-poses were clustered and interpolated to generate the original dance. [4] extracts rapid directional changes in motion.

On the other hand, a combination of music and motion analysis is applied in [18] [19] [17] [10]. [18]. These authors identify key-points of the motion rhythm, as local minimums of the 'weight effort' (linear sum of rotation of each body joint), which indicate stop-motions and are recognized as key-poses; and motion intensity points as the average of instant motion from the previous key-pose. Their music analysis focus on the music intensity (sound chunk whose spectral power is strongest between the neighboring frequency sounds) and music rhythm, that is found by analyzing the repetition of several phrases and patterns presented in the music structure. In [19], stop frames of the hand motion are considered as key-frames, and motion intensity is determined as the difference in the velocity of the hands between frames. The performed musical analysis is similar to [18], extracting the music beat and degree of chord change for the beat structure analysis, and music intensity for mood analysis. There are also works based on temporal scaling techniques, for upper-body [17] and leg [10] motion. In [17] the dance motion is captured at different speeds and by comparison of the variance in the motion, the authors observed that some poses stay preserved. These poses are considered key-poses since they tend to represent important moments to the music. The analysis in [10] is similar to [17], but focusing on the analysis of the step motion. The determination of the key-poses is made by using the indication that the original timings for tasks around key-poses are maintained and that stride length is also close to the original, even at different speeds. In [11] a spatiotemporal dance analysis model is presented, based on the Topological Gesture Analysis (TGA) [9], that conveys a discrete point-cloud representation of the dance. This method describes the space that the dancer occupies at each musical class (1 beat, half-beat, 2 beats, etc) in terms of point-clouds, and generates a spatiotemporal representation of the occupied positions, by projecting musical cues onto spatial trajectories. We followed [11]'s method of dance motion analysis since it conveys a parameterizable representation of the original dance, incorporating its intrinsic variability and musical rhythmic qualities.

2.2 Dance Motion Generation

Dance motion generation techniques typically aim at generating motion from the key-poses extracted in a prior motion analysis phase. [14], [15], [3], [10]

and [8] apply Inverse Kinematics (IK) to transform the marker positions from the motion capture data into robot joint angles. In [3], the Inverse Kinematics is only applied to the upper-body, while the pelvis, leg and feet motion is generated by optimization based on the Zero Motion Point (ZMP) trajectory and dynamic mapping. [10] checks the intervals between steps to keep a stable ZMP and then also applies Inverse Kinematics to map the leg joint positions to robot joint angles. [14] applies optimization to ensure the physical restrictions of the robot, ensuring that the angle, velocity and acceleration limits are met. [15] applies sequential motion restrictions by optimization by firstly limiting the joint angles, then solving self-collision avoidance and, finally, overcoming velocity and dynamic force constraints. For assuring self-collision avoidance the authors increased the critical distances for the periods that present collisions [20], by applying kinematics mapping to translate the motion capture data to the humanoid, using similarity functions based on the value of the angles of the original data. For improving the balance, an algorithm based on the number of feet in contact with the ground is used, modifying the hip trajectory to satisfy a constraint based on a ZMP criterion. [8] also limits angles and the angular velocity and finally modifies the ZMP trajectory in order to keep balance. [4] and [18] use motion graphs to represent the motion. [4] aimed to generate motion on-the-fly, using a library of motion graphs, matching then the motion represented by the graph with the input sound signal. On its hand, [18] traces the motion graph based on the correlation between the music and motion features. Finally, [18] determines the better graph path by choosing the highest value from the correlation between the music and motion intensity and a correlation between music beats and motion key-frames. In terms of motion retargeting, [4] uses a real-time algorithm to adapt the motion to the target character. [17] applies optimization to overcome the joint angles limitations of the target character. [16] presents a way to extract the joint angles from the three-dimensional point representation of a pose. This method can be applied not only for computer animation but also for robotics. After generating the motion, [14] applies a phase of motion refinement to detect trajectory errors and correct them. On its hand, [10] makes a final refinement of the generated movement in order to keep the robot's balance and avoid self-collisions. To generate the robot joint angles from the pose point representation obtained from the previous dance movement representation [11] we based our approach in [16], for extracting Euler angles in the 3 dimensions based on a body-centered axis system.

3 Methodology

3.1 Dance Movement Analysis

Our motion analysis stage is based on the approach presented in [11]. As such, we recurred to the same dance sequences of Afro-Brazilian samba, which were captured with a MoCap system, and synchronized to the same genre of samba music (manually annotated by experts). Upon these, we also applied the TGA

(Topological Gesture Analysis) method [9] for building a spatiotemporal representation of the original dance movement in relation to the respective music temporal structure (musical meter). This method relies in the projection of musical metric classes onto the motion joint trajectories, and generates a spherical distribution of the three-dimensional space occupied by each body joint according to every represented metric class. In such way, this representation model offers a parameterizable spatiotemporal description of the original dance, which translates both musical qualities and variability of the considered movement.

3.2 Dance Movement Generation

The actual dance movement generation is based on three steps: (a) key-poses synthesis from the given representation model, (b) morphological adaption of the key-poses to the used body model, in terms of segments length and number of joints, and (c) the actual key-poses' retargeting from the used character to the simulated robot NAO.

Key-Poses Synthesis. By following [12], the synthesis of key-poses consisted on calculating a set of full-body joint positions (one for each considered metric class). In order to translate the variability imposed in the original dance, for every key-pose the joint positions were calculated by randomly choosing rotations circumscribed by every joints' TGA distributions without violating the fixed geometry of the human body.

As described in [12], each key-pose is split into 5 kinematic chains. From the anchor to the extremity of each kinematic chain, each joint position p_j^m is calculated based on a random rotation circumscribed by the possible variations of its rotation quaternion $qv_{s_j^m}$ (*i.e.* the 3d rotation of a target unity vector $\overrightarrow{v}'_{s_j^m}$ around its base unity vector $\overrightarrow{v}_{s_j^m}$) between every two body segments:

$$p_j^m = p_{j-1}^m + l_{j-1,j} * \overrightarrow{v}'_{s_j^m} : p_j^m \in T_j^m, \tag{1}$$

where m is the considered musical metrical class, p_j^m is the current joint position under calculation, p_{j-1}^m is the former calculated joint position, $l_{j-1,j}$ is the current segment length (linking p_{j-1}^m to p_j^m), and T_j^m is the current TGA spherical distribution.

Morphological Adaption. In order to get a representation of the key-poses in the target humanoid morphology the prior spatiotemporal representation of the dance must be adapted, maintaining the spatial relationship and expression of the poses across all the represented metrical classes. To achieve this, we must look at the target morphology in terms of segment lengths, number of joints, joints' degree of freedom, and other target physical kinematic constraints.

(a) Different Segment Lengths: Prior to the actual humanoid key-poses' generation, the segment lengths of each body part length must be changed to those of the target body model (figure 1). For each joint j, $l_{j-1,j}$ is the length of the

segment that connects $j-1$ to j, and D_j^m is the spherical distribution, with radius r_j^m and center o_j^m. The distance from o_{j-1}^m to o_j^m is considered as $d_{j-1,j}^m$ and direction vector from o_{j-1}^m to o_j^m is $\overrightarrow{vo}_{j-1,j}^m$. In order to change the segment length from $l_{j-1,j}$ to $l'_{j-1,j}$ we proceeded as follows:

$$\begin{cases} redim = l'_{j-1,j}/l_{j-1,j} \\ d'^m_{j-1,j} = d^m_{j-1,j} * redim \\ r'^m_j = r'^m_j * redim \\ o'^m_j = o^m_{j-1} + d'^m_{j-1,j} * \overrightarrow{vo}^m_{j-1,j} \end{cases}, \tag{2}$$

where $bd'^m_{j-1,j}$ is the new distance from o_{j-1}^m to o_j^m, and r'^m_j the adapted radius of the spherical distribution D_j^m, with o'^m_j as its new center point. The translation that was applied from o_j^m to o'^m_j is then applied to all the following joint centers in the considered kinematic chain (noted by *trans* in figure 1). This method allows to resize any body segment by manipulating the spherical distributions of the movement representation according to the target segment lengths. Only the anchor sphere of the body model isn't resized or moved. The relation between the segment length and the radius was considered linear, as pointed by *redim* in eq. (2). The method only performs a translation of the spherical distributions centers maintaining the relation between the spherical distributions centers. The change in the spherical distribution radius, is regarded a linear an adaption of the segment reach.

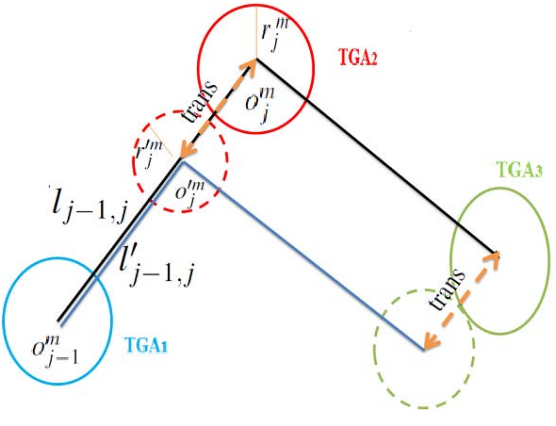

Fig. 1. Resize of a segment (from TGA1 to TGA2) and application of the translation over the rest of the kinematic chain (TAG3)

(b) Different Number of Joints:

In order to have the correct body morphology, it may be necessary to remove joints from the MoCap body model that the target humanoid body doesn't have. Considering three adjacent joints j, $j+1$ and $j+2$, the goal is to determine a segment from j to $j+2$ that is closest to the spherical distribution of $j+1$, and discard $j+1$ as required (figure 2a)). So, in order to choose points inside D_j^m

and D^m_{j+2}, we calculate the interception, in the form of the spherical cap C^m_j, of the spherical distribution D^m_{j+2} for the target joint, with a sphere S^m_j centered in the position of the first joint, p^m_j, and with radius equal to $l_{j,j+1} + l_{j+1,j+2}$. Then, if C^m_j is an empty set, the center of D^m_{j+2} is translated in the direction of the vector from p^m_j to o^m_{j+2} (figure 2b)), increasing or decreasing the distance, from p^m_j to o^m_{j+2}, this way assuring to obtain interception between D^m_{j+2} and S^m_j. Finally, a search for a point in C^m_j is employed. The segment that connects this calculated point to p^m_j must be closest to the sphere center of the eliminated joint, o^m_{j+1}.

In the special case where p^m_j is the anchor point of the model (and first point to be determined in the model), we can also move p^m_j, keeping it inside D^m_j, in the direction of the vector from p^m_j to o^m_{j+2} to enable the interception, and then move p^m_j, in order to approach it from p^m_{j+1}. This enables a better fit of the segment that will be traced. This method is applied to erase the middle joint of the spine (joint MSP in figure 4).

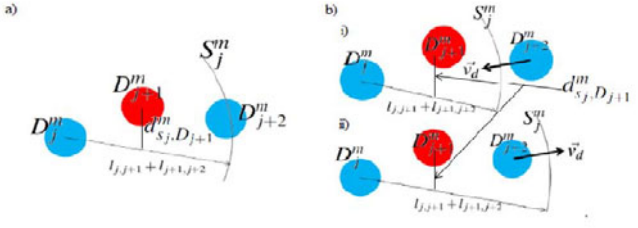

Fig. 2. Morphological adaption of a kinematic chain with three joints (j, $j+1$, $j+2$) to one with two joints (j, $j+2$), by "erasing" the middle joint

(c) Additional Physical Restrictions: Another problem that we faced in the motion retargeting was related to the necessity to ensure that certain body segments were collinear. In order to ensure that the segment from j to j' is collinear with the segment from $j2$ to $j2'$, firstly we generate, to the segment from j to j', the random quaternion and its corresponding direction vector $\overrightarrow{v}^m_{j,j'}$ and try to apply it to the other segment involved (situation exemplified in figure 3).

$$(p^m_j + \overrightarrow{v}^m_{j,j'} * l_{j,j'}) \in D^m_{j'} \tag{3}$$

$$(p^m_j + \overrightarrow{v}^m_{j,j'} * l_{j,j'}) \in D^m_{j'} \wedge (p^m_{j2} + \overrightarrow{v}^m_{j,j'} * l_{j2,j2'}) \in D^m_{j2'}. \tag{4}$$

Condition (3) is the constraint applied to accept a generated point in [12], which the algorithm will try to ensure while not reaching the maximum number of attempts. When this maximum is reached, condition (4) must be ensured instead, by using symmetric vectors in both segments. If the method can't generate a vector that satisfies condition (4), then it will generate a vector that can satisfy (3), and calculate a new point $pf^m_{j2'}$ from it:

$$pf^m_{j2'} = p^m_{j2} + \overrightarrow{v}^m_{j,j'} * l_{j2,j2'}. \quad (5)$$

This new point will be outside the spherical distribution $D^m_{j2'}$ of $j2'$, so the distance from $pf^m_{j2'}$ to the center of $D^m_{j2'}$ is assumed as the translation that must be applied to all the following spherical distribution center points in the same kinematic chain. This translation ensures that the centers of the remaining spherical distributions still maintain their spatial relation until the extremity joint of the chain, this way keeping the original key-pose posture without compromising the fixed geometry of the humanoid body model. This method was applied over the hip segments of the model.

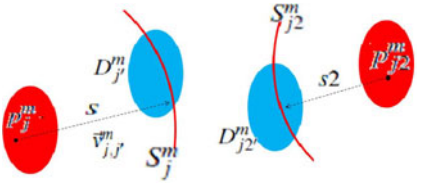

Fig. 3. Physical Restriction example. Segment s and $s2$ must be parallel.

Key-Poses Retargeting. To generate the actual robot joint angles from the representation obtained from the application of the previous adaption methods, we applied a motion retargeting technique based on [16] for extracting the Euler angles of each joint, in the 3 dimensions, based on a body-centered axis system. This method considered the robot joint's degrees-of-freedom (DoFs) and their singularities of different natures. The following procedure was applied to all synthesized key-poses adjusted to the target humanoid morphology.

At first, the local coordinate system for the robot upper-body R_{ub} is defined in the chest of the previously resized and adapted body model, as follows:

$$\begin{cases} R^y_{ub} = \boldsymbol{v}_{SH} = p_{LSH} - p_{RSH} \\ R^{z'}_{ub} = \boldsymbol{v}_{SP} = p_{CSH} - p_{CHIP} \\ R^x_{ub} = R^y_{ub} \times R^{z'}_{ub} \\ R^z_{ub} = R^x_{ub} \times R^y_{ub} \\ R_{ub} = [norm(R^x_{ub}), norm(R^y_{ub}), norm(R^z_{ub})] \end{cases}, \quad (6)$$

where $norm(X) = \frac{X}{|X|}$ and $\times$ is the cross product between two dimensions. And p_{Name} represents the position of a body joint illustrated in 4a).

Using the vectors from the global coordinate system, the correspondent vectors in the local coordinate system are calculated and the angles in all the axis determined. As R_{ub} represents a rotation matrix, the product of that rotation matrix and a global unit vector will result in the corresponding local unit vector.

The retargeting started at the left shoulder, considering a vector $\boldsymbol{v}_{LSE}$ connecting the shoulder's joint global position p_{LSH} to the elbow's p_{LELB}.

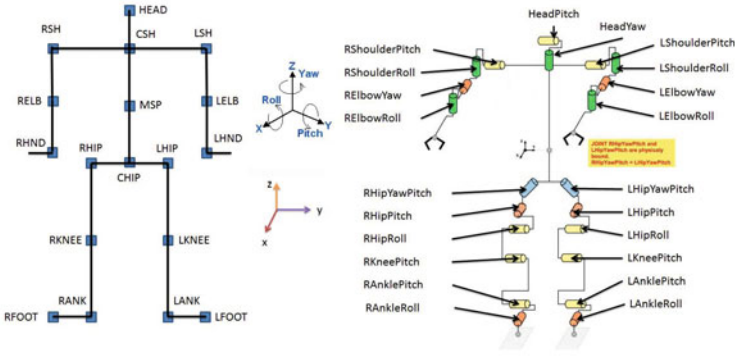

Fig. 4. a) Considered labels for the joints of the original MoCap body model (left); b) Joint rotation and Cartesian axes. c) Humanoid robot NAO body model description.

According to the specified joint's characteristics, firstly, the rotation angle in the y axis is extracted, which corresponds to the pitch rotation of the robot's left shoulder $LShoulderPitch$:

$$\begin{cases} \boldsymbol{v}_{LSE} = p_{LELB} - p_{LSH} \\ \boldsymbol{v'}_{LSE} = R_{ub}^{T} \times \boldsymbol{v}_{LSE} \\ LShoulderPitch = atan2(\boldsymbol{v'}_{z_{LSE}}, \boldsymbol{v'}_{x_{LSE}}) \end{cases} . \tag{7}$$

Then, the rotation of the left shoulder in the z axis (roll rotation $LShoulderRoll$), is calculated:

$$\begin{cases} \boldsymbol{v''}_{LSE} = R_y(LShoulderPitch) \times \boldsymbol{v'}_{LSE} \\ LShoulderRoll = atan2(\boldsymbol{v''}_{y_{LSE}}, \boldsymbol{v''}_{x_{LSE}}) \end{cases} , \tag{8}$$

where $R_y(LShoulderPitch)$ is the rotation matrix in y by $LShoulderPitch$ degrees.

To complete the left shoulder rotation, only the x rotation is missing, that represents the rotation of the shoulder over itself. This rotation is applied on the robot elbow. For such, it is now considered the vector $\boldsymbol{v}_{LEH}$ from the elbow joint p_{LELB}, to the hand's p_{LHND}. Following this, both existing elbow rotations, in the coronal (roll rotation) $LElbowRoll$ and transverse (yaw rotation) planes $LElbowYaw$, are also calculated in relation to R_{ub}:

$$\begin{cases} \boldsymbol{v}_{LEH} = p_{LHND} - p_{LELB} \\ \boldsymbol{v'}_{LEH} = R_z(LShoulderRoll) \times R_y(LShoulderPitch) \times R_{ub}^{T} \times \boldsymbol{v}_{LEH} \\ LElbowRoll = atan2(\boldsymbol{v'}_{z_{LEH}}, -\boldsymbol{v'}_{y_{LEH}}) \\ \boldsymbol{v''}_{LEH} = R_x(LElbowRoll) \times \boldsymbol{v'}_{LEH} \\ LElbowYaw = atan2(\boldsymbol{v''}_{y_{LEH}}, \boldsymbol{v''}_{x_{LEH}}) \end{cases} . \tag{9}$$

For the extraction of the legs' joint rotations a new local coordinate system R_{lb} is defined. This new coordinate system R_{lb} use both the hip and the spine directions, as follows:

$$\begin{cases} R_{lb}^{y} = \boldsymbol{v}_{HIP} = p_{LHIP} - p_{RHIP} \\ R_{lb}^{z'} = \boldsymbol{v}_{SP} = p_{CSH} - p_{CHIP} \\ R_{lb}^{x} = R_{lb}^{y} \times R_{lb}^{z'} \\ R_{lb}^{z} = R_{lb}^{x} \times R_{lb}^{y} \\ R_{lb} = [norm(R_{lb}^{x}), norm(R_{lb}^{y}), norm(R_{lb}^{z})] \end{cases} . \tag{10}$$

The following steps describe the calculation of all joints presented in the robot's left leg, starting with the hip joint rotations, proceeding to the knees, and finally to the feet joints.. Firstly we extracted the robot's left hip roll rotation $LHipPitch$, that controls the hip movement along the body's sagittal plane:

$$\begin{cases} \boldsymbol{v}_{LKH} = p_{LHIP} - p_{LKNEE} \\ \boldsymbol{v'}_{LKH} = R_{lb}^{T} \times \boldsymbol{v}_{LKH} \\ LHipPitch = atan2(\boldsymbol{v'}_{x_{LKH}}, -\boldsymbol{v'}_{z_{LKH}}) \end{cases} . \tag{11}$$

Then the $LHipRoll$ is extracted:

$$\begin{cases} \boldsymbol{v''}_{LKH} = R_y(LHipPitch) \times \boldsymbol{v'}_{LKH} \\ LHipRoll = atan2(\boldsymbol{v''}_{x_{LKH}}, -\boldsymbol{v''}_{z_{LKH}}) \end{cases} . \tag{12}$$

Using this new vector, the last angle for the hip section $LHipYawPitch$, over the local z axis is determined. This hip freedom corresponds to an actuator that is shared by both legs. As such, this rotation defines the hip movement, symmetrically, for both legs:

$$\begin{cases} \boldsymbol{v}_{LAK} = p_{LKNEE} - p_{LANK} \\ \boldsymbol{v'}_{LAK} = R_x(LHipRoll) \times R_y(LHipPitch) \times R_{lb}^{T} \times \boldsymbol{v}_{LAK} \\ LHipYawPitch = atan2(\boldsymbol{v'}_{y_{LAK}}, -\boldsymbol{v'}_{x_{LAK}}) \end{cases} . \tag{13}$$

Having calculated the three rotations for the hip section we proceeded to the calculation of the knee and ankle (both knee and ankle only present 1 DoF), that enables rotations over the y axis of the local coordinate system. The robot body model also presents another rotation freedom at the ankles concerning its roll rotation around the z plane, $RAnkleRoll$ and $LAnkleRoll$. Yet, this rotation was not considered since it doesn't have any correspondence to our synthetic body model, being rather important for the maintenance of the robot's balance (which is outside the scope of this paper). As such, the respective pitch rotations of the left knee $LKneePitch$ and left ankle $LAnklePitch$ were calculated as follows:

$$\begin{cases} \boldsymbol{v''}_{LAK} = R_z(LHipYawPitch) \times \boldsymbol{v'}_{LAK} \\ LKneePitch = atan2(\boldsymbol{v''}_{x_{LAK}}, -\boldsymbol{v''}_{z_{LAK}}) \end{cases} , \tag{14}$$

$$\begin{cases} \boldsymbol{v}_{LFA} = p_{LANK} - p_{LFOOT} \\ \boldsymbol{v'}_{LFA} = R_y(LKneePitch) \times R_z(LHipYawPitch) \times R_x(LHipRoll) \\ \qquad\qquad \times R_y(LHipPitch) \times R_{lb}^{T} \times \boldsymbol{v}_{LFA} \\ LAnklePitch = atan2(\boldsymbol{v'}_{z_{LFA}}, -\boldsymbol{v'}_{x_{LFA}}) \end{cases} . \tag{15}$$

All the former processes are similarly applied to the right arm and leg of the robot.

3.3 Robot Motion Generation

The actual generation of the robot dance motion is accomplished by the cyclic repetition of the synthesized and adapted key-poses, generating this way the dance motion in the humanoid. The several key-poses are concatenated, and are synthesized with variability at quarter-beat resolution, according to the dance TGA representation. The obtained key-poses are ordered accordingly with the dance representation model (at the considered metrical resolution), and the poses concatenation is done by sine interpolation of the joint angles.

The beat-synchrony aspect of the dance was achieved by determining the dance motion frames where the determined key-poses occur and then calculating for each pose p the correspondent duration of the transition until the next pose $p+1$.

4 Evaluation and Results

4.1 Key-Poses Comparison

The following figure 5 present a comparison between the synthesized-adjusted key-poses and their representation in the simulated humanoid NAO.

4.2 Key-Poses Degree of Similarity

In order to compare the similarity between poses we defined a measure based on the evaluation of distances between the same joint positions among both poses, as follows:

$$s_{pose_{i,j},robot_{i,j}} = \sum_{j}^{n} \sum_{i=j}^{n} \left| d_{pose_{i,j}} - d_{robot_{i,j}} \right| \tag{16}$$

This similarity function aims to compare the feasibility of the humanoid reproduction of the adapted synthesized key-poses. For such, the distance between the whole joint positions of the adapted synthesized key-poses, $d_{pose_{i,j}}$ in (16), is compared with the actual humanoid joints position that resulted from the angles retargeted onto the robot, $d_{robot_{i,j}}$ in (16).

With this evaluation function, the joint positions of the adapted synthesized key-poses were extracted (noted as *synthPose*), and compared with the spatial joint positions of the same robot key-poses generated by: *i)* the respective joint angles extracted from the adapted synthesized poses, $s_{synthPose,poseAdapted}$; *ii)* setting all its joint angles to 0 (neutral pose), $s_{synthPose,poseNeutral}$; and *iii)* the angles extracted from synthesized key-poses without morphological adaptation, $s_{synthPose,poseNormal}$.

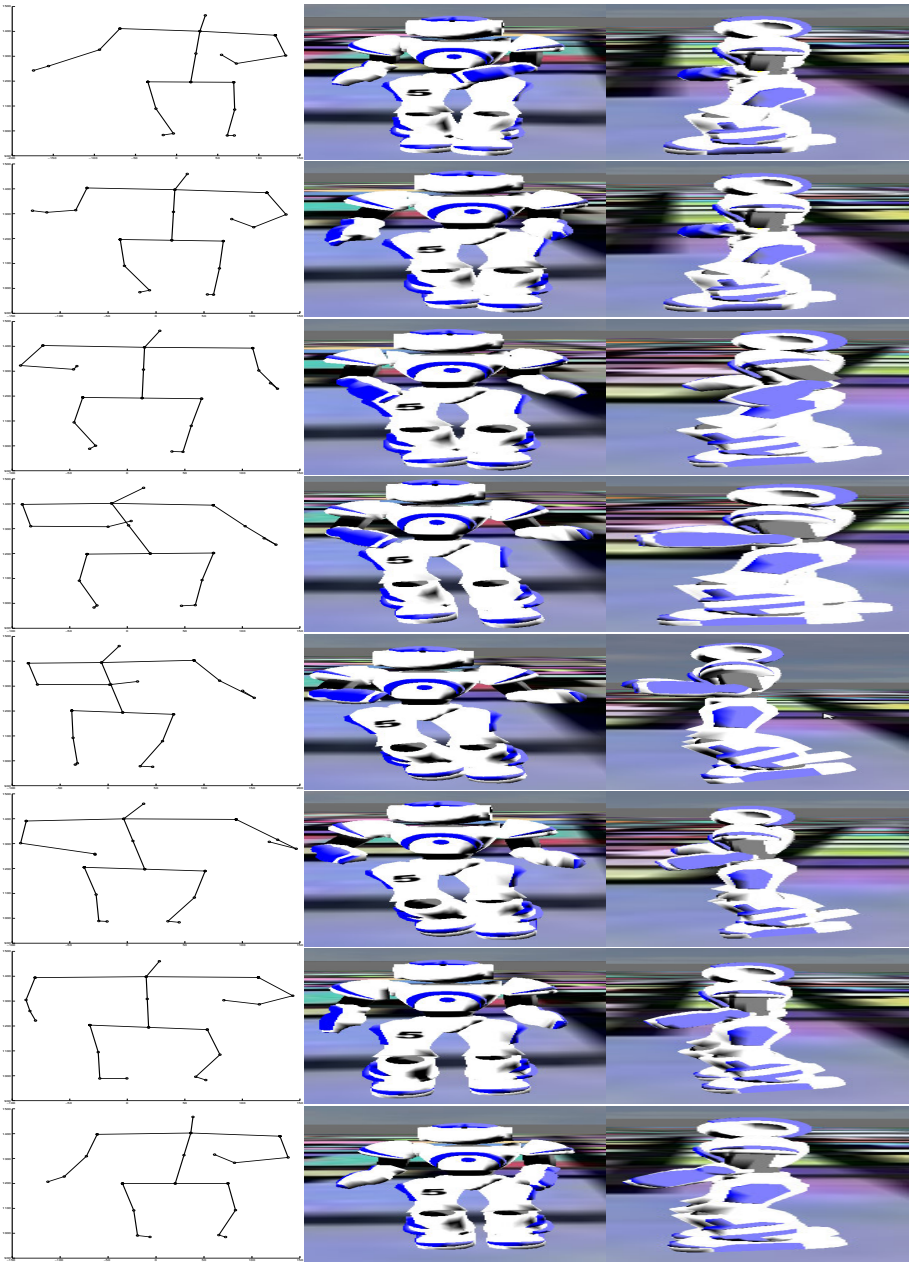

Fig. 5. Visualization of key-poses, from key-pose 1 (*top*) to key-pose 8 (*bottom*) (each row represents a new pose), synthesized at "variability-4": a) Synthesized-adjusted (*left*); b), c) Retargeted to simulated humanoid NAO, in frontal (*middle*) and lateral (*right*) views

Table 1. Similarity comparison of the generated poses, values in mm

	$s_{synthPose,poseAdapted}$			$s_{synthPose,poseNeutral}$			$s_{synthPose,poseNormal}$		
Pose	Arms	Legs	Total	Arms	Legs	Total	Arms	Legs	Total
1	571	543	2714	818	546	6300	328	637	3166
2	328	741	2911	489	934	6566	346	633	3146
3	182	1008	2847	494	826	6397	187	686	2776
4	393	739	2918	517	714	6687	434	676	3173
5	514	802	3177	516	819	6400	709	864	4041
6	368	912	3051	453	1081	7164	417	829	3199
7	322	1013	2912	424	750	6432	385	884	3065
8	220	579	1896	453	662	6762	230	579	2012

The results of these comparisons are presented in table 1.

4.3 Discussion

The method for resizing the body model by scaling and translating the joints' spherical distributions seems effective, presenting the desired result. The methods presented for changing the body morphology work separately, and aim to solve their specific problem without considering the global solution. This can signify that the achieved solution isn't the optimal solution, but rather a possible solution facing the individual constraints applied.

The key-pose angle extraction shows good overall similarity results. These results seem feasible, both numerically and visually, at the arms section, but the legs section seem to compromise it, especially in the hips. This difference can be explained by a bigger morphological similarity between the human's and the robot's arms than between the human's and robot's legs. The greater legs difference is mainly caused by the hip section due to the angular limits of the correspondent humanoid joint that makes impossible to achieve certain types of positions.

The results in table 1 shows better similarity results for the extracted angles from the adapted key-poses than from the non-adapted key-poses. In this table, the comparison with the neutral poses helps to have a sense of gain in similarity by the application of the angle extraction method, which seems to be in the order of 135%.

The proposed measure to evaluate the similarity between the synthesized and humanoid key-poses seemed to correctly qualify the degree of similarity and compare the different key-poses, but still lacks a zero value in order to get normalized similarity measures.

5 Conclusions and Future Work

In this study we proposed and evaluated methods for adjusting and retargeting human dance MoCap data, of Samba, into a target humanoid body model.

The process starts from information extracted from the analysis and representation of human dance MoCap data, and then generates random joint rotations that satisfy the desired model. Finally, the key-poses are extracted and reproduced into the target humanoid. The overall results seem to show that the methods used for the resizing and retargeting are valid, needing further evaluation with different humanoid morphologies and experimentation with MoCap data of different dance styles. In relation to the angle extraction method, it seems to be valid and obtains good results to the upper-body, but still only offers an approximation for the legs. The importance of the hip movements for the considered dance style, and the several morphological differences between the human hip and the robot hip, present obstacles that reduce the legs similarity. In a way to improve the degree of similarity of the humanoid poses with the original ones, the proposed similarity evaluation function may be also used as the fitness function in an optimization process. Finally, in order to obtain stable movements, motion refinement should be also performed, in order to assure the robot biped balance and avoiding self-collisions.

References

1. Simspark comunity, `http://simspark.sourceforge.net/wiki`
2. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Marnier, B., Serre, J., Serre, J.: The nao humanoid: a combination of performance and affordability. Computing Research Repository CoRR, abs/0807 3, 1–10 (2008)
3. Kim, S., Kim, C.H., You, B., Oh S.: Stable Whole-Body Motion Generation for Humanoid Robots to Imitate Human Motions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, pp. 2518–2524. IEEE (October 2009)
4. Kim, T.-h., Park Il, S., Shin, S.Y.: Rhythmic-Motion Synthesis based on Motion-Beat Analysis. ACM Transactions on Graphics 22(3), 392 (2003)
5. Lau, N., Reis, L.P.: High-level coordination methodologies in soccer robotics, robotic soccer, pp. 167–192. Itech Education and Publishing (2007)
6. Nagata, N., Okumoto, K., Iwai, D., Toro, F., Inokuchi, S.: Analysis and Synthesis of Latin Dance Using Motion Capture Data. Advances in Multimedia Information Processing 3333, 39–44 (2005)
7. Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., Ikeuchi, K.: Generating Whole Body Motions for a Biped Humanoid Robot from Captured Human Dances. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 3, pp. 3905–3910. IEEE (2003)
8. Nakazawa, A., Nakaoka, S., Ikeuchi, K., Yokoi, K.: Imitating human dance motions through motion structure analysis. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), pp. 2539–2544 (2002)
9. Naveda, L., Leman, M.: The spatiotemporal representation of dance and music gestures using Topological Gesture Analysis (TGA). Music Perception 28(1), 93–111 (2010)
10. Okamoto, T., Shiratori, T., Kudoh, S., Ikeuchi, K.: Temporal Scaling of Leg Motion for Music Feedback System of a Dancing Humanoid Robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, pp. 2256–2263 (2010)

11. Oliveira, J.L., Naveda, L., Gouyon, F., Leman, M., Reis, L.P.: Synthesis of Dancing Motions Based on a Compact Topological Representation of Dance Styles. In: Workshop on Robots and Musical Expressions (WRME) at IROS, Taipei, Taiwan (2010)
12. Oliveira, J.L., Naveda, L., Gouyon, F., Leman, M., Reis, L.P., Sousa, P.: A Spatiotemporal Method for Synthesizing Expressive Dance Movements of Virtual Humanoid Characters. Special Issue on Music Content Processing by and for Robots from the EURASIP Journal on Audio, Speech, and Music Processing (submitted to, 2011)
13. Lau, N., Reis, L.P.: FC Portugal 2001 Team Description: Flexible Teamwork and Configurable Strategy. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 515–518. Springer, Heidelberg (2002)
14. Ruchanurucks, M., Nakaoka, S., Kudoh, S., Ikeuchi, K.: Generation of Humanoid Robot Motions with Physical Constraints using Hierarchical B-Spline. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 674–679. IEEE (2005)
15. Ruchanurucks, M., Nakaoka, S., Kudoh, S., Ikeuchi, K.: Humanoid Robot Motion Generation with Sequential Physical Constraints. In: IEEE International Conference on Robotics and Automation (ICRA), Orlando, FL, USA, pp. 2649–2654. IEEE (2006)
16. Shiratori, T.: Synthesis of Dance Performance Based on Analyses of Human Motion and Music. Phd thesis, University of Tokyo (2006)
17. Shiratori, T., Kudoh, S., Nakaoka, S., Ikeuchi, K.: Temporal Scaling of Upper Body Motion for Sound Feedback System of a Dancing Humanoid Robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego CA, USA, pp. 3251–3257. IEEE (October 2007)
18. Shiratori, T., Nakazawa, A., Ikeuchi, K.: Dancing-to-Music Character Animation. In: EUROGRAPHICS, vol. 25, pp. 449–458 (September 2006)
19. Shiratori, T., Nakazawa, A., Ikeuchi, K.: Synthesizing Dance Performance using Musical and Motion Features. In: IEEE International Conference on Robotics and Automation, pp. 3654–3659. IEEE (2006)
20. Zhao, X., Huang, Q., Du, P., Wen, D., Li, K.: Humanoid Kinematics Mapping and Similarity Evaluation Based on Human Motion Capture. In: International Conference on Information Acquisition (ICIA), pp. 426–431. IEEE (2004)