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

Authors

Institute

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 data. Motion data of samba dance was synchronized to samba music manually annotated by experts in order to build, 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 a 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. The achieved results validated our approach, suggesting that it is possible to generate poses from motion capture and reproduce them on a humanoid robot with a good degree of similarity.

Keywords: Humanoid Robot Motion Generation and Motion Retargeting

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 an form of interaction and expression, that allows to enrich communication and interaction, improving humanoid robot motion expressiveness and realism is 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, 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. Processes are based on [9], this method was applied to motion data of samba dance style synchronized to samba music (manually annotated by experts), for building 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 morphologies. The method is tested for robot NAO [1], using the SimSpark simulation environment, 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 the motion analysis and 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 done. 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 referred techniques below use dance motion capture data and synthesize new motion from it. The first step in the process 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 allowing to determined key-poses to present real valor, and constitute meaningful representation of the dance, to the dance and so represent the dance. From this key-poses, the motion is then transfered to the robot, 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, its to interpolate transition motion between key-poses. So key-poses most show representative instances of the motion. The techniques to analyze and determine the appropriate key-poses are based on the analysis of the body motion [4] [3] or by analyzing the dance music [9], and in some cases a simultaneous analysis of both aspects [15] [16] [14] [8]. Working on Latin dances [4], 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, [6] and [5], segmented the dance, 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. [3] extracts rapid directional changes in motion. A combination of music and motion analysis is applied in [15] [16] [14] [8]. [15] identifies key-points of the motion rhythm as local minimums of the 'weight effort' (linear sum of rotation of each body joint), indicating stop motions and recognized as key-poses, and motion intensity points as the average of instant motion from the previous key pose. Its music analysis focus on the music intensity (sound whose spectral power is strongest between the neighboring frequency sound) and music rhythm, that is found by analyzing the repetition of several phrases and patterns present in the music structure. In [16], 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. Musical analysis is similar to [15], 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 [14] (for upper-body motion) and [8] (for leg motion). In [14] dance motion is captured at different speeds and the comparison of the variance in the motion, allows to observe that some poses stay preserved. This poses are considered key-poses since they tend to represent important moments to the music. The analysis in [8] is similar to [14] , but focusing on the analysis of the step motion. The determination of the key-poses is made 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 [9], spatiotemporal dance analysis model is presented, based on the Topological Gesture Analysis (TGA) [7], that conveys a discrete point-cloud representation of the dance. The model describes the spatiotemporal variability of the dance gestural trajectories in spherical distributions, according to the music metrical classes. 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 generated a spatiotemporal representation of the occupied points, by projecting musical cues onto spatial trajectories. We followed the [9] method of dance motion analysis since it conveys a parameterizable representation of the original dance, incorporating its intrinsic variability.

2.2 Dance Motion Generation

Dance Motion Generation techniques typically aim at generating motion from the key-poses extracted in the Motion Analysis phase. [11], [12], [2], [8] and [6] apply inverse kinematics to transform the markers position from the motion capture data into robot joint angles. In [2] 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. [8] checks the intervals between steps to keep a stable ZMP and then applies inverse kinematics to map the leg joint positions to robot joint angles. [11] applies optimization to ensure the physical restrictions of the robot, ensuring that limits of angles, velocities and acceleration are met. [12] applies sequential motion restrictions by optimization, limiting angles first, then solves collision avoidance and overcomes velocity and dynamic force constraints. Solving collision avoidance by increasing the critical distances for the periods that present collisions, [17] uses 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 ZMP criterion. [6] also limits angles and the angular velocity and finally modifies the ZMP trajectory in order to keep balance. [3] and [15] use motion graphs to represent the motion. [3] 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, [15], traces the motion graph based on the correlation between the music and motion features. Finally, [15] 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, [3] uses a real-time algorithm to adapt the motion to the target character. [14] applies optimization to overcome the joint angles limitations of the target character. [13] 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, [11] applies a phase of motion refinement to detect trajectory errors and correct them. On its hand, [8] 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 [9] we will base our approach in [13], 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 [9]. As such, we recurred to the same dance sequences of Afro-Brazilian samba, which were captured with a MoCap (Motion Capture) 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 [7] 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 [10], 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 [10], 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 p_{j-1}^m is the former calculated joint position, $l_{j-1,j}$ is the current segment length, 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 point cloud 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 size, joints' degree of freedom and other target kinematic physical constraints. 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. 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 $l'_{j-1,j}$:

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

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 new center point. The translation that happens from o_j^m to $o_j'^m$ is then applied. To all the following joint centers in the considered kinematic chain. This method allows to resize any body segment by manipulating the spherical distributions of the movement representation according th 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 as an adaption of the segment reach.

In order to have the correct body segment length, it may be necessary to remove joints that the target humanoid body doesn't have. In order to choose a point in D_j^m and D_{j+2}^m, eliminating D_{j+1}^m, we calculate the interception, in the form of the spherical cap C_j^m, of the spherical distribution for the target joint, D_{j+2}^m, with a sphere S_j^m centered in the position of the previous joint, p_j^m, and with radius equal to $l_{j,j+1} + l_{j+1,j+2}$. If C_j^m is a empty set, the center of D_{j+2}^m will be translated in the direction of the vector from p_j^m to o_{j+2}^m, increasing or decreasing the distance, from p_j^m to o_{j+2}^m. Allowing to obtain interception between D_{j+2}^m and S_j^m, finally a search for a point in C_j^m, that connected to p_j^m will be closer to the sphere center of the eliminated joint, o_{j+1}^m. In the special case where p_j^m is the anchor point of the model (and first point to be determined in the model), we can also move p_j^m, keeping it inside D_j^m, in the direction of the vector from p_j^m to o_{j+2}^m to enable the interception, and then move p_j^m, in order to approach him of to p_{j+1}^m to allow a better fit of the segment that will be traced.

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 $\overrightarrow{v}_{j,j'}^m$ and try to apply it to the other segment involved.

$$(p_j^m + \overrightarrow{v}_{j,j'}^m * l_{j,j'}) \in D_{j'}^m \tag{3}$$

$$(p_j^m + \overrightarrow{v}_{j,j'}^m * l_{j,j'}) \in D_{j'}^m \wedge (p_{j2}^m + \overrightarrow{v}_{j,j'}^m * l_{j2,j2'}) \in D_{j2'}^m \tag{4}$$

Condition 3 is the constraint applied to accept a generated point in [10], the algorithm will try to ensure this condition while not reaching the maximum number of tries. The condition 4 must be ensured instead, to use simetric vectors in both segments. If the method can't generate a vector that satisfies condition 4, then will generate a vector that can satisfies 3, and calculate a new point from it:

$$pf_{j2'}^m = p_{j2}^m + \overrightarrow{v}_{j,j'}^m * l_{j2,j2'} \tag{5}$$

This new point will be outside the spherical distribution of $j2'$, $D_{j2'}^m$, so the translation from $pf_{j2'}^m$ to the center of $D_{j2'}^m$ and assumed as the translation that must be applied to all the following spherical distribution center points in the kinematic chain. The translation will ensure that the remaining centers still maintain the spatial relation and until extreme of the chain the pose is still similar to the original.

Key-Poses Retargeting To generate the actual robot joint angles from the representation obtained from the application of the previous methods we applied a motion retargeting technique based on [13] for extracting the Euler angles of each joint, in the 3 dimensions, based on a body-centered axis system.

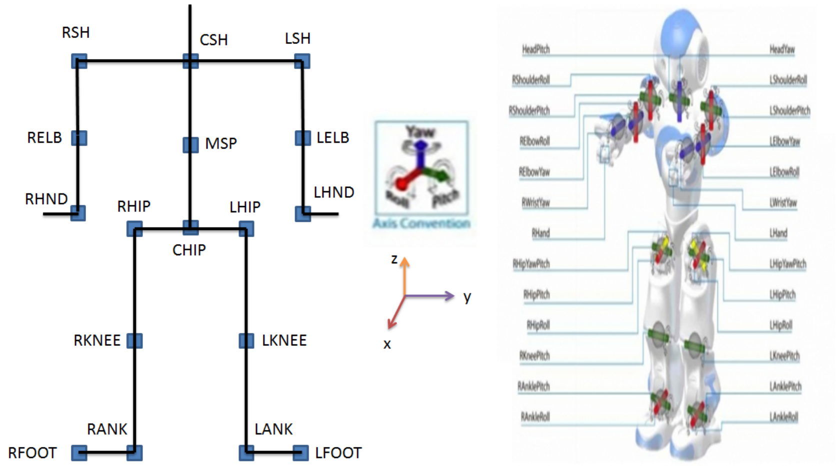

Fig. 1. Considered body models: (*left*) motion capture original; (*rigth*) humanoid robot NAO.

The starting point of this method is to determine a local coordinate system in the chest of the previously resized and adapted body model:

$$\begin{cases} y = p_{LSH} - p_{RSH} \\ z_{tmp} = p_{CSH} - p_{SHIP} \\ x = y \times z_{tmp} \\ z = x \times y \\ R_{axis} = [norm(x), norm(y), norm(z)], \end{cases} \tag{6}$$

where $norm(X) = \frac{X}{|X|}$ and $\times$ is the cross product between two dimensions.

Now, for each vector in the global coordinate system, we calculate the correspondent vector in the local coordinate system and calculate the angles in all the axis that we need. This calculation is specific to each segment of the considered humanoid morphology due to the presence of singularities of different natures. For the left arm we start at the upper arm segment $LELB$ by extracting the angle in the y axis (which moves the arm front or back), and retarget it to the pitch rotation of the robot's left shoulder $LShoulderPitch$:

$$\begin{cases} v_0 = p_{LELB} - p_{LSH} \\ v_1 = R_{axis}^T \times v_0 \\ LShoulderPitch = atan2(v_{1z}, v_{1x}). \end{cases} \tag{7}$$

The rotation of the left shoulder in z axis (moves the arm left or right), is extracted in the same way by applying the previous calculated y rotation to the previous vector v_1:

$$\begin{cases} v_2 = R_y(LShoulderPitch) \times v_1 \\ LShoulderRoll = atan2(v_{2y}, v_{2x}), \end{cases} \tag{8}$$

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

We then apply the same method to extract the angles for the left arm elbow:

$$\begin{cases} v_{3t} = p_{LHND} - p_{LELB} \\ v_3 = R_z(LShoulderRoll) \times R_y(LShoulderPitch) \times R_{axis}^T \times v_{3t} \\ LElbowRoll = atan2(v_{3z}, -v_{3y}) \\ v_4 = R_x(LElbowRoll) \times v_3 \\ LElbowYaw = atan2(v_{4y}, v_{4x}). \end{cases} \tag{9}$$

The extraction of the legs uses the same local coordinate system, R_{axis}, and starting by the angles for the hip joints proceeding to the knee and feet joint.

$$\begin{cases} v_{5t} = p_{LHIP} - p_{LKNEE} \\ v_5 = R_{axis}^T \times v_{5t} \end{cases} \tag{10}$$

$$\begin{cases} RHipRoll = atan2(v_{5z}, -v_{5z}) \\ v_6 = R_x(RHipRoll) \times v_5 \end{cases} \tag{11}$$

$$\begin{cases} RHipYawPitch = atan2(v_{6y}, v_{6x}) \\ v_7 = R_z(RHipYawPitch) \times v_6 \end{cases} \tag{12}$$

$$\begin{cases} RHipPitch = atan2(v_{7x}, -v_{7z}) \\ v_{8t} = p_{LKNEE} - p_{LANK} \\ v_8 = R_y(RHipPitch) \times R_z(RHipYawPitch) \times R_x(RHipRoll) \times \\ \qquad R_{axis}^T \times v_{8t} \end{cases} \tag{13}$$

$$\begin{cases} RKneePitch = atan2(v_{7x}, -v_{7z}) \\ v_9t = p_{LANK} - p_{LFOOT} \\ v_9 = R_y(RKneePitch) \times R_y(RHipPitch) \times R_z(RHipYawPitch) \\ \qquad \times R_x(RHipRoll) \times R_{axis}^T \times v_{9t} \end{cases} \tag{14}$$

$$RAnklePitch = atan2(-v_{7z}, v_{7x}) \tag{15}$$

The angle $RHipYawPitch$ isn't a simple rotation over z axis, and it's a shared joint with the rigth side, so it doesn't allow to easily give the desired orientation to the legs. Since the calculated angle is in the z axis, and the joint will move in z rotated by 45?, there was made an option to use the value closest to 0, between the calculated $RHipYawPitch$ and $LHipYawPitch$. Ultimately, the robot presents another joint, $RAnkleRoll$, that wans't determined in the calculations and was considered important only to the maintenance of balance, which is outside the scope of this paper. The joint angles for the right arm and leg are calculated in similar way.

4 Evaluation and Results

4.1 Key-poses Comparison

The following images present a comparison between the poses generated and it's representation in the NAO humanoid (see figure 2 and figure 3).

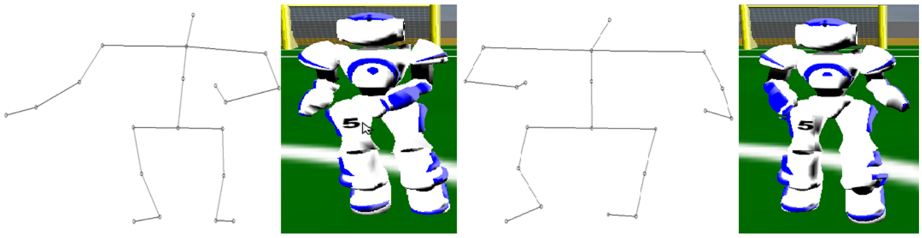

Fig. 2. Comparison of pose 1 (*two left images*). Comparison of pose 3 (*two right images*). In each set the *left image* is the generated by the method and the *right image* the reproduced in the robot.

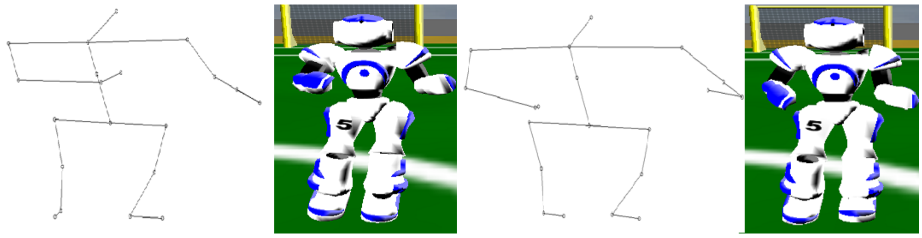

Fig. 3. Comparison of pose 5 (*two left images*). Comparison of pose 6 (*two right images*).

4.2 Level of Similarity

In order to compare the similarity we created a way to evaluate the similarity of the poses based on the evaluation of distances. With this we can evaluate the similarity of the generated pose points with the spatial position of the same body parts in the robot. For this, we will use a comparison between the distance of points in the pose and in the robot. For each pair of joint, j and i, we will determine the distance between them in the robot, $d_{robot_{i,j}}$, and in the expected distance of the pose, $d_{pose_{i,j}}$. Using this we will evaluate the similarity of the pose presented by the robot versus the expected as the sum of the difference between the two already calculated distances:

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

Using this measure, we calculated the similarity between the poses with morphological adaptation, without morphological adaptation, with resize of the body, without resize and finally pose generated only using the centers of the spherical distributions for each joint (see **??**). Since the some of the poses have different sizes the evaluation will have higher values, in order to have a measure of comparison, we considered the height, h_b, and the wingspan, w_b, of the different bodies and created a adjustment factor:

$$f = (h_b * w_b)/(h'_b * w'_b) \tag{17}$$

With this factor we aim to compare the similarity of the reproduction of the poses from a resized body and from a normal human body. The factor tries to determinate the linear relation between the max area covered by the different morphologies. Using both the evaluation function and the adjustment factor, we calculated the similarity between the generated key-poses and the reproduced key-poses in the humanoid. The similarity was calculated for the upper-body, arms mainly, and for the legs. This allows to demonstrate if the algorithm generates better angles for one part of the body. A third value considering all the body values was also calculated (presented in the 1).

Table 1. With Resizing and Morphological Adaptation (*left*); No method applied (*rigth*)

Pose	Arm	Leg	Total	Pose	Arm1	Leg	Total
1	459.566	666.27	2742.05	1	2768.45	3716.05	21657.68
2	373.065	1038.88	2742.05	2	2808.8	3499.0	21019.6
3	175.81	1038.88	4645.39	3	2766.87	3529.73	21116.14
4	437.592	987.604	3887.03	4	2837.17	3723.22	21116.14
5	584.048	583.348	3093.91	5	2689.28	3633.93	21871.87
6	452.228	734.822	2974.51	6	2807.12	3423.22	22211.47
7	326.499	550.647	1904.72	7	2579.38	3756.02	20365.76
8	276.419	599.648	2093.8	8	2713.37	3766.74	205648.66

In order to compare the values in both tables the adjust factor was calculated as 2.602 and all the values for the evaluation without any resizing were divided by it.

4.3 Discussion

Both the figures presented and the numerical evaluation of the similarity show good results for the arms posture, and a bit worst for the legs. This difference is mostly due to problems in the generation of the angles for the hip of the robot. In this area, we have 3 different joints that work in 3 different angles, while 2 of them control the rotation in x and y respectively. The third controls the movement in a z axis, rotated 45, and its shared by both legs, this makes the determination of the rotation a problem. We tried to minimize the value given to this joint, as explained, but in cases that both values are great, the joint will induce to much error in the pose. The principal error in the poses is in that section, leading to smaller but existent errors in the remaining joints of the leg.

The numerical evaluation of the similarity shows that there is a advantage on performing the morphological adaptation, with better similarity results for the poses that suffered the adaptation process. Either way, the results from this evaluation have some error associated, since the comparison on the robot presents variations on the positions even when the body is still. And the adaption factor used may not present the best relation to perform a comparison between the adapted poses and the non-adapted poses.

5 Conclusions and Future Work

In this study we proposed and evaluated methods for resizing a robot model and retargeting the joint trajectories. The process starts from information extracted from the dance motion analysis, and then generates random joint positions that satisfies the desired model, finally the key-poses are extracted and reproduced in the target humanoid. The overall results seem to show that the methods used for the resizing and retargeting work, needing further evaluation with a different method and appliance to other dance motions. In relation to the angle extraction method, it seems to be valid and obtains good results to the upper-body, but still only offers a approximate pose for the legs. The review of the extraction of the hip angles may prove to be a improvement in the method. In a form to improve the similarity the evaluation function may be used as the evaluation function in a optimization process. Finally, in order to obtain stable movements, motion refinement should be made, ensuring balance and avoiding collisions.

References

1. David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Brice Marnier, Julien Serre, and Julien Serre. The nao humanoid: a combination of performance and affordability. *Computing Research Repository CoRR*, abs/0807.3:1–10, 2008.
2. Seungsu Kim, ChangHwan Kim, Bumjae You, and Sangrok Oh. Stable Whole-Body Motion Generation for Humanoid Robots to Imitate Human Motions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2524, St. Louis, MO, USA, October 2009. Ieee.
3. Tae-hoon Kim, Sang Il Park, and Sung Yong Shin. Rhythmic-Motion Synthesis based on Motion-Beat Analysis. *ACM Transactions on Graphics*, 22(3):392, July 2003.

4. Noriko Nagata, Kazutaka Okumoto, Daisuke Iwai, Felipe Toro, and Seiji Inokuchi. Analysis and Synthesis of Latin Dance Using Motion Capture Data. *Advances in Multimedia Information Processing*, 3333:39–44, 2005.
5. Shin'ichiro Nakaoka, Atsushi Nakazawa, Kazuhito Yokoi, Hirohisa Hirukawa, and Katsushi Ikeuchi. Generating Whole Body Motions for a Biped Humanoid Robot from Captured Human Dances. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3905–3910 vol.3. Ieee, 2003.
6. Atsushi Nakazawa, Shinichiro Nakaoka, Katsushi Ikeuchi, and Kazuhito Yokoi. Imitating human dance motions through motion structure analysis. In *In Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 2539–2544, 2002.
7. Luiz Naveda and Marc Leman. The spatiotemporal representation of dance and music gestures using Topological Gesture Analysis (TGA). *Music Perception*, 28(1):93–111, 2010.
8. Takahiro Okamoto, Takaaki Shiratori, Shunsuke Kudoh, and Katsushi Ikeuchi. 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)*, pages 2256–2263, Taipei, Taiwan, 2010.
9. Removed. Removed for blind review.
10. Removed. Removed for blind review.
11. Miti Ruchanurucks, Shin'ichiro Nakaoka, Shunsuke Kudoh, and Katsushi Ikeuchi. Generation of Humanoid Robot Motions with Physical Constraints using Hierarchical B-Spline. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 674–679. Ieee, 2005.
12. Miti Ruchanurucks, Shin'ichiro Nakaoka, Shunsuke Kudoh, and Katsushi Ikeuchi. Humanoid Robot Motion Generation with Sequential Physical Constraints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2649–2654, Orlando, FL, USA, 2006. Ieee.
13. Takaaki Shiratori. *Synthesis of Dance Performance Based on Analyses of Human Motion and Music*. Phd thesis, University of Tokyo, 2006.
14. Takaaki Shiratori, Shunsuke Kudoh, Shin'ichiro Nakaoka, and Katsushi Ikeuchi. 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)*, pages 3251–3257, San Diego, CA, USA, October 2007. Ieee.
15. Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Dancing-to-Music Character Animation. In *EUROGRAPHICS*, volume 25, pages 449–458, September 2006.
16. Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Synthesizing Dance Performance using Musical and Motion Features. In *IEEE International Conference on Robotics and Automation*, pages 3654–3659. Ieee, 2006.
17. Xiaojun Zhao, Qiang Huang, Peng Du, Dongming Wen, and Kejie Li. Humanoid Kinematics Mapping and Similarity Evaluation Based on Human Motion Capture. In *International Conference on Information Acquisition (ICIA)*, pages 426–431. Ieee, 2004.