FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Humanizing Robot Dance Movements

Paulo Sousa

Master in Informatics and Computing Engineering

Supervisor: Luis Paulo Reis Ph.D

Second Supervisor: Fabien Gouyon Ph.D

Co-Supervisor: João Lobato Oliveira MSc.

18st July, 2011

Humanizing Robot Dance Movements

Paulo Sousa

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: António Fernando Vasconcelos Cunha Castro Coelho (Ph.D)
External Examiner: Luis Miguel Parreira Correia (Ph.D)
Supervisor: Luis Paulo Gonçalves dos Reis (Ph.D)

18^{st} July, 2011

Abstract

Robotic applications are increasing in a daily basis. Robots can and will be able to perform from simple day-life routines to complicated actions. There is, so, an increasing need of achieving a more complete and meaningful interaction between human and robots. Human interaction recurs to body motion as a form of communication and expression, and as such it presents a good field to exploit in order to improve humanoid robots.

Expressiveness and naturalness in robotic motions and behaviors can be replicated with the usage of captured human motion data.

As dance performances present moments of great expressiveness, where emotion and feelings can be transmitted to the audience by means of complex motion, they can be used as a rich case study and as a vast space of motion to replicate in a humanoid robot. As such, in order to feasibly and easily replicate these type of motions we recurred to captured human motion data of Samba dance, properly synchronized to Samba music, previously annotated by experts. Using this, a spatiotemporal representation of the dance motion was built in relation to the respective musical temporal structure (musical meter). This representation presents the variability of the dance space occupation during the dance performance, and its intrinsic musical relation. From this representation key-poses were synthesized with variability according to the original motion capture human body model, at defined metrical resolutions, by means of random processes calculated with rotations between body segments. In order to replicate the dance in the robot, the key-poses were morphologically resized and the original trajectories were adapted, overcoming the robot's varied kinematic constraints. From the synthesized and adapted key-poses, joint rotations were extracted to allow the reproduction of these same poses onto the target humanoid robot. Key-pose refinement was then applied in order to increase the similarity of the humanoid pose to the original key-poses, by means of Tabu Search optimization. Finally, the resultant robot dance motion was generated by sine-interpolation of the key-poses, at quarter-beat resolution and with variability, and performed at beat-synchronous velocity, to replicate the original musical relationship.

All the methods used in the process were numerically and visually tested, in terms of pose similarity with the original dance, and valid adaption of the represented body morphology. This was performed in order to evaluate the several steps of the process. Finally, an user-survey was also performed for obtaining subjective measures of the evinced degree of similarity and musical expressiveness between several dance motion excerpts, produced from the several methods. All tests and evaluation where performed on a simulated humanoid robot NAO in SimSpark, a generic robotics simulator.

The obtained results present a robot dance motion with a good degree of similarity with the original dance motion. This similarity was greater in the arms, since the robot arms and the human's are quite morphologically similar, but it was a bit compromised

in the robot's leg movements, due to especial restrictions on its hip section, which is of higher specific relevance to Samba dance. Ultimately, concerning the evinced degree of musical expressiveness, the vast majority of the inquired people validated the beat-synchrony of the robot dance performance. Yet, further studies are needed to evaluate the effect of introducing variability in the overall expressiveness of the resultant dance.

Resumo

As aplicações da robótica aumentam diariamente. Assim, os robôs podem e devem ser capazes de executar variadas acções, desde acções do dia-a-dia até a realização de acções mais complexas, complicadas ou delicadas. Portanto, existe uma necessidade de conseguir uma interacção mais completa e significativa entre humanos e robôs. Se atentarmos na interacção humana, esta pode ser realizada, ou complementada, por movimentos do corpo. Logo, dando a possibilidade de um robô humanóide realizar movimentos parecidos com os humanos iria melhorar a sua capacidade de interacção com os humanos. A expressividade e naturalidade dos movimentos robóticos podem ser melhoradas pela replicação de movimentos humanos.

A dança apresenta momentos de grande expressividade, que podem transmitir emoção e sentimento para quem assiste, e assim pode ser utilizado como um rico caso de estudo oferecendo um basto leque de movimentos a replicar num robô humanóide.

Para este efeito, e de forma a facilmente replicar movimentos de dança em humanóides, recorremos a dados de captura de movimento humanos relativos a dança de samba, estes dados foram sincronizados com a música que havia sido previamente anotada por especialistas. Com esta informação, uma representação espácio-temporal do movimento da música foi construída em relação a respectiva estrutura temporal da música. Esta representação apresenta a variabilidade inerente a uma actuação de dança e representa o espaço ocupado pelo bailarino durante uma actuação. Tendo como base esta representação, poses chaves foram construídas de acordo com o modelo do corpo humano presente nos dados de captura de movimento. Esta geração das poses foi feita em ciclos métricos definidos por meio da escolha aleatória de rotações entre os segmentos do corpo. De forma a reproduzir a dança no robot as poses foram adaptadas morfologicamente, de forma a representarem o tamanho do robot e a cumprirem as várias restrições físicas do mesmo. A partir das poses geradas e adaptadas os ângulos entre os vários segmentos do corpo foram extraídos possibilitando a reprodução das poses no robô humanóide. Estas poses forem depois refinadas de forma a aumentar a similaridade com as originais, para este efeito foi utilizada optimização com base no algoritmo Tabu Search. Finalmente o movimento de dança foi gerado por interpolação das poses, permitindo a execução síncrona do movimento de dança.

Todos os métodos utilizados foram testados numericamente e visualmente em relação à similaridade das poses e em relação a correcção da morfologia do corpo utilizado. Complementarmente um inquérito foi realizado de forma a obter uma avaliação subjectiva da dança realizada pelo robot. Todos os testes foram realizados com base numa versão simulada do humanóide NAO no ambiente de simulação Simspark.

Os resultados obtidos apresentam um robô a dançar Samba, com similaridade ao movimento de dança original. Esta similaridade é maior nos braços, enquanto nas pernas as

diferenças morfológicas na zona da cintura criam algumas diferenças no movimento e impossibilitam outros movimentos.

Acknowledgements

First and foremost i would like to thank to João Lobato for all the numerous conversations, orientation and advices that he gave me during the work of this thesis. I would also like to thank Luís Paulo Reis and Fabien Gouyon for the orientation, guidance and help in this thesis. A special thank to Luís Cruz for the help on the understanding of the humanoid NAO and in the help with the optimization framework. And also to everyone that helped with advices or conversations over my work. I also thank to everyone that helped in the visual evaluation of the results at the final stage of this work.

Last but not least, a special thank to my family and girlfriend for all the support, patience and help.

Paulo Sousa

Contents

List of Figures

List of Tables

Acronyms

DoF	Degree of Freedom
FK	Forward Kinematics
GA	Genetic Algorithm
HC	Hill Climbing
HMM	Hidden Markov model
HRI	Human-Robot Interaction
IK	Inverse Kinematics
LDA	Linear Discriminant Analysis
MoCap	Motion Capture
PSO	Particle Swarm Optimization
PFS	Partial Fourier Series
SA	Simulated Annealing
TCP	Transmission Control Protocol
TGA	Topological Gesture Analysis
TS	Tabu Search
UDP	User Datagram Protocol
ZMP	Zero Moment Point
XML	Extensible Markup Language

Chapter 1

Introduction

This thesis focus on mapping motion captured human movements onto humanoid robots. The goal is to generate Samba dance motion for a humanoid robot, for this motion samba motion data is analyzed to build a spatiotemporal representation. From this representation key-poses are generated, this poses will suffer morphological adaptation to ensure the humanoid kinematic constraints. Refinement of this key-poses is done to increase the similarity and in the end the key-poses will be interpolated in order to build the robot dance motion.

1.1 Motivation

Robotics applications grow daily, and the creation of realistic motion for humanoid robots increasingly plays a key role, as important forms of interaction between humanoid robots and humans already happen by means of non-verbal communication. This gives greater importance and interest on increasing the robot motion similarity to the different kinds of human motion behaviors.

As dance motion forms a rich, complex and expressive class of human motions it presents a good case study to help designing realistic forms of robotic motion. Dance movements also have a strong emotional meaning and expressive symbolism, making them powerful forms of non-verbal communication that would allow to improve human-robot social interaction by means of bodily communication. In this sense, the usage of human motion capture (MoCap) data provides a detailed description of the original movements trajectories, enabling an easier and more realistic replication of these motion by humanoid robots.

The proposed method, in this case applied to dance movements, can allow an easier and more detailed construction of rich and diverse movements for humanoid robots, with

different body morphologies. Such process can be easily extended to the learning of different humanoid movement patterns (repetitive motions), inherent to the performance of several simple daily-life tasks, such as biped locomotion or even basic sports' gaits, such as kicking or swimming. Not only would it give humanoid robots a greater amplitude of movements and possible actions, but would also introduce a new and simpler way to design, and in a certain way program, different kinds of robot movements and behaviors.

1.1.1 Applications

Further motivation for work in this area can be found in the several areas of the possible applications of such techniques:

- **Entertainment:** As we assist to the increasing role of robotics in entertainment areas, enhancing robotic movements, such as dancing, would help achieving more realistic and expressive behaviors, which would greatly enhance the amuse of the human-robot interaction (HRI).
- **Education:** The proposed approach can be used as the basis beyond choreographic tools, for robots and humans, and as an edutainment platform for captivating students from different backgrounds and ages to interdisciplinary themes such as dance, music, rhythm, robotics, kinematics and dynamics.
- **Research:** Many of the problems addressed are of great interest in researching, with especial focus on artificial (computational) musical cognition and motor embodiment, and generic issues beyond robotic motion kinematics and dynamics, applied to different humanoid body models.

As stated, the presented method was applied, as case-study, to dance movements, but could also be easily adapted to other different kinds of motion, increasing the number and types of potential future applications. Besides, the proposed method was designed for supporting different humanoid body models, which open application channels across different humanoid robotic platforms.

1.2 Goals

The aim of this thesis is to map motion captured human periodic motion, dance motion in this case, onto humanoid robots. In this thesis we will focus on the generation of dance motion, replicating different dance styles, mainly Samba.

In order to archive this main goal there are some more specific goals for this thesis:

- Study relevant kinematic techniques for generating robotics motion;

- Study and get familiarized with human motion capture systems and data;
- Analyze motion/musical relations in Traditional dance styles.
- Get familiarized with different choreographic methods for representing dance according to musical rhythm;
- Develop methods for adapting captured human motion data onto different humanoid morphologies;
- Synthesize the original dance motion assuring musical synchrony;
- Test and simulate the generated motion using a generic robotics simulator;
- Quantitatively and qualitatively evaluate the generated humanoid robot dance in terms of similarity with the original dance and degree of musical expressiveness.

The final result of this thesis should consist on several methods that allow the reproduction, by humanoid robots, of captured dance motion patterns, with especial on the kinematics of the motion.

1.3 Methodology and Tools

To achieve the proposed goals, firstly there should be a study of the different motion capture systems to understand the way as human motion is recorded. Afterwards we should analyze different motion data, previously captured from human dancing performances for extracting fundamental dance patterns in the original dance motion, and treat it in order to create a middle-term representation of the key-poses that will be mapped into the used humanoid robot. Knowing what poses the robot must perform, we need to adapt those movements from the human morphology to the robot morphology, having in mind the morphological differences between both bodies and the kinematic constraints imposed by the used humanoid body model. This transformation/transference should be tested in a proper robotics simulation environment towards keeping the overall aspect of the key-poses, and the relative relation of all the robot body parts. Ultimately, an overall optimization of the synthesized key-poses should take place in order to achieve greater similarity with the original motion.

In the following subsections, some topics, tools, and techniques that have interest to this thesis are presented, serving as base knowledge to the following work.

1.3.1 Motion Capture and Analysis

1.3.1.1 Motion Capture System

Motion capture is the term applied to the process of recording real motion and converting that motion onto a digital model. There are many kinds of motion capture systems, such as mechanical, magnetic and optical.

In a mechanical system, the performer wears a mechanical device, or exoskeleton, which was angle measuring equipments installed at the joints locations. As the performer moves, the sensors also move, measuring the joint angles of the performer, and so obtaining the orientation of the performer body parts. These systems are accurate and cheap, however it is not easy to perform fast and expressive motion due to the weight of the exoskeleton and the limited rand of the angle measurement devices [dA03].

Magnetic systems utilize sensors placed on the body and a known magnetic field is set up. The sensors and source are connected by a cable to an electronic control unit that calculates the locations and orientation of the limbs based on the generated magnetic field [Dic] . This technique enables real-time data collection and overcomes occlusion [dA03]. Yet, this type of systems has many limitations: it requires a physical connection (wires) with the sensors; it has a limited range; and sensors from different actors will interfere with each other. Making it harder to perform some movements.

The last type of motion capture systems are the optical ones. In these systems the performer wears a suit with special markers attached to it. The markers are retro-reflexive, and are placed in a way that the position of every body part can be easily acquired by several cameras surrounding the space where the performer moves. Each marker must be captured by at least 2 cameras, and a greater number of cameras diminishes the possibility of occlusions [Sch10]. Yet, for full body capture there must be 8 to 16 cameras (or more) [Dic]. The cameras will capture the reflex of the markers and the several images of a marker, from the various cameras, are matched using triangulation to compute the markers positions in 3D space [Dic]. The main advantages of optical systems are the very high rates of data collection and the possibility to create a great range of motion in a relatively large space [dA03]. Such systems can then capture larger areas, and are especially useful for capturing complex types of motion, such as sport actions or dancing [Dic].

Given the scope of this work upon dance movements, we used MoCap data of samba dance, performed by professional dancers, which was captured by an optical motion capture system. The used MoCap system was an Optitrack from Natural Point [Nat], recurring to 8 cameras positioned around the dancer.

1.3.1.2 MATLAB MoCap Toolbox

The MoCap Toolbox [TB10] is a Matlab toolbox that contains functions for the analysis and visualization of motion capture data. The toolbox is mainly aimed for the analysis of music-related movement, but might be useful in other areas of study as well [TB10].

It will be used for analyzing the kinematics of the original human body motion trajectories and studying its musical relationship, obtaining this way a middle-term representation of the original dance style.

1.3.2 Musical Rhythmic Qualities

Musical meter represents the temporal regularity present in the music [Kla03]. The musical meter concept embraces the idea that music is organized in a hierarchical structure of rhythm and temporal regularities of different (typically three) metrical-levels (temporal resolutions). As illustrated in figure 1.1, these level are often decomposed into tatum, that represents the lowest, and fastest, level, typically following the occurrence of the musical note events; tactus (or the actual beat), which is the most prominent level, also described as the foot-tapping rate; and the musical measure (or bar) which consists on the highest (slowest) metrical-level and represents the boudaries of a given rhythmic pattern.

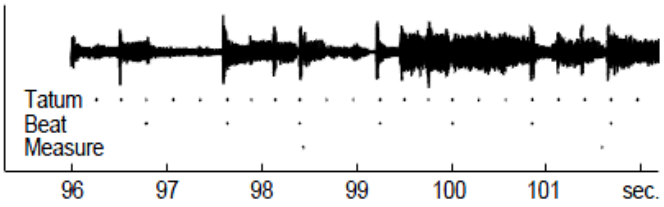


Figure 1.1: Illustration of the hierarchical musical rhythmic metrical-levels [Kla03].

Beats represent music regular rhythmic pattern, and the group made by strong and weak beats is called a meter. Beat induction refers to the human search of periodical occurrences of music events [GWSF06]. Pulse sensations describe all the rhythmic levels invoked in the mind of the listener [Par94]. And tempo defines the rate of the beats in a metrical level [GD05]. The perceiving of musical meter can be characterized as processing the musical events to detect the underlying periodicities [LJJ96].

Musical meter also organizes dance choreographies, the timing of gestures and the music structure [ONG+11]. When dance gestures are synchronized with the musical meter, then dance is integrated with meter in the spatiotemporal domain. Then as music and dance share the same time domain, regular events in the music are reflected in the use of space in the dance [ONG+11]. Figure 1.2 illustrates the projection of dance trajectories onto the spatiotemporal domain.

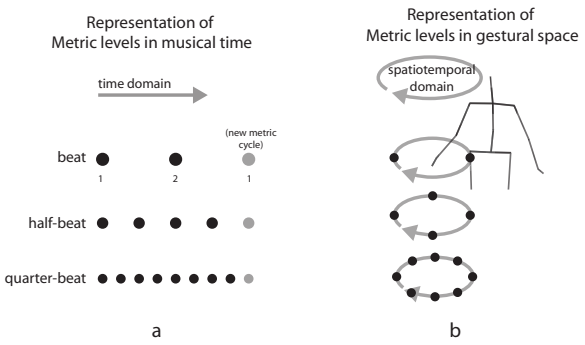


Figure 1.2: (a) Hierarchical representation of meter structure. Each hierarchical metric level is then subdivided, or grouped, in other levels. (b) Spatiotemporal representation of metric accents in a dance gesture with a period of 2 beats. [ONG+11]

1.3.3 Kinematics

Kinematics is the formal description of motion, and the study of motion of a body or system of bodies. A humanoid robot can be considered as a set of bodies connected by joints. While dynamics is the study of forces and why objects are in motion, in other words, dynamics is the study of the causes and changes in motion. There are two main branches of kinematics: Inverse and Forward Kinematics. Inverse kinematics is the process of finding the angles that a joint should have to be in the desired pose. Forward kinematics allows to find the position of a given body part using the actual position of the joints and its angles.

Kinematics of human motion are also relevant to refer, specially techniques used for the description of the relative orientation between adjacent body segments. This relative orientation can be regarded as a combination of translation and rotation, in our case the main interest is purely the rotation. Two ways to represent rotation of the body segments are Euler angles and Quaternions. Euler angles describe the rotation over tree axis, the order of the appliance of the rotations matters, since different values/combinations can give the same rotation. Euler angles suffer from the gimbal lock problem, that occurs when two rotational axis of an object are pointing in the same direction, causing the rotation to loose one degree of freedom (DOF). Quaternions also can represent the rotation between two segments, rotation quaternions describe the rotation using three imaginary parts and a real part. The three imaginary parts represent the axis of rotation and the forth element represents the amount of rotation.

In the description of human motion, it is also important to consider the anatomical description of the motion according to the sagittal, frontal, or coronal, and transverse planes, that define the plane in which the motion is performed [Zat98] 1.3.

By following figure 1.3, the transverse plane is parallel to the ground and separates the upper-body from the lower-body, perpendicular to the coronal and sagittal planes.

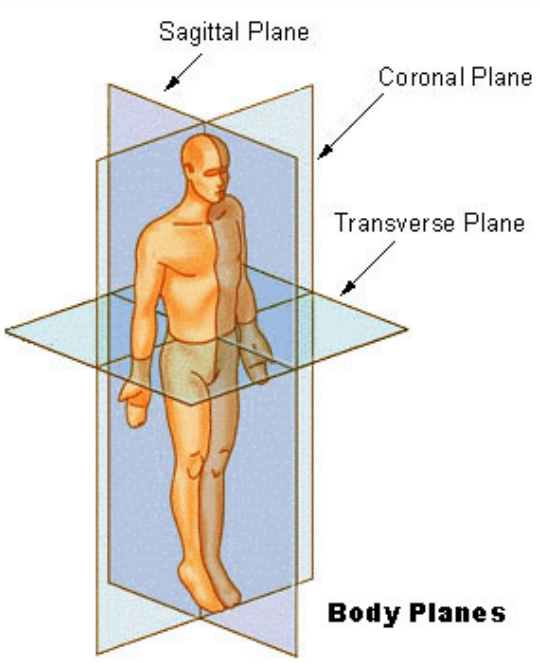


Figure 1.3: Body planes positions relative the human body. [Hea]

The coronal, or frontal, plane is a Y-X plane perpendicular to the ground and separates the front from the back of the body. The sagittal plane is a Y-Z plane also perpendicular to the ground, which separates the left from the right parts of the body [Zat98]. By following Euler notations, the body rotations may then be defined as pitch, yaw and roll, representing the rotation axes of the body respectively in the sagittal, transverse, and coronal planes.

Another important concept are kinematic chains, that consist in the linkage of rigid bodies. A human leg or arm can be considered examples of kinematic chains [Zat98]. Finally, the degrees-of-freedom describe the possible independent directions in which a body can move in the 3-dimensional space [ON91].

1.3.4 Robotic Platforms and Simulators

Robotic platforms are usually expensive leading to the necessity of the usage of simulation environments that provide numerous advantages. Simulation platforms are less expensive, allow the easier development and testing and allow the usage of detailed information from the simulation. The usage of a simulation platforms for the present thesis takes full advantage of all the aspects described.

1.3.4.1 SimSpark

SimSpark is a generical simulator, that supports developing physical simulations and robotics research. Different agents can participate in one simulation, connecting to SimSpark using UDP or TCP. SimSpark uses the Open Dynamics Engine (ODE) for simulating rigid body dynamics. It is commonly used in academic research and education. [OR04]

It will allow simulating and debugging the motion developed for robot NAO. It will serve as platform for the experimentation on a simulated humanoid NAO.

1.3.4.2 NAO

Robot NAO (see figure 1.4) is a humanoid robot developed by Alderbaran Robotics. The NAO weighs 4.5kg and has 57cm of height. Both real (figure 1.4 *a)*) and simulated versions (figure 1.4 *b)*) are endowed with 21 DoFs. As depicted in figure 1.4 (*c)*), it has five DoFs in each leg, two in the ankle, two (plus one, in a complex structure) in the hip, and one at in the knee [GHB+08].

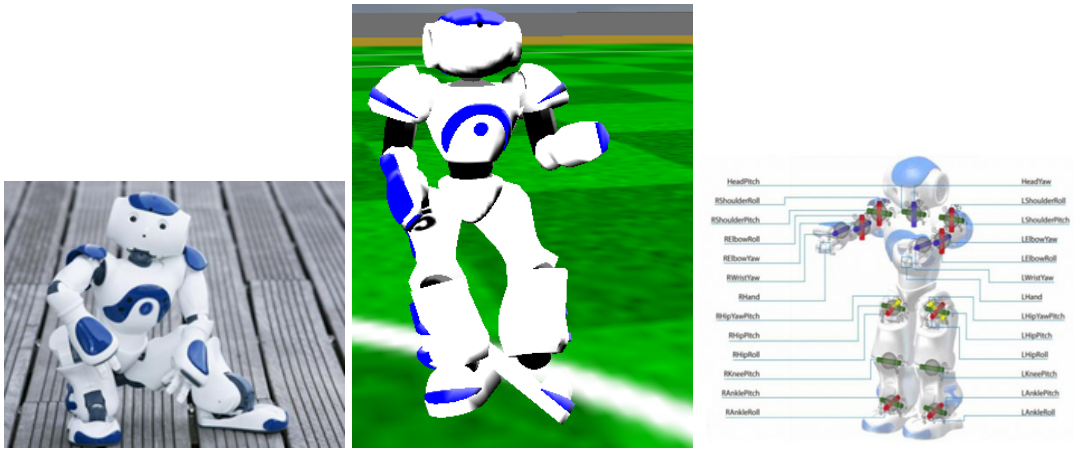


Figure 1.4: Humanoid robot NAO. *a)* Real robot. *middle)* Simulated robot. *right)* Kinematic body model description.

It is based on a Linux platform and scripted with URBI. It also provides interfaces for C++, Java, Matlab, Python and Ruby. Additionally, the real NAO is also equipped with Wi-Fi, which allows remote control in order to improve its CPU capacity.

The NAO simulation agent is based on a state machine ruled by the execution of predefined behaviors. The skills, or behaviors, consist in actions that the agent may execute [Rei10]. These behaviors are defined in Extensible Markup Language (XML) files, providing the parameters for the behavior execution. This definition is composed by the name of the behaviors and the type of the behaviors. In the case of Slot behaviors type (exemplified in figure 1.5), the XML will also assign a final value to each robotic joint

and a duration for this movement, a behavior can have several states, or slots, that will be executed in the defined order and in the defined time.

```
<behavior name="sambaKeypose_tw" type="SlotBehavior">
    <slot name="sambaKeypose_1" delta="0.420000">
        <move id="&rarm4;" angle="40.956333" />
        <move id="&head1;" angle="-176.612545" />
        <move id="&head2;" angle="108.579088" />
        <move id="&lleg1;" angle="-10.528882" />
        <move id="&lleg2;" angle="-5.374154" />
        <move id="&lleg3;" angle="8.077894" />
        <move id="&lleg4;" angle="-33.485049" />
        <move id="&lleg5;" angle="17.169619" />
    </slot>
    <slot name="sambaKeypose_2" delta="0.340000">
        <move id="&larm1;" angle="-98.927806" />
        <move id="&larm2;" angle="-10.059020" />
        <move id="&larm3;" angle="-84.428486" />
        <move id="&rleg1;" angle="47.371126" />
        <move id="&rleg2;" angle="8.429384" />
        <move id="&rleg3;" angle="2.927514" />
        <move id="&rleg4;" angle="-29.587846" />
    </slot>
</behavior>
```

Figure 1.5: Example of a simple Slot Behavior.

1.3.5 Optimization

Optimization tries to find the best element of a set of alternatives, in other words, it is a search for the best solution possible. This search can be done taking in consideration several parameters and also several constraints of the problem.

1.3.5.1 Hill Climbing

Hill Climbing (HC) is a simple optimization technique and performs well in several situations. To find a solution this algorithm tries to find a better solution by changing a single element of the current solution. If the change in the current solution produces a better solution, this new solution is used as current solution and another change is done to this solution. The algorithm will incrementally change the solution until no further improvements to the current solution can be found.

It is useful when the amount of time to perform a search is limited and when the desired, or the best, solution is close to the current solution in the search space and there is no local optimum between them.

1.3.5.2 Simulated Annealing

Simulated Annealing (SA) [KGV83] is inspired in a metallurgy technique involving heating and controlled cooling of a material. This slow and controlled cooling process is known as annealing.

By analogy with this metallurgic process, the SA algorithm takes the current solution to a problem and replaces it with a new random one that is close to the current solution. This random choosing of a new solution is done using a probability that depends on the difference between the solutions and on a global parameter, the temperature that decreases during the search process. With larger temperatures the solution changes more randomly. This allows that in the beginning the method avoids becoming stuck at local optima. In contrast to HC, that only updates the solution when finds a better one, in SA the new solution can be worse than the current solution.

1.3.5.3 Tabu Search

Tabu Search (TS) [Glo86], was proposed in 1986 by Fred Golver. In TS, a list of all the visited solutions is kept, called to tabu list. This avoids visiting twice the same solution.

In the algorithm, each visited node is declared as a tabu and is placed in the tabu list. And then the algorithm searches the neighboring nodes that still aren't in the tabu list. TS will test all the possible solutions and in the end chose the best one.

1.3.5.4 Particle Swarm

Particle Swarm Optimization (PSO) [Ebe06] is based on an evolutionary algorithm. PSO tries to optimize a solution by iteratively trying to improve a candidate solution, with regard to the solution evaluation. Starting with a set of randomly generated solutions, PSO then searches for a optima solution by upgrading the solutions. Each particle has two characteristics: position and velocity. The potential solutions, or particles, will wander around the problem space, and always remembers the best position visited, and the solution value to that position. The particles can communicate with which other and adapt their velocity or position to the information received. During the several iterations of the algorithm the particles may converge to the optimum or diverge from it.

1.3.5.5 Genetic Algorithm

Genetic Algorithm (GA) [Hol75] is an optimization method inspired, in a certain way, in the natural reproduction system and in the evolution of biological systems. In a GA, the initial population, that may be called chromosomes), is optimized toward a better solution. Each solution, or chromosome, is nothing else but a set of genes.

The initial solution can be calculated randomly or by a creation function, it only needs to be an acceptable solution to the problem. Using this initial population, the algorithm starts and creates new populations using selection, crossover and mutation operations. Selection will specify the parents to the next generation. Crossover will generate a new child from the two parents; the child will inherit genes from the parents. Mutation produces mutation of children, in order to include random variations on the children's genes.

1.4 Research Institutions

This thesis wad done at LIACC, under the supervising of Ph.D Luis Paulo Reis, and at INESC Porto, under the supervising of Ph.D Fabien Gouyon. It also counted with the co-supervision of João Lobato Oliveira, from both institutions

LIACC (Laboratory of Artificial Intelligence and Computer Science of the University of Porto) was created in 1988 in order to promote the collaboration of researchers that were separately working in the fields of Computer Science and Artificial Intelligence in different Faculties. It aims to help solving general computer science problems, from security to software reliability. LIACC works in different research areas: Advanced Programming Systems, Distributed Artificial Intelligence and Robotics, Formal Models of Computation, and Language, Complexity and Cryptography.

INESC (Institute for Systems and Computer Engineering) of Porto, is an interface institution between the academic world and the world of industry and services, as well as the public administration. The area of activity ranges from research and development, to technology transfer, consulting and advanced training. The main research areas are Telecommunications and Multimedia, Power Systems, Manufacturing Systems Engineering, Information and Communication Systems, and Optoelectronics.

1.5 Thesis Outline

The remainder of this thesis is divided in four chapters. Section II describes some related work to this thesis, and several approaches in the area. Section III presents the proposed methodology and describes the developed work. In Section IV the obtained results are discussed. Finally in Section V the conclusions are depicted and some paths for the future work are presented.

Chapter 2

Related Work

This chapter describes the state-of-the-art in this area, and also presents some related works and their methodology expressing the results and problems that they faced. This will help to identify the main difficulties that can appear in the project and also some of the problems that this project can resolve.

One of the main topics of interest for this work is the analysis of motion data, concerning the methods used to analyze and mine the motion capture data. In other words, the way that the information is treated, the tools or methods used and also some specifications of each approach.

Another interesting topic in this area is how that motion can be then synthesized and replicated by different humanoid models. Here is useful to study robotic approaches, and methods used in computer animated systems, for laboring the extracted motion data until replicating it by the chosen humanoid model. Given this problem's relevance in the literature, it is important to also look at the results obtained and the problems faced by the recent approaches from the literature. In relation to robotic motion generation, it is also useful to study some applications of optimization or refinement of movement. Specifically to this case, it is of greater interest to look at works that use the same robotic platform intended for this project, robot NAO.

In the end, a small review of all the work is made, presenting some of the problems and challenges found and faced by similar works. This is important in order to propose some novel approaches to some of the related problems.

2.1 Dance Motion Analysis

As already stated, the first goal is to use motion analysis in order to search and work with captured human dance motion data, and this way this way build a middle-term represen-

tation of the dance motion. Several methods have been used for analysis of dance motion captured data: some only analyze motion data of specific parts of the human body while others analyze all the body; and some only consider the body motion while others also take into account its musical relation.

Firstly, some methods regarding only motion analysis are presented and after it methods that use music and motion analysis to keep their intrinsic synchronism.

Nagata et al. takes basic knowledge from the main characteristics of a Latin dance [NOI+05]. Using that, they determine what information of the motion data will be extracted to represent the dance. Information related to the movement of shoulders and hips were extracted and analyzed. In particular, the rotation related to a plane vertical to the floor and related to a plane horizontal and parallel to the floor were calculated.

Nakaoka et al., [NNIY02] and [NNY+03], 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.

Different approaches that try to analyze, study and generate dance motion well matched to music are also important and of even greater interest, since there is a close relationship between musical rhythm and motion rhythm.

Siratori et al. uses a transformation of the original motion capture data to a simple human body model [SNI06a]. At each frame the human pose, from the motion capture data, is converted into a body center coordinate system, were the origin is the waist of the human model, the x-coordinate is the direction from the left to the right thigh, the y-coordinate is the forward direction of the body, and the z-coordinate is a vertical upper direction (see figure 2.1). Using this coordinate system each body segment is converted to vector for further analysis. Then, for each moment of the motion they calculate two different attributes: the motion rhythm and the motion intensity (see figure 2.1). For this, a "weight effort" component is calculated, physically considered as the linear sum of rotation velocity of each body joint. The local minimums of the weight effort component are used to extract the motion rhythm, as indicators of stop motions, which are considered impressive instances of the dance performance, being so recognized as key-poses. The motion intensity is calculated as the average of the instant motion from the previous key-pose. After analyzing the motion, further analysis is performed in order to recognize the matching musical features such as: music rhythm and music intensity. The first component is determined using the knowledge that the music structure consists of the repetition of several phrases, and that the music is segmented by repeated patterns. To extract the intensity of the music, they calculated the sound chunk whose spectral power is strongest among the neighboring frequency sounds.

Shiratori et al. also considered two components as the motion features: key-frames and intensity (see figure 2.2) [SNI06b], and another three as music features: music beat and degree of cord changes for beat structure analysis, and music intensity for mood

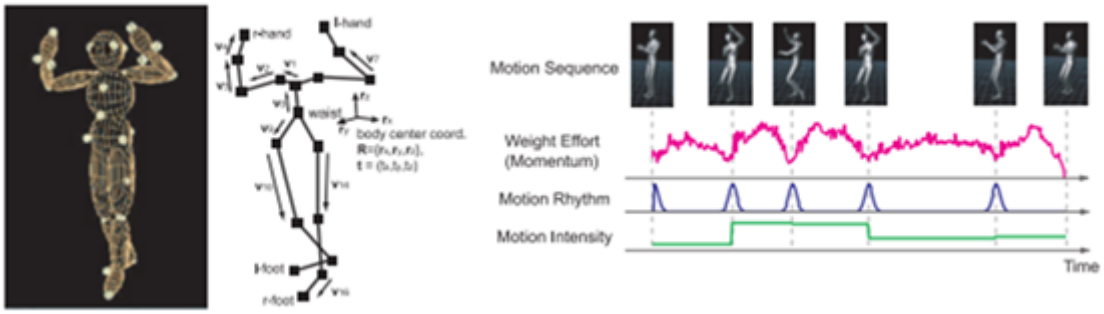


Figure 2.1: MoCap dance motion analysis [SNI06a]: a) Simple human body model (*right*); b) Extracted motion features (*left*).

analysis. For the motion key-frame component, they assumed that key-frames are 'stop' frames of the hands' motion, and so they can be determined by finding the local minimum of the hands' velocity. The motion intensity is assumed as the difference in the velocity of the hands in two neighboring key-frames, so greater velocities mean high motion intensity. They just look for the maximum values of this difference in velocity. The extraction of the music intensity is done in a similar way to the last approach [SNI06a]. To extract the musical rhythmic features, a frequency analysis is performed, and, finally, a beat analysis.

Shiratori et al. also focus on temporal scaling techniques for upper-body [SKNI07] and for leg motion [OSKI10]. In [SKNI07], dance motion is captured at three different speeds, and then a hierarchical B-spline interpolation is applied for controlling the frequency resolution, by setting control points at temporal intervals, using for the intervals the musical rhythmic features. Comparing the variance of the motion in the same moment of the music, they found that there are some valleys where all variance sequences have a local minimum. It was demonstrated that even at different speeds, there are some postures that stay preserved in the dance, which represent the important stop motions, or key-poses, of the original dance (figure 2.3).

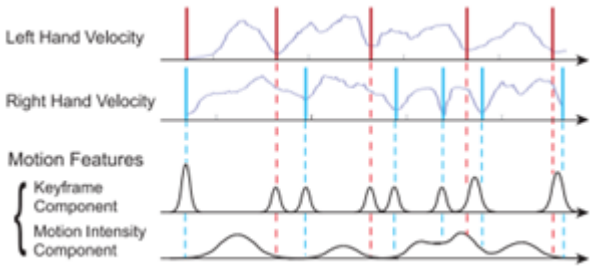


Figure 2.2: MoCap dance motion analysis [SNI06b]: graph of motion features according to the hands' trajectories.

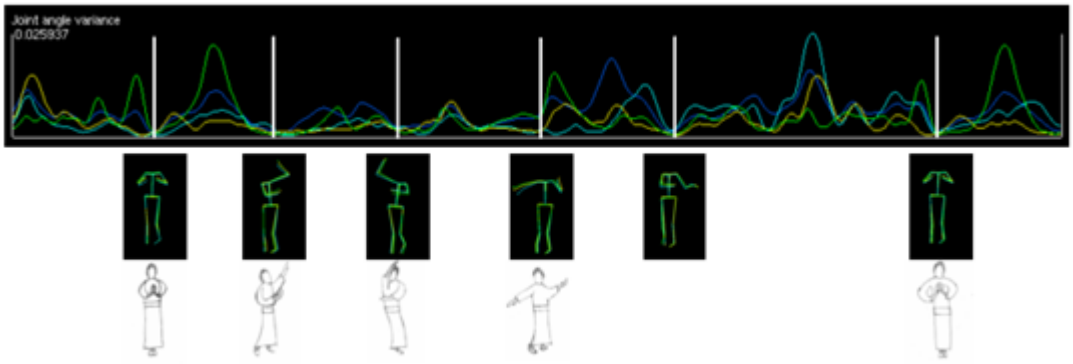


Figure 2.3: a) Joint angles variance performed at different musical speeds (*top row*); b) Extracted key-poses (*middle row*); c) Important stop-motions, drawn by professional choreographers (*bottom row*) [SKNI07].

In [OSKI10], *Shiratori et al.* proposes a similar method to [SKNI07], but now adapted to the leg motion. The concept of key-pose is also used, but now the analysis focus on step motions. From the observation of the motion in the different velocities, they found that the time and stride of the step, near a key-pose, tends to be maintained, that a kicking action with the increase of velocity tends to look like a normal step, and that the speed of a swing foot won't accelerate in the same proportion as the increase of velocity of the music. All this indicates that the dancers made more effort to maintain the original timings for tasks around key-poses and that the dancers tried to maintain stride length as close to the original one as possible around key-pose timings.

Ofli et al. used 3 Hidden Markov model (HMM's) to capture the dynamic behavior of the dancing body trajectories [OCFT+08]. One models the motion of the torso, other the movement of the arms and a third for the movement of the legs. For the music analysis, the tempo and the relevant beat information is used to drive the movement. Tempo is estimated in terms of beat per minute, and the beat location is computed from periodicity estimation. For motion analysis, the start and end frames of each dance figure are manually labeled, and then the three HMM models of each dance figure are trained in a supervised manner with the body posture parameters captured from those manually labeled segments.

Kim et al. extracts rapid directional changes in a motion [KPS03]. These moments are detected from the zero-crossing of the second derivative of each motion signal at every frame. So they extract the zero-crossing moments of the joint orientation signals, and classify these moments as candidates for motion beats. The sequence of candidates extracted from a motion signal embeds a periodic pattern. From this sequence are calculated some reference beats, and then estimated the actual motion beats from the candidates using these reference beats as guide (figure 2.4).

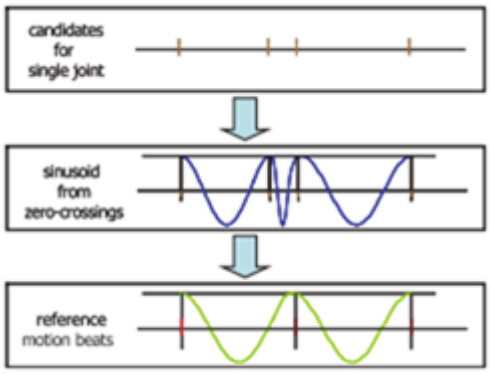


Figure 2.4: Estimation of reference motion beats according to joint motion cues, in [KPS03].

Oliveira et al., in [ONG+10] [LOGPR09] [LOGPR08], presents a spatiotemporal dance analysis model, based on the Topological Gesture Analysis (TGA) [NL10], 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 respective musical rhythmic metrical classes, at different resolutions (metrical levels).

2.2 Motion Synthesis and Generation

In the area of motion generation from motion capture data, most literature focus on the generation of motion to animated characters, from simple animated avatars, to show the results of the previous motion analysis, to the generation of motion to animate virtual figures. It is interesting to look at these results as they have many phases in common with the generation of motion to humanoid robots. It is also important and interesting to look mostly at works that also use dance motion as case study.

2.2.1 Computer Animated Systems

Ofli et al., in the synthesis phase [OCFT+08] aims to generate the corresponding body posture synchronized with the test musical audio signal. Initially the audio signal is classified in respect to its genre. The audio is then analyzed, and with the extraction of the tempo, beat information and genre the motion is performed off-line. The genre determines the dance figure to be synthesized, whereas the tempo and beat determine the duration and location of the figure. The body posture parameters corresponding to each dance figure are generated using the associated HMM structures learnt at the motion analysis stage. In the end, the avatar is able to recognize the genre of the dance and to perform concordant

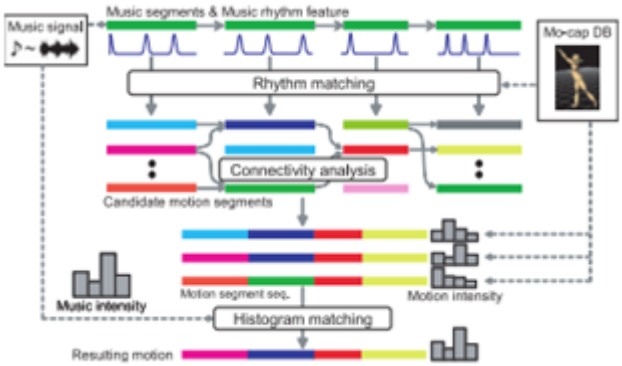


Figure 2.5: Overview of the dance motion synthesis algorithm proposed by [SNI06a].

dance figures. The main problems with this method is the necessity of supervised learning of the agent and an accurate capture of the motion of the dancer used in the respective teaching.

Nagata et al., as a way to confirm that there is isolation between shoulders and hips as observed in the motion analysis [NOI+05], created a dance animation using the motion data and the information extracted in the analysis phase. The created animation shows the dance in good detail. This method extracts little information from the motion data, and so don't allow a realistic representation of every dance.

Shiratori et al., using all the information extracted during the previous phase - music and motion analysis [SNI06a] - tries to synthesize a new dance motion (following the algorithm presented in figure 2.5). Initially the rhythm components are evaluated and candidate motion segments are detected in correspondence to each music segment. Then, it is checked if the synthesized transition motion between the neighboring motion segments looks natural, and possible sequences of motion segments are extracted. The end step in the generation is to analyze the similarity of the intensity components between the music segments and the selected motion segment sequences, and then synthesize new dance motions by concatenating the motion segments with each other.

The music analysis is successful in the majority of the cases, but not every time, failing to music genres that don't contain repeating melody lines. The synthesis phase is successful and the resulting motions are well matched to the input music.

Concerning the motion/music synchrony, *Shiratori et al.* sets the extracted motion features [SNI06b] in a motion graph, and a new dance motion is synthesized by calculating the correlation of the music features and the motion features, and the motion graph is traced based on the correlation results. To create this transition motion, were used 3rd order interpolation of body links, considering the smoothness of position, velocity and acceleration. The duration of each transition is calculated by the angular distance of the frames and the maximum velocity in the concatenated motions. In the analysis, the max-

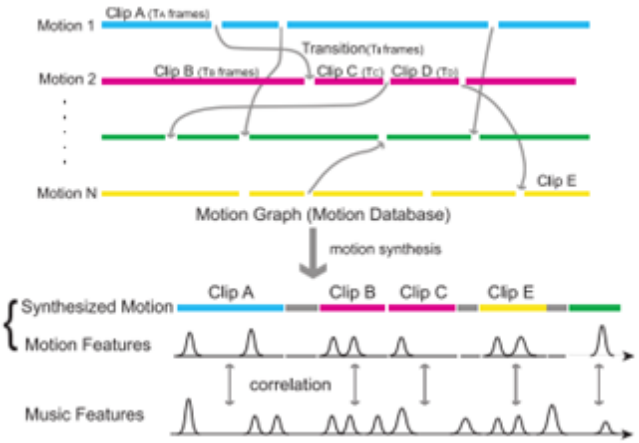


Figure 2.6: Motion Synthesis Algorithm [SNI06b]

imum angular velocity for all body portions is calculated for all motion data. After the construction of the motion graph and extraction of music features from input music data, the next step is to correlate the synthesized dance. For this purpose, a correlation between the music beat component and the motion key-frame component, and a correlation between music and motion intensity are evaluated (figure 2.6). The final step is to detect the best motion graph path. Here a search for the highest evaluation value is done and the resulting motion is produced.

In the end, a good match between motion rhythm and music beat was accomplished (see figure 2.7), and the resulting motion of this method is synchronized to both rhythm and music intensity.

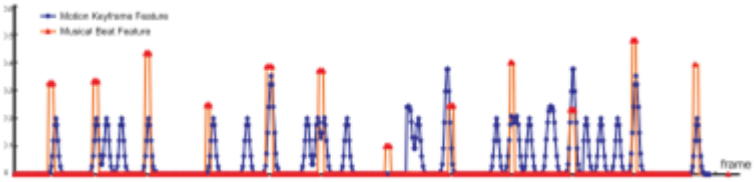


Figure 2.7: Motion-music correlation as proposed by [SNI06b]. The blue line represent motion key-frame feature and the red lines shows the music beat feature.

Kim et al. aimed to generate rhythmic motion on-the-fly [KPS03], so for the motion synthesis a library of movements' transition graphs is used. From this library, a movement transition graph is chosen according to the music, and from this graph a starting node is chosen, and until the music ends the movement transition graph is traversed from node to node, using transition probabilities to find the next node (figure 2.8). At each node a basic movement is synthesized.

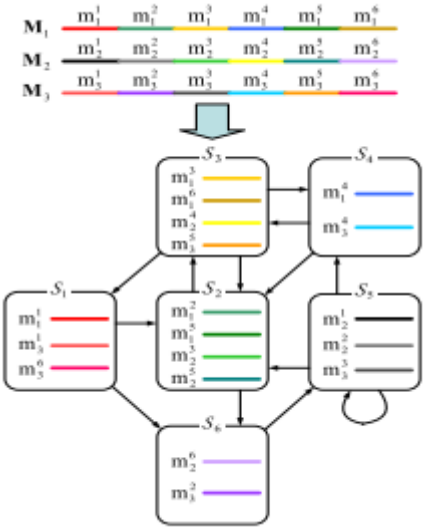


Figure 2.8: Examples of motions and transitions by the motion graph proposed by [KPS03].

The different size and proportions between the target character and the performer of the captured motion don't allow applying the motion directly to the target character. So a real-time motion retargeting algorithm is used to adapt the motion to the target character.

Oliveira et al., in [ONG+10], presented a method to generate key-poses from the already defined point cloud representation. For each of the determined metrical classes the method iteratively works on every kinematic chain, determining random coordinates to represent the position of each joint. The starting joints, or anchors, were determined as the mean values of their TGA spherical distributions. Then for every joint, the coordinates of the joint are determined by choosing a random point from the spherical cap, that results from the interception of the spherical distribution for the considered joint with a sphere centered in the position of the previous joint and with radius equal to the segment, or body part, length which links both joints (as presented in figure 2.9). As the position of the joints are randomly chosen, this method has the advantage of generating key-poses with variability, among successive metrical cycles of the same dance pattern.

2.2.2 Robotic Systems

Shiratori et al., in the final step of [SKNI07] generates temporally-scaled motion for a humanoid robot. Simple temporal scaling is done by adjusting the temporal frame of B-spline control points with the specified scaling ratio. To solve possible violations of angular limitations, like joint angular velocity, the joint limitations are considered and the upper body motion is modified (figure 2.10). The motion corresponding to the musical rhythm frames is segmented and the optimized so that the resulting joint angle satisfies

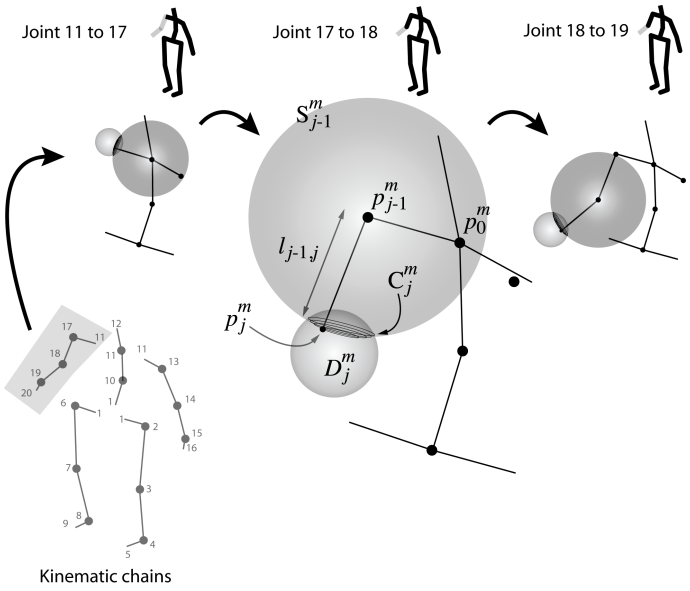


Figure 2.9: Example of the application of the key-pose synthesis method with variability, for one kinematic chain, starting from joint 11 until the extremity (joint 20) [ONG+10].

these conditions. The results of this method are good, as the robot can stably imitate the human dance motion, and it also preserves the details in the original motion. The main drawback of this method is the lack of the leg motion generation, which includes many balance and self-collision problems. Also the method risks making the robot fall because of rapid changes in acceleration.

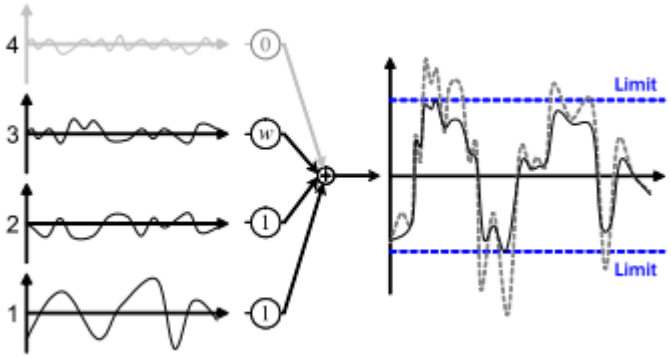


Figure 2.10: Consideration of joint limitations in the humanoid upper-body adaption of the original dance motion [SKNI07].

In [OSKI10], *Shiratori et al.* present a complement of [SKNI07], by combining the leg motion from [OSKI10] with the upper-body motion from [SKNI07]. To generate the trajectory of a swing foot in a set, interpolation is used based on a cubic polynomial that uses the starting point, the middle point and the landing point of the step. To maintain

balance the interval between steps is checked to keep a stable Zero Moment Point (ZMP). The leg motion is finally generate solving inverse kinematics and using refinement to keep the humanoid balance and perform collision avoidance.

Using both methods [SKNI07] and [OSKI10], *Shiratori et al.* generates full body motion. The result motion was synchronized to the music. Also the robot's biped balance was assured.

Zhao et al. uses kinematics mapping to translate the captured motion data to the humanoid morphology, while satisfying the humanoid constraints [ZHD+04]. The use of similarity functions helps the mapping of the motion, keeping the motion as close as the human but also ensuring that the physical constraints of the humanoid are not violated. This similarity function is maximal when the joint angles of the humanoid are equal to the ones of the human actor. For the upper limbs, the constraints include mainly the joint rotational range and the different number of DoFs. For the lower limbs, the ground contact conditions are critical. The algorithm for generate the movements starts by computing the robot waist position and then proceeds to the maintenance of the robot balance by recurring to 2 different algorithms, depending on the number of feet on the ground: an algorithm to the single-support phase and an algorithm for the double support phase, as described in figure 2.11.

Even by satisfying the kinematics constraints, the stability of the robot cannot be guaranteed, so the hip trajectory is modified to satisfy a constraint based on the ZMP criterion. To demonstrate the method the humanoid robot performed Chinese Kongfu "Taiji" (figure 2.12).

Kim et al. introduce a simplified human model to obtain a ZMP trajectory of a human based on the marker trajectories of motion capture data [KKYO09]. From the motion capture data some of the parameters are extracted to build a simplified human body with cylinders, spheres or boxes (figure 2.13). The body parts are connected as a chain, each having 3 DOF. In order to find the unknown parameters for building the body optimization is used, by minimizing the error between the ZMP trajectories obtained from the capture data, the reaction data, and the approximated ZMP trajectories of the simplified human model.

Kinematic and dynamic mapping is also used to generate the humanoid motion. In the kinematic mapping, the motion for the upper-body is created using Inverse Kinematics and optimization. The arms are scaled to resolve the geometric difference between the robot arms and the human arms. A cubic spline interpolation is used to smooth the discrete joint trajectory, providing position and velocity values of joints as references during real-time control. The mapping of lower-body deals mostly with the feet and pelvis motions. Another important step is the need to modify and scale the ZMP trajectories to avoid that the humanoid fall down. After this, the dynamic mapping is applied. The ZMP trajectory may still be inconsistent with the lower and upper-body motions, since no constraint on

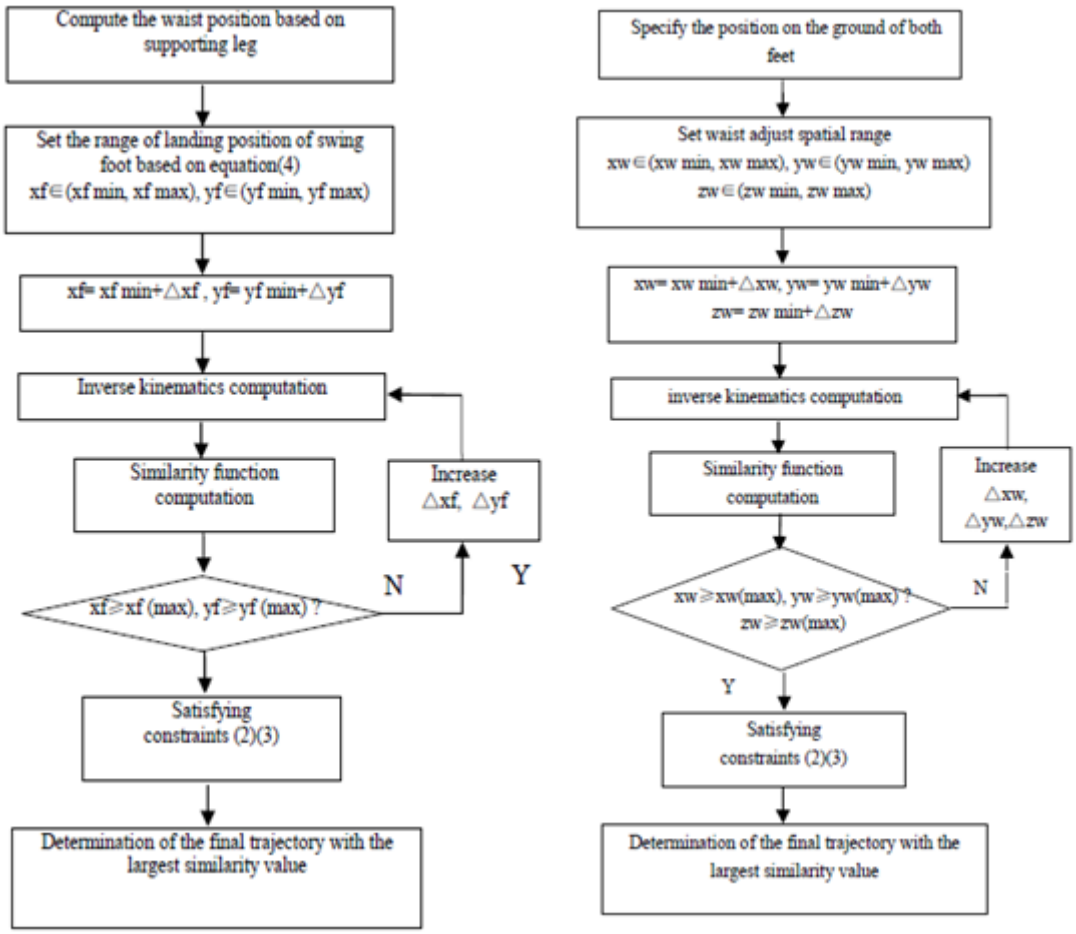
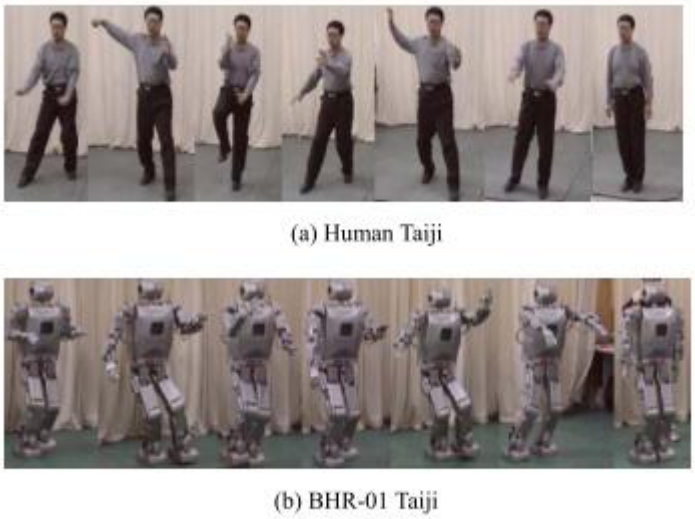


Figure 2.11: Motion Generation algorithms, considering the maintenance of similarity with the original dance, and biped balance [ZHD+04]: a) Single-support phase (*left*); b) Double support phase (*right*).

dynamics of the humanoid is considered. In the dynamic mapping the x-axis and y-axis values of the pelvis position are modified to satisfy the desired ZMP trajectories (figure 2.14).

The conversion of the motion from the human to the humanoid was properly done, and there wasn't a significant loss in motion similarity (figure 2.15).

Ruchanurucks et al. presented a method to generate motion based on trajectory optimization for a humanoid robot, with the application of physical constraints [RNKI05]. This optimization guarantees that the motion meets the physical constraints of the robot. Inverse kinematics is used to convert the markers position, from the motion capture data, to joint angles, that can be used by the robot. In order to assure that the calculated joint angles met the physical limits of the robot motors, optimization is applied by imposing objective functions according to the robot capabilities: angle, velocity and acceleration limits; dynamic force constraints and a composition of all. Motion refinement is also



(a) Human Taiji

(b) BHR-01 Taiji

Figure 2.12: Taiji key-poses: comparison of performer (a) and humanoid (b) [ZHD+04].

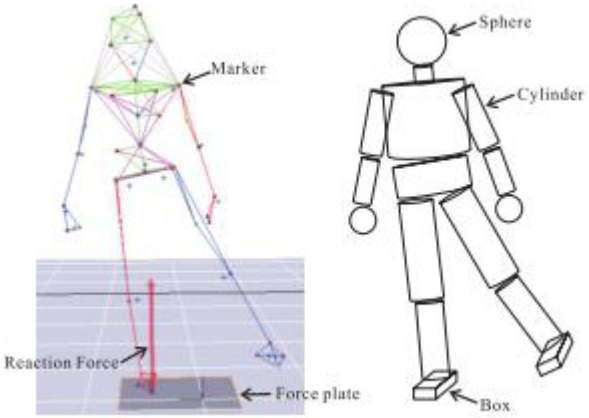


Figure 2.13: a) MoCap human body model (*left*); b) Simplified human body model (*right*) [KKYO09].

performed, in order to detect trajectory errors, and solve the errors found. The method preserves the detailed characteristics of the original motion, and at the same time ensures that physical limitations are met using several constraints (see figure 2.16 for a motion comparison).

Ruchanurucks et al., in a similar way to [RNKI05], proposes, in [RNKI06], a method to apply space-time constraints focusing on physical limits, without considering the balance of the robot's body. Constraints referring to joint rotational limitations are applied in first place, and then self-collision avoidance is checked. Velocity and dynamic force constraints are finally applied. This order is adopted because collision avoidance may pose discontinuity in trajectories, which can be solved by force and velocity-constraints.

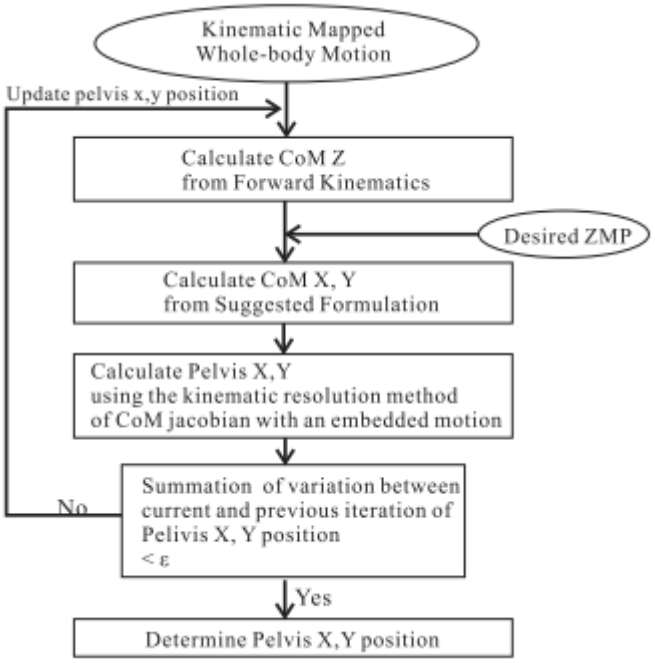


Figure 2.14: Algorithm for stable robot pelvis trajectories, in kinematic and dynamic mapping of human motion onto a humanoid robot [KKYO09].

Figure 2.15: MAHRU dance: a) Human performance (*top row*); Humanoid performance (*bottom row*) [KKYO09].

Collision could be solved automatically by increasing the critical distances for the periods that have collisions, by using an objective function.

Shiratori et al., in [Shi06], present a methods to extract the rotation quaternion from a spatial point representation of a body, and for the determination of Euler angles of each joint. The presented methods were only applied to the robot's upper-body and are based in Inverse Kinematics. This method uses the same body-centered local coordinate system presented in [SNI06a] and [SNI06b], and allows the calculation of the necessary arms joint angles for the humanoid HRP2 to replicate the original dance key-poses.

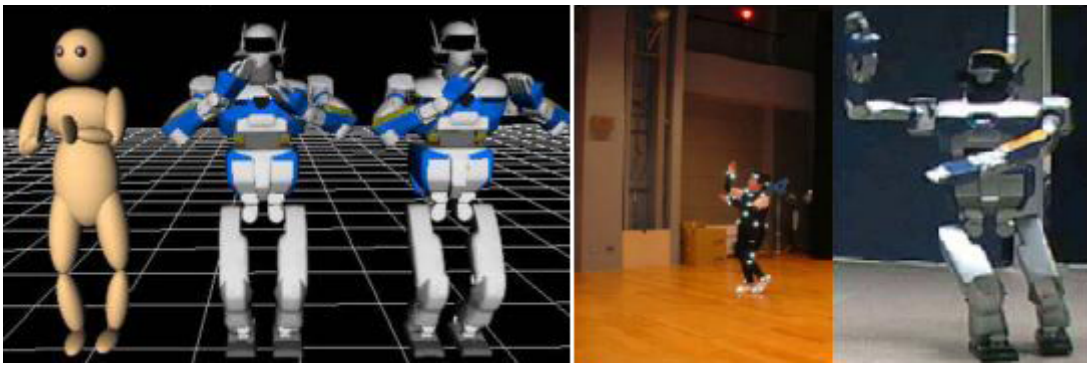


Figure 2.16: Humanoid dance motion generation [RNKI05]: a) Comparison of the hand markers between the MoCap body model and the robot (*left*); b) Exemplar Japanese dance performance (*left*).

2.3 Humanoid Motion Optimization

In the robotic motion generation there are some works that are also interesting to refer. Even if they didn't work with dance motion, they mostly use optimization for motion generation, and work closely with the humanoid NAO. The following referred works were accomplished within the FC Portugal 3D robotic football team [RL01] [PGLLPR09] [LR07].

Picado, in [Pic08], used HC and GA algorithms to generate and optimize football behaviors. In particular the algorithms were applied to the biped walking behavior. A four-phase walking behavior was optimized using HC and GA, and GA seem to have achieved better results but was slower. A walking style based on Partial Fourier Series (PFS) was generated using GA, and this style proved to be faster and more stable, being less sensitive to disturbances.

Rei, in [Rei10] [RRL11], created a skill optimization framework for the FCPortugal3D humanoid agent. The optimization framework can use different algorithms, namely HC, SA, TS and GA. The optimization is processed by changing the several parameters in the Slot Behaviors, as the joint angles values and the slot duration, and then evaluating the results of the new behavior, according to a pre-defined fitness function. This enabled the optimization of several and different types of behaviors with the definition of different evaluation functions and the appliance of the developed optimization algorithms. The optimization process was applied to different humanoid behaviors, and the algorithms with best results were HC and TS, being TS usually faster than HC at finding optimal solutions.

2.4 Conclusions and Proposal

With the analysis of the literature some lessons can be learned. In first place, we can see many limitations, or problems, with some of the presented approaches. The following table 2.1 presents an overview of some of the most important aspects of the presented works, concerning this thesis proposal.

Some of this methods lack in music perception [NOI+05], while others take it in consideration but only generate motion to animated platforms [OCFT+08], [NOI+05], [SNI06a], [SNI06b], [SKNI07], [KPS03].

There are few methods that labored the generation of full body humanoid motions. [SKNI07] and [KKYO09] only apply their methods to upper-body and [OSKI10] only generates legs motion. Although by combining some methods, [SKNI07] and [OSKI10], we could get full-body robotic motions, they have only been applied to a dance style and further experiments with different dance performances are needed.

On the other side, the literature also gives a good insight on the various problems and challenges that lay ahead, and that need to be addressed, and solved, to successfully create and present the proposed solution. These challenges are various: the analysis of the motion data must allow extracting meaningful information, but also must allow that the motion can keep some variability; the anatomic differences between the humanoid and the human actor need careful transformations, and the motion/music relationship must also be addressed. In the motion generation, the robot physical constraints, the balance of the humanoid, the necessity to avoid self-collisions, and the maintenance of the motion-music synchrony are the most important factors towards this goal.

For the development of further work, we shall base our motion analysis and motion generation on [ONG+10], taking advantage of the point cloud representation, and the parameterization and maneuverability that it offers and also the dance variability that this representation can translate. For extending the work to the robotics area, and to enable the reproduction of the generated key-poses, we shall focus on the appliance of the method presented in [Shi06]. With the extraction of the humanoid joint angles, Slot Behaviors can be created in SimSpark and the humanoid may reproduce the desired poses. Finally, for improving and refining the generated poses in the humanoid, in order to increase its similarity with the synthesized poses (resulting from the motion synthesis phase), and, consequently, with the real human dance performance, we shall apply optimization, recurring to the framework presented in [Rei10], by using the TS algorithm. Ultimately, the actual motion-music synchrony shall be assured by the adopted spatiotemporal representation of the original dance performance [ONG+10], and shall be replicated by interpolating the generated key-poses with the transition timings given by this representation.

Table 2.1: Overview related work, according to each identified problem to this thesis proposal (NA refers to Not Applied).

Approach	**Dance Motion Analysis**	**Motion/Music Synchrony**	**Motion Synthesis and Generation**	**Morphological Adaptation and Optimization**
Shiratori et al [SNI06a]	Music features (rhythm and intensity) correlated with motion features	Music features (rhythm and intensity) correlated with motion features	Motion Segment Interpolation	NA
Shiratori et al [SNI06b]	Motion features (key frame and intensity) extraction	Music features (music beats, degree of cord changes and intensity) correlated with motion features	Motion Graph + third order interpolation of body links	NA
Shiratori et al [SKNI07] [OSKI10]	Temporal scaling techniques + key-poses for maintaining the dance motion shape at different music speeds	NA	B-spline Interpolation [SKNI07] + Cubic Polynomial Interpolation and Inverse Kinematics [OSKI10]	Considered Joint constraints [SKNI07] + ZMP verification [OSKI10]
Kim et al [KPS03]	Motion Beats: Moments of rapid change in motion	Incremental time-wrapping	Motion Graph	Real time Morphological Adaptation
Ruchanurucks et al [RNKI05] [RNKI06]	NA	NA	Inverse Kinematics	Objective Functions [RNKI05] + Sequential Constraints [RNKI06]
Zhao et al [ZHD+04]	NA	NA	Kinematics Mapping	Similarity functions + Physical constraints
Nagata et al [NOI+05]	Extraction of shoulders and hips movements	NA	Motion Graph + joints linear interpolation by line-blending	NA

Chapter 3

Humanizing Robot Dance Movements

In this section the methodology used in order to archive the proposed goal will be described.

The implementation will be based on the method used in [ONG+10] to analyze the motion capture data. This analyze shall enable the extraction of the dance fundamental key-poses and their point cloud representation. With the information for each joint in each metric class, adaption will be done over the point cloud representation, so that it fits the target robot morphology. After this adaption, the key-poses will be randomly generated [ONG+11] (process illustrated in figure 3.1), and the angles between the body segments extracted. Finally the poses will be reproduced onto the humanoid robot.

Having robot performing the poses generated, optimization will be allied to increase the similarity of the motion generated to a robot model and the motion performed by the robot. Finally, the motion will be generated by interpolation of the key poses with the calculated interval between the poses.

3.1 Dance Motion Analysis

The dance motion analysis stage is based on the approach presented in [ONG+10]. 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 [NL10] for building a spatiotemporal representation of the original dance movement in relation to the respective music temporal structure (musical meter).

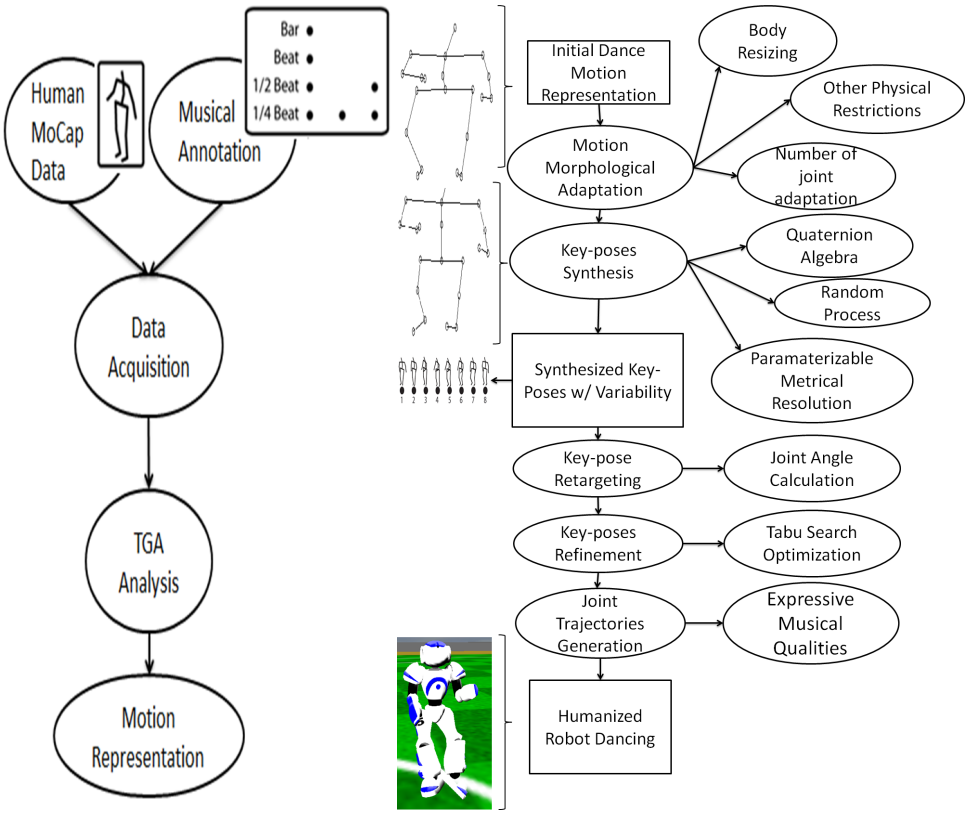


Figure 3.1: System Architecture. Motion Analysis Architecture [ONG+10] (*rigth*). Motion Generation: Adaptation and Retargting (*left*).

As illustrated in figure 3.2, this method relies in the projection of musical metric classes onto the motion joint trajectories, and generates point-clouds with the three-dimensional space occupied by each body joint according to every represented metric class, at different metrical (quarter-beat, half-beat, and beat) resolutions. These point-clouds are treated as topological spaces equipped with musical qualities which are further clustered and discriminated, by using Linear Discriminant Analysis (LDA) [NL10], into uniform spherical distributions, whose radii are defined by the mean of the Euclidean distances of all points to the centroid of the distribution.

Figure 3.3 illustrates the final spherical distributions for the hands of a samba dancer, at quarter-beat resolution.

In such way, this representation model offers a compact description for the possible positions of a joint in each metrical class. Besides, it supplies a parameterizable spatiotemporal description of the original dance, which translates both musical qualities and variability of the considered movement.

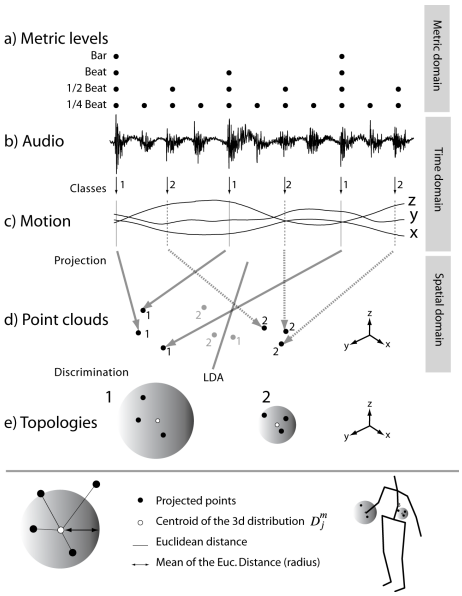


Figure 3.2: Projection of musical cues (metric classes) onto the dance trajectories. *a)* annotation of metric structure of the music is snchronized with the MoCap recording. These cues are projected *c)* onto the movement vectors (in the example, right hand movements) as different classes of points (e.g.: 1st beat, 2nd beat - receptively described as 1 and 2 in the figure). Finally, the point clouds are discriminated using LDA analysis which guarantees the separation of point-clouds. In this study we assumed a spherical distribution for the point clouds whose radius is defined by the average of the Euclidean distances from all points of the class to the mean [ONG+11].

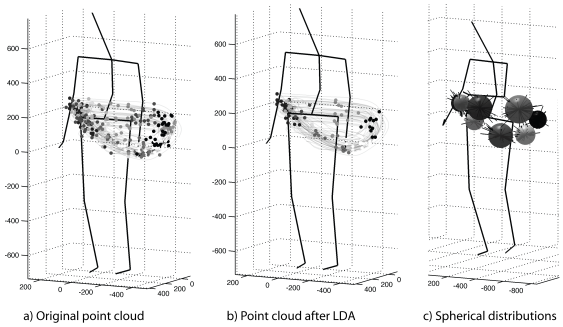


Figure 3.3: a) A point cloud representation of the left hand trajectories for the considered metrical classes, at quarter-beat resolution; b) Point cloud after applying LDA analysis; c) Spherical distributions (one per metrical class) representing the point clouds. [ONG+11]

3.2 Key-Poses Synthesis with Variability

By following [ONG+10], our method for synthesizing key-poses, while translating the variability of the original dance, consisted on calculating a set of full-body joint positions (one for each considered metrical class) according to the given TGA dance motion representation. [ONG+10] calculated the joint positions for every key-pose (metrical class)

by randomly choosing coordinates inside each joint's TGA distribution, without violating the fixed geometry of the human body.

Yet, this method presents some limitations when randomly determining, at every metrical class m, the spatial position of every joint, p_j^m. As illustrated in figure 3.4*c)*, this random joint position is randomly chosen inside the spherical cap C_j^m, which results from the interception of the TGA spherical distribution for the considered joint D_j^m, with a sphere S_{j-1}^m centered in the position of the previous joint, p_{j-1}^m, and with radius equal to the segment (or body part) length $l_{j-1,j}$:

$$\begin{cases} p_j^m = rand_p : p \in C_j^m \\ C_j^m = D_j^m \cap S_{j-1}^m \end{cases}, p, C_j^m, D_j^m, S_{j-1}^m \in \mathbb{R}^3. \tag{3.1}$$

rom this, for the current joint, a point from its spherical distribution is randomly chosen, considering that this point must be at the current distance from the last calculated joint position. Yet, in some situations, when there is no interception between D_j^m and S_{j-1}^m, C_j^m can be an empty set of points. Facing this problem the method "forces" (*i.e.* translates) the joint position p_j^m to the center of the spherical distribution of the current joint D_j^m. Since p_j^m is forced, there are no length restrictions applied to this generation, and the distance from p_j^m to p_{j-1}^m may not be equal to the length of the segment that connects this two body joints, $l_{j-1,j}$. This situation forces the body model to have different segment lengths than the desired, and so the fixed geometry of the human body model is compromised. Besides, this method only works with points, and the usage of a segment-relation-based description is more useful to our proposed robotic application.

In order to to improve [ONG+10], a method is proposed to generate the body joints positions based on the random rotation of their connected body segments [ONG+11], in relation to every previous segments (see figure 3.4*c)*). For this process, for each metric class, m, will be determinated the possible variation of the rotation quaternion defined between every two body segments. This rotation quaternion will define the 3d rotation of a target unity vector $\overrightarrow{v}'$ around its base vector $\overrightarrow{v}$.

In the beginning, the base vector for calculating the orientation of the anchor segment that connects the two anchor points of the used body model (joint 1 to joint 10 in figure 3.4 *a)*) is considered fixed in space at $\vec{v}_{s_0^m} = (0,-1,0)$. This anchor segment s_0^m is the base of all kinematics chains. After this starting point, each generated target vector is used as the base vector for the following segment, and the process is recursively applied until the extremity segment of each kinematic chain.

So, in every iteration, for each metrical class m, two segments (see figure 3.4*d)*) are then considered: the current segment s_j^m, that links the last calculated joint position p_{j-1}^m to the center of the spherical cap C_j^m, resulting from the interception of D_j^m and S_{j-1}^m; and

the previous segment s^m_{j-1}, that links p^m_{j-1} to its previous calculated joint position p^m_{j-2} of the same kinematic chain. And the possible variations of each segment rotation quaternion qv^m_j are calculated. To determine this variation, the extreme points of the spherical cap C^m_j are found as the maximum and minimum value that each coordinate can archive inside this cap, obtaining six extremity vectors $C^i_{ext\,j}{}^m$ (one maximum and one minimum for each spatial dimension d) that connect the previous joint position p^m_{j-1} to the extremes of C^m_j. The rotation quaternion to each of the six vectors $qv^i_{s^m_j}$ are then calculated using the former segment unity vector $\vec{v}'_{s^m_{j-1}}$ as the base vector, as follows:

$$\begin{cases} \vec{v}'^i_{s^m_j} = \vec{v}^i_{s^m_{j+1}} = C^i_{ext\,j}{}^m - p^m_{j-1} : \\ C^i_{ext\,j}{}^m = \min_d\left(C_{ext\,j}{}^m\right) \cup \max_d\left(C_{ext\,j}{}^m\right), d = \{x,y,z\} \\ qv^i_{s^m_j} = \cos\left(\alpha^i_{s^m_{j-1},s^m_j}/2\right) + \vec{u}^i_{s^m_j} * \sin\left(\alpha^i_{s^m_{j-1},s^m_j}/2\right) \\ \vec{u}^i_{s^m_j} = \vec{v}_{s^m_j} \times \vec{v}'^i_{s^m_j} \\ \alpha^i_{s^m_{j-1},s^m_j} = \arccos\left(\vec{v}_{s^m_j} \cdot \vec{v}'^i_{s^m_j}\right) \\ \vec{u}^i_{s^m_j}, \vec{v}_{s^m_j}, \vec{v}'^i_{s^m_j}, p^m_{j-1}, C^i_{ext\,j}{}^m \in \mathbb{R}^3; \quad qv^i_{s^m_j} \in \mathbb{R}^4 \end{cases}, \tag{3.2}$$

where $\vec{u}^i_{s^m_j}$ is the unity vector representing the 3d axis of rotation between both segments towards one of the extremities $C^i_{ext\,j}{}^m$, and $\alpha^i_{s^m_{j-1},s^m_j}$ the correspondent rotation angle.

The next step is to calculate the desired random quaternion $qv_{s^m_j}$ inside the spatial range of the previously calculated quaternion variations $qv^i_{s^m_j}$. Ultimately, using $qv_{s^m_j}$, the target direction vector $\vec{v}'_{s^m_j}$ is obtained:

$$\begin{cases} qv_{s^m_j} = \overline{qv}_{s^m_j} \pm \left|\overline{qv}_{s^m_j} - qv^i_{s^m_j}\right| * rand(0,1) \\ \vec{v}'_{s^m_j} = qv_{s^m_j} * \vec{v}_{s^m_j} * {q_{s^m_j}}^{-1} = \vec{v}_{s^m_{j+1}} \end{cases}, \tag{3.3}$$

where $\overline{qv}_{s^m_j}$ is the quaternion from the last calculated joint position to the center of the current spherical cap $\overline{C}^m_j$.

Finally, the current joint position p^m_j is calculated using the former joint position p^m_{j-1}, the obtained target direction vector $\vec{v}'_{s^m_j}$, and the current segment length $l_{j-1,j}$:

$$p^m_j = p^m_{j-1} + l_{j-1,j} * \vec{v}'_{s^m_j} : p^m_j \in T^m_j. \tag{3.4}$$

This process is iteratively repeated for each kinematic chain (see figure 3.4*d)*), starting in the trunk section (joint 1, joint 10, joint 11, joint 12), and then processing the arms and legs.

If the calculation of a chain is unable to determine the correct and valid interceptions for each, and every, joint of the kinematic chain, the calculation is repeated until a max-

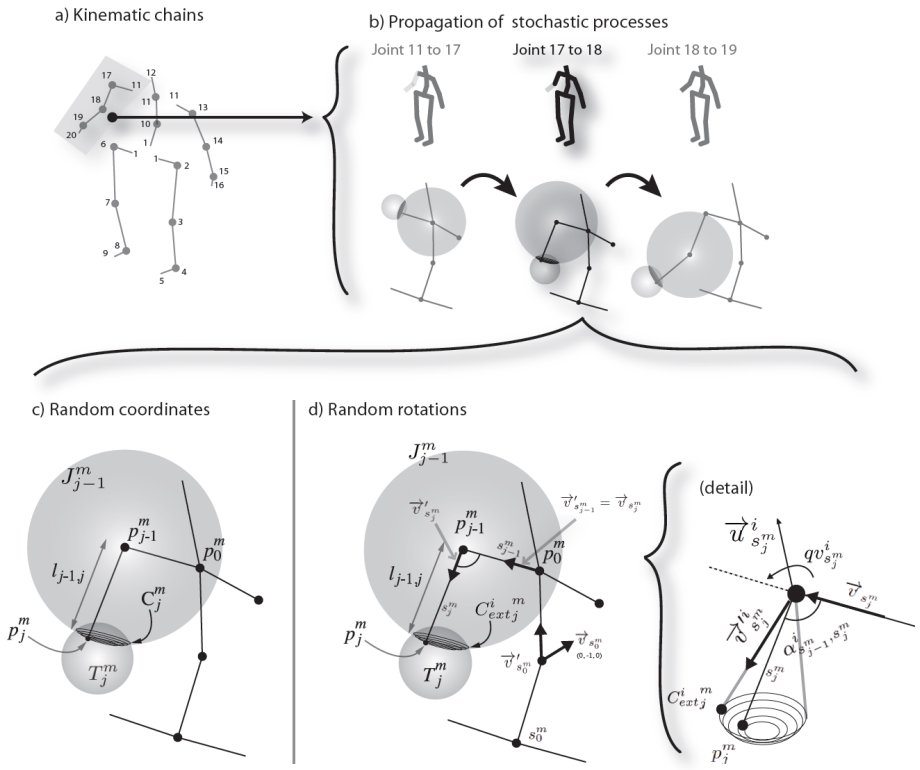


Figure 3.4: Body joints and kinematic chains (*a*)). Propagation of stochastic processes along a kinematic chain (*b*)). Example of the application of the method to calculate a random point, for the joint 18 (*c*)). Calculation of the joint 18 position based on random angles (*d*)). [ONG+11]

imum number of tries, that was fixed at 25. If, for any reason, no interception is found between D_j^m and S_{j-1}^m, making C_j^m an empty set, the current joint position is still calculated using equation 3.4, but $\vec{v}'_{s_j^m}$ is determined using the direction from the previous joint position p_{j-1}^m to the center of the current joint spherical distribution D_j^m. In this special case, the translation from the calculated joint position to its new spherical distribution's center is determined and applied to all the following spherical distributions' centers in the same kinematic chain. If reaching the maximum number of iterations without managing to build a "correct" kinematic chain, the chain with less associated error (*i.e.* less empty interceptions) will be used to represent the given key-pose's joint positions. This process assures that all the kinematic relationships, described in the TGA dance representation, are kept, inside the considered kinematic chain, without compromising the fixed geometry of the humanoid body.

3.3 Key-Poses Morphological Adaption

In order to get a representation of the key poses, in the target humanoid morphology, the original TGA-based dance representation must be adapted. This adaption must maintain

the spatial relationships between all TGA spherical distribution, for all the joints of each metrical class, without compromising the key-pose (and, consequently, the dance motion) shape, across all represented metrical classes. In order to archive this is important to look at the differences between the original and target morphologies, in terms of size, joints' degree of freedom, and other target kinematic physical constraints.

This adaption can be applied prior to the key-poses synthesis step (presented in section 3.2) or by adaptation of the already synthesized key-poses, in order to overcome the target humanoid constraints in the affected joints.

3.3.1 Different Segments 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, in order to allow the generation method to generate body segments in agreement with the target lengths [SOR11]. Considering 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, the center of the spherical distribution for the previous joint in the kinematic chain, to o_j^m is considered as $d_{j-1,j}^m$ and the direction vector from o_{j-1}^m to o_j^m is $\vec{vo}_{j-1,j}^m$. In order to change the segment length from $l_{j-1,j}$ to $l'_{j-1,j}$ is applied:

$$\begin{cases} redim = l'_{j-1,j}/l_{j-1,j} \\ d'^m_{j-1,j} = d^m_{j-1,j} * redim \\ r'^m_j = r^m_j * redim \\ o'^m_j = o^m_{j-1} + d'^m_{j-1,j} * \vec{vo}^m_{j-1,j} \end{cases} \tag{3.5}$$

where $d'^m_{j-1,j}$ 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 that has in o'^m_j its new center point, as ilustrated in figure 3.5.

Using equation (3.5), o'^m_j will be translated, this translation will change the relationship between this joint and the following joints of the kinematic chain. So the translation that from o_j^m to o'^m_j is calculated and then applied to all the following joints centers in the considered kinematic chain (exemplified in figure 3.5). The process of resizing is then applied iteratively to the remaining joints in the 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. 3.5. This relation is considered linear because if a segment has its size changed, its expected that the distance between the centers spherical distributions, for the initial joint and final joint of this segment, will have its size changed in the same proportion. In the same way, as the radius of a joint spherical distribution j is the reach of the segment from $j-1$ to j, if

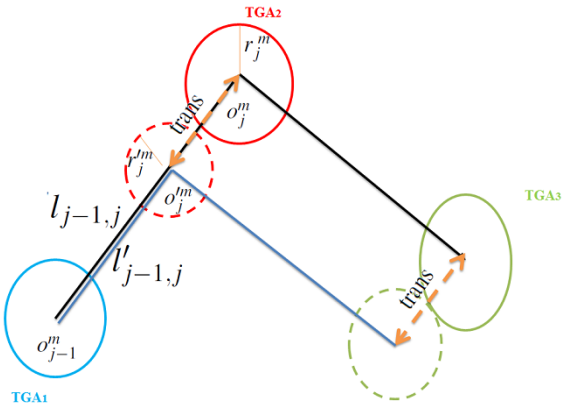


Figure 3.5: Resize method. Resizing of a segment (from TGA_1 to TGA_2) and translation of the segment target sphere (TGA_2). Example of the appliance of the translation, represented by "trans" in the image, to the rest of the kinematic chain (TGA_3).

the segment from $j-1$ to j has its size changed the radius of the spherical distribution of j will be changed in the same proportion. The method only performs a translation of the spherical distributions centers maintaining the spatial relation between them. The change in the spherical distribution radius, is regarded as an adaption of the segment reach.

3.3.2 Different Number of Joints

During the process of morphological adaptation the target body model may have a different number of joints than the original body model [SOR11]. In the described approach there was the need to "erase" a joint at the body model trunk, requiring the adaption of its original model, with three joints (joint 1, joint 10, joint 11 in figure 3.4*a)*) and two segments, to the target model with only two (extremity) joints linked by the whole trunk consisting of one segment. This operation could be done by simply connecting the position of the first trunk joint (identified in figure 3.6 as p_j^m) to the position of the last (identified in figure 3.6 as p_{j+2}^m), for every key-pose (metrical class m). Although this would be a simplistic solution it would introduce higher loss of information relative to every original pose, and so further edition of the affected joints would be necessary. For this purpose, for every key-pose, a method was implemented in order to find the best segment that stays inside the spherical distribution D_j^m of the first joint position p_j^m and the one D_{j+2}^m from the last joint p_{j+2}^m, but at the same time would be closest to the spherical distribution D_{j+1}^m of the middle joint position p_{j+1}^m.

As illustrated in figure 3.6, the first step in this process is to check if a sphere S_j^m, centered in the position of the first joint p_j^m, with radius $l_{j,j+2} = l_{j,j+1} + l_{j+1,j+2}$, intersects the spherical distribution of the extremity joint D_{j+2}^m.

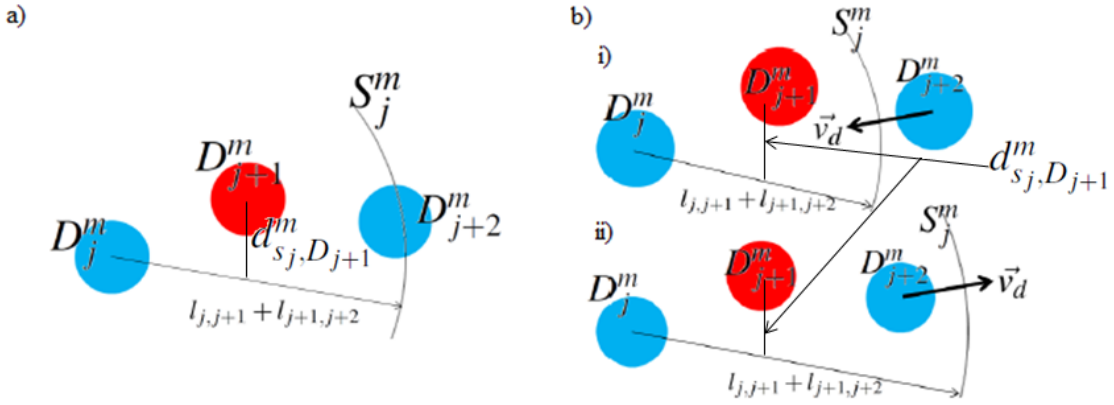


Figure 3.6: Morphological adaption of a kinematic chain with three joints (j, $j+1$, $j+2$) to one with two joints (j, $j+2$), by "erasing" the middle joint: a) A sphere S^m_j, centered in the position of the first joint p^m_j, with radius $l_{j,j+2} = l_{j,j+1} + l_{j+1,j+2}$, intersects the spherical distribution of the extremity joint D^m_{j+2}; b) S^m_j doesn't intersects D^m_{j+2} requiring a translation of D^m_{j+2} towards D^m_j, pointed by $\vec{v}_d$.

Following this step, two situations may occur: The intersection plane C^m_{j+2} between S^m_j and D^m_{j+2} may be an empty cap (exemplified in figure 3.6 *b)*) or not (exemplified in figure 3.6 *a)*). In the case that C^m_{j+2} is an empty set, the center of D^m_{j+2} must then be translated in the direction of the vector from the center of D^m_{j+2} to p^m_j (see 3.6 *b)i)*, or the opposite (see 3.6 *b)ii)*, respectively decreasing or increasing the distance from p^m_j to D^m_{j+2}. This is done until assuring an interception between S^m_j and D^m_{j+2}. After assuring the interception between D^m_{j+2} and S^m_j, the last step is to search for a point (the actual joint position p^m_{j+2}) in C^m_{j+2} that connected to p^m_j would be closer to the center o^m_{j+1} of the eliminated joint's spherical distribution D^m_{j+1}. This is done by selecting the p^m_{j+2} that would minimize the distance $d^m_{s_j,D_{j+1}}$ from the segment s^m_j, connecting p^m_j to the selected p^m_{j+2}, to the center of D^m_{j+1}.

There is a special case, though, where p^m_j is the anchor point of the model (and first point to be determined in the model), this allows to choose p^m_j, while keeping it inside D^m_j. So the direction of the vector from p^m_j to the center of D^m_{j+2} is considered, and p^m_j will be moved instead of the center of D^m_{j+2}. Enabling the interception, and then p^m_j will be moved, in order to approach him of to D^m_{j+1}, allowing a better fit of the segment that will be traced. Only this special case was applied to the spine in order to erase the middle joint. This is a best fit problem, where is tried to fit a line to 3 points, the spherical distributions centers, but the method presented differs from a normal best fit problem because the appliance of restrictions in terms of distance from the first and last point.

3.3.3 Different DOF

The proposed motion representation model and morphological adaption techniques can also support the simple restriction of any joint's DoFs. To apply this reduction of the degrees-of-freedom, at each metrical class m restrictions must be applied on the way as each joint position p_j^m is calculated. This reduction should be applied over the calculation of the interception between a sphere S_{j-1}^m, centered in the previous joint's calculated position p_{j-1}^m, and radius equal to the segment length $l_{j-1,j}$ linking p_{j-1}^m to p_j^m, and the current joint's spherical distribution D_j^m. This interception between the two spheres would give a 3-dimensional spherical cap that would permit the joint freedom across all the three spatial directions. Yet, the reduction of this freedom to 2 DoFs, and so only two directions (x,y), can be simply made by transforming the sphere S_{j-1}^m, centered in p_{j-1}^m and with radius $l_{j-1,j}$, into two circumferences, also centered in p_{j-1}^m and with radius $l_{j-1,j}$, but one described horizontally and the other vertically to the plane given by a segment linking p_{j-1}^m to the center o_j^m of D_j^m. This would give as result of the interception two arcs, that would only variate in two directions. Ultimately, for reducing the joint freedom to only 1 DoF we would simply chose one of the former arcs, according to the desired dimension (x or y).

Although this process would potentially solve the kinematic constraints imposed by the used body model in terms of degrees-of-freedom, while keeping a good similarity with the represented dance, this issue wasn't needed in the morphological adaptation to the desired target humanoid. Yet, the implementation of this method would be pursuit in the future, and further testing shall be presented.

3.3.4 Additional physical restrictions

In terms of physical restrictions that are only referent to the robot NAO were applied constraints over some segments in order to ensure that they are parallel [SOR11].

Looking at NAO morphology further adaption was needed, since both sides of the hip section can't move independently. This problem can be generalized as the need to ensure that certain body segments must be collinear (illustrated in figure 3.7).

So, in order to ensure that the segment s is collinear the segment $s2$, a random quaternion will be generated and the direction vector, $\vec{v}_{j,j'}^m$, for the segment s will be determined. Using this direction vector, a test will be made to generate a valid point inside the spherical distribution of $j2$ using the vector $\vec{v}_{j,j'}^m$:

$$p_{j2'}^m = p_{j2}^m + \vec{v}_{j,j'}^m * l_{j,j'} \tag{3.6}$$

$$(p_j^m + \vec{v}_{j,j'}^m * l_{j,j'}) \in D_{j'}^m \tag{3.7}$$

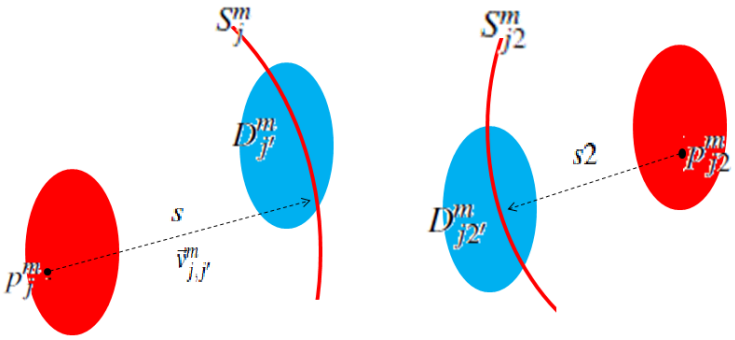


Figure 3.7: Physical Restriction. Situation where the segment (s) from p^m_j to a point in the spherical cap $C^m_{j'}$, that results from the interception of must be S^m_j with $D^m_{j'}$, must be parallel to the segment ($s2$) from $p^m_j 2$ to a point in the spherical cap $C^m_{j2'}$, that results from the interception of must be $S^m_j 2$ with $D^m_{j2'}$.

$$(p^m_j + \vec{v}^m_{j,j'} * l_{j,j'}) \in D^m_{j'} \wedge (p^m_{j2} + \vec{v}^m_{j,j'} * l_{j2,j2'}) \in D^m_{j2'} \tag{3.8}$$

If the newly generated point, $p^m_{j2'}$, is inside $D^m_{j2'}$ (respecting the condition 3.8) the problem is solved. If this new point doesn't respects the condition 3.8 the process will be repeated until reaching 25 attempts. Condition 3.7 is the constraint applied to accept a generated point in [ONG+10] and in the presented motion generation method based on random rotations, the algorithms tries to ensure this condition while not reaching the maximum number of attempts. For the special case of this physical restriction, the condition 3.8 must be ensured instead, to use symmetric, or equal, vectors in both segments.

If a point that satisfies condition 3.8 isn't generated and the algorithm reaches the maximum number of attempts, then only 3.7 will be used instead, and $p^m_{j2'}$ is calculated. With this $p^m_{j2'}$ will be forced, being outside the spherical distribution for this joint $D^m_{j2'}$, but the segments generated s and $s2$ will be parallel.

$$p^m_{j2'} = p^m_{j2} + \vec{v}^m_{j,j'} * l_{j2,j2'} \tag{3.9}$$

As this new point will be outside the spherical distribution of $j2'$, the translation from $p^m_{j2'}$ to the center of $D^m_{j2'}$ is determined. This translation is then applied to all the following spherical distribution center points in the kinematic chain. The translation will ensure that the remaining centers still maintain their spatial relation and until the extreme of the considered kinematic chain the pose is still similar to the original. The appliance of this method was done over the hip section, considering that j and $j2$ are the same joint (joint 1) and that j' and $j2'$ are the other hip joints (joint 2 and 6 respectively).

3.4 Key-Poses Retargeting

To generate the actual robot joint angles from the previously synthesized key-poses, a motion retargeting technique based on [Shi06] was applied for extracting the necessary robot joint angles based on the joint coordinates of each key-pose. This technique is based on an Euler rotational representation and allows to directly mapping the synthesized key-poses' joint positions into the respective robot's joint rotations, while trying to overcome the limited DoFs of the robot and the individual singularities of each of its joints [SOR11]. As such, this method calculates every joint rotations according to the defined local coordinate system, which is based on a body-centered axis.

Given the use of a different humanoid body model (robot NAO instead of [Shi06]'s HRP2) and the presence of singularities of different natures (especially between the robot's upper and lower body), we adjusted [Shi06]'s method to every specific joint of our model. Given the different natures of the upper and lower body of our robot, we also defined different local coordinate systems for each of these body parts. The described method below is referent to any given key-pose.

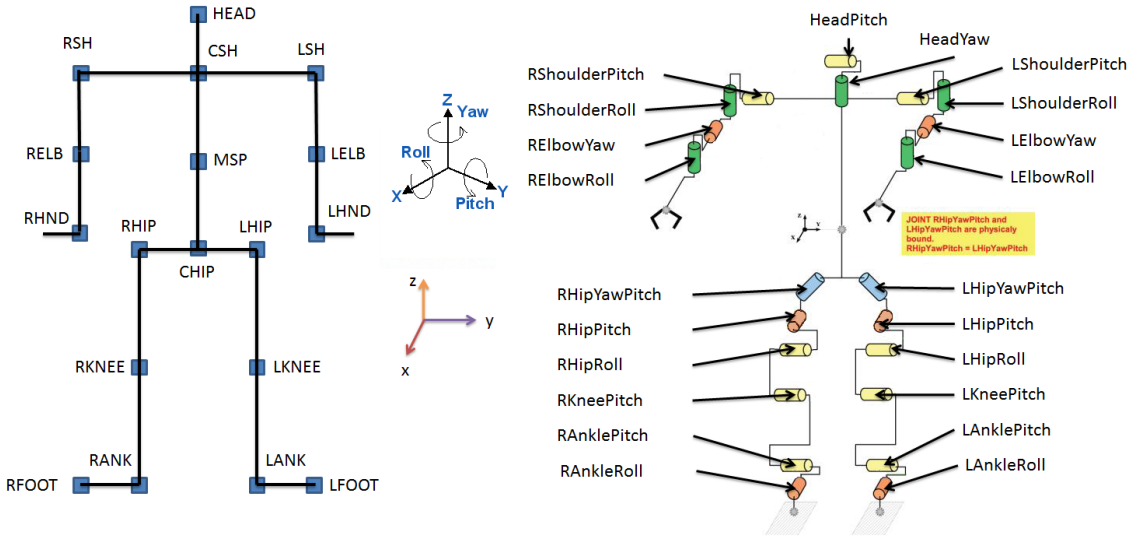


Figure 3.8: a) Considered labels for joints. b) Joint rotation axis and body local coordinate axis. c) NAO joint distribution and joint rotation axis (adapted from [Sim]).

3.4.1 Calculation of the Upper-Body Joint Rotations

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:

$$\begin{cases} R^{y}_{ub} = \vec{v}_{SH} = p_{LSH} - p_{RSH} \\ R^{z'}_{ub} = \vec{v}_{SP} = p_{CSH} - p_{CHIP} \\ R^{x}_{ub} = R^{y}_{ub} \times R^{z'}_{ub} \\ R^{z}_{ub} = R^{x}_{ub} \times R^{y}_{ub} \\ R_{ub} = [norm(R^{x}_{ub}), norm(R^{y}_{ub}), norm(R^{z}_{ub})] \end{cases}, \tag{3.10}$$

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

Now, for each vector in the global coordinate system, the correspondent vector in the local coordinate system is calculated and the angles in all the axis are determined. This calculation is specific to each segment of the considered humanoid morphology due to the presence of singularities of different natures. As R_{ub} is in fact a rotation matrix, the product of that rotation matrix and a global unit vector will result in the corresponding local unit vector.

By following figure 3.8a), the retargeting will start at the left shoulder. For such, we consider a vector $\vec{v}_{LSE}$ connecting the shoulder's joint global position p_{LSH} to the elbow's p_{LELB}, and proceed to a transformation of this vector to its 2 DoFs rotation, according to the defined local coordinate system R_{ub}. 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* (arm movement in the sagittal plane):

$$\begin{cases} \vec{v}_{LSE} = p_{LELB} - p_{LSH} \\ \vec{v'}_{LSE} = R^{T}_{ub} \times \vec{v}_{LSE} \\ LShoulderPitch = atan2(\vec{v'}_{z_{LSE}}, \vec{v'}_{x_{LSE}}) \end{cases}. \tag{3.11}$$

Then, the rotation of the left shoulder in the z axis (roll rotation *LShoulderRoll*, representing the arm movement in the coronal plane) is extracted in the same way by applying the previous calculated y rotation to the previous local vector $\vec{v'}_{LSE}$, as follows:

$$\begin{cases} \vec{v''}_{LSE} = R_{y}(LShoulderPitch) \times \vec{v'}_{LSE} \\ LShoulderRoll = atan2(\vec{v''}_{y_{LSE}}, \vec{v''}_{x_{LSE}}) \end{cases}, \tag{3.12}$$

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 $\vec{v}_{LEH}$ from the elbow joint $p_{L}ELB$, to the hand's $p_{L}HND$. Extracting the rotation in x axis and then robot's left shoulder *LShoulderRoll*.

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}:

$$\begin{cases} \vec{v}_{LEH} = p_{LHND} - p_{LELB} \\ \vec{v'}_{LEH} = R_z(LShoulderRoll) \times R_y(LShoulderPitch) \times R_{ub}^T \times \vec{v}_{LEH} \\ LElbowRoll = atan2(\vec{v'}_{z_{LEH}}, -\vec{v'}_{y_{LEH}}) \\ \vec{v''}_{LEH} = R_x(LElbowRoll) \times \vec{v'}_{LEH} \\ LElbowYaw = atan2(\vec{v''}_{y_{LEH}}, \vec{v''}_{x_{LEH}}) \end{cases} . \tag{3.13}$$

All the former process is similarly applied to the right arm of the robot.

For the head section, the same local system from the arms R_{ub} was used (see eq. (3.10)). Identically, the two existing head joint rotations (2 DoFs), $HeadYaw$ in the z local coordinate axis (movement of the head in the transverse plane) and $HeadPitch$ over the y axis (movement over the sagittal plane), were calculated as follows:

$$\begin{cases} \vec{v}_{CSH} = p_{HEAD} - p_{CSH} \\ \vec{v'}_{CSH} = R_{ub}^T \times \vec{v}_{CSH} \\ HeadYaw = atan2(\vec{v'}_{y_{CSH}}, -\vec{v'}_{x_{CSH}}) \\ \vec{v''}_{CSH} = R_z(HeadYaw) \times \vec{v'}_{CSH} \\ HeadPitch = atan2(\vec{v''}_{z_{CSH}}, \vec{v''}_{x_{CSH}}) \end{cases} . \tag{3.14}$$

3.4.2 Calculation of the Lower-Body Joint Rotations

For the extraction of the legs' joint rotations a new local coordinate system R_{lb} is defined. This process start by the hip joint rotations, proceeding to the knees, and finally to the feet joints. This new coordinate system R_{lb} use both the hip and the spine directions, as follows:

$$\begin{cases} R_{lb}^y = \vec{v}_{HIP} = p_{LHIP} - p_{RHIP} \\ R_{lb}^{z'} = \vec{v}_{SP} = p_{CSH} - p_{CHIP} \\ R_{lb}^x = R_{lb}^y \times R_{lb}^{z'} \\ R_{lb}^z = R_{lb}^x \times R_{lb}^y \\ R_{lb} = [norm(R_{lb}^x), norm(R_{lb}^y), norm(R_{lb}^z)] \end{cases} . \tag{3.15}$$

The following steps describe the calculation of all joints presented in the robot's left leg. Again, the same process was identically defined for the right leg.

Starting at the hip joints, firstly we extracted the robot's left hip roll rotation $LHipPitch$, that controls the hip movement along the body's sagittal plane:

$$\begin{cases} \vec{v}_{LKH} = p_{LHIP} - p_{LKNEE} \\ \vec{v'}_{LKH} = R_{lb}^T \times \vec{v}_{LKH} \\ LHipPitch = atan2(\vec{v'}_{x_{LKH}}, -\vec{v'}_{z_{LKH}}) \end{cases} . \tag{3.16}$$

Then the *LHipRoll*, representing the left leg movement over the coronal plane, is extracted:

$$\begin{cases} \vec{v''}_{LKH} = R_y(LHipPitch) \times \vec{v'}_{LKH} \\ LHipRoll = atan2(\vec{v''}_{x_{LKH}}, -\vec{v''}_{z_{LKH}}) \end{cases} . \tag{3.17}$$

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 over the transverse plane, symmetrically, for both legs:

$$\begin{cases} \vec{v}_{LAK} = p_{LKNEE} - p_{LANK} \\ \vec{v'}_{LAK} = R_x(LHipRoll) \times R_y(LHipPitch) \times R_{lb}^T \times \vec{v}_{LAK} \\ LHipYawPitch = atan2(\vec{v'}_{y_{LAK}}, -\vec{v'}_{x_{LAK}}) \end{cases} . \tag{3.18}$$

After calculating all the three rotations for the hip section we proceed to the calculation of the left knee and ankle rotations. For both knee and ankle joints only 1 functional DoF is provided by the humanoid NAO, which only enables both joints rotations over the y axis (sagittal plane) of the local coordinate system. The robot body model also presents another rotation freedom at the ankles concerning its roll rotation around the z (coronal) plane, *RAnkleRoll* and *LAnkleRoll*. Yet, this rotation was not considered since it doesn't have any correspondence to our synthetic body model. It would be rather important for the maintenance of the robot's biped balance, which is outside the scope of this thesis.

As such, ultimately, the respective pitch rotations of the left knee *LKneePitch* and left ankle *LAnklePitch* were calculated as follows:

$$\begin{cases} \vec{v''}_{LAK} = R_z(LHipYawPitch) \times \vec{v'}_{LAK} \\ LKneePitch = atan2(\vec{v''}_{x_{LAK}}, -\vec{v''}_{z_{LAK}}) \end{cases} , \tag{3.19}$$

$$\begin{cases} \vec{v}_{LFA} = p_{LANK} - p_{LFOOT} \\ \vec{v'}_{LFA} = R_y(LKneePitch) \times R_z(LHipYawPitch) \times R_x(LHipRoll) \\ \qquad\qquad \times R_y(LHipPitch) \times R_{lb}^T \times \vec{v}_{LFA} \\ LAnklePitch = atan2(\vec{v'}_{z_{LFA}}, -\vec{v'}_{x_{LFA}}) \end{cases} . \tag{3.20}$$

3.5 Key-Poses Refinement

As a final step, to improve the degree of similarity of the generated robot dance with the original, all key-poses were additionally refined by means of optimization.

Previous to the actual key-pose refinement, angle restriction was done. Angles constraints were applied to ensure that the angles passed to the robot were inside the limits of the joints. To that all the angles over the maximum limit were considered equal to the limit, and all the angles below the minimum limit were considered equal to the minimum

limit. This was done to facilitate the optimization, because if the angle is outside the range of the joint, changes in that angle during optimization may not reproduce change in the poses and consequently in the evaluation function.

For this, the framework presented in [Rei10], and [RRL11] was used, allowing the simple optimization of the NAO behaviors using several optimization processes. From the several optimization algorithms available in this framework, the one that presented best and faster results was the Tabu Search algorithm, so it was the optimization algorithm applied in the key-poses refinement. For such refinement, we defined a fitness function that evaluates the similarity between the generated robot key-poses and the previous synthesized-adjusted ones. This similarity function considers the distance between every pair i, j of existing body points (joints location), and measures the difference between such distances in the synthesized-adjusted $d_{pose_{i,j}}$ and robot $d_{robot_{i,j}}$ key-poses, as such:

$$Sim_{pose,robot} = \sum_{j}^{n} \sum_{i=j}^{n} \left| d_{pose_{i,j}} - d_{robot_{i,j}} \right| \tag{3.21}$$

As described in eq. (3.21), the sum of all differences between the distances of every i, j points both in the synthesized-adjusted and robot key-poses, can then be seen as the deviation of a generated robot pose in comparison to the original one. Therefore, the greater the value of this measure the worse is the similarity of the evaluated robot pose with its original.

The application of this similarity function was done over some points in the synthetic-adjusted body model, by trying to match both these points and the robot's equivalents. Following figure 3.4 *a)*) and figure 3.8*a)*, the given matching was applied and the following joints were considered in this similarity evaluation: n the torso the joints *MSP*, for the neck the joint *CSH* and for the head the joint *HEAD* were used; for the arms points for the shoulders, joints *RSH* and *LSH*, for the elbows, joints *RELB* and *LELB*, and for the wrists, joints *RHND* and *LHND*; for the legs the hips, joints *RHIP* and *LHIP*, the knees, joints *RKNEE* and *LKNEE*, the ankles, joints *RANK* and *LANK*, and the foot, joints $RFOOT$ and $LFOOT$, were considered. With this the considered joint positions for the generated key-poses was compared with the actual 3D position of the simulated humanoid. The usage of the distance in the evaluation function allows to have different axis systems, just needing to keep the same scale between the compared distances.

3.6 Robot Motion Generation

After having all the necessary key-poses adapted and optimized, the actual robot dance motion is finally generated by ordering all these key-poses according to the respective dance representation model (at the considered metrical resolution), and concatenating

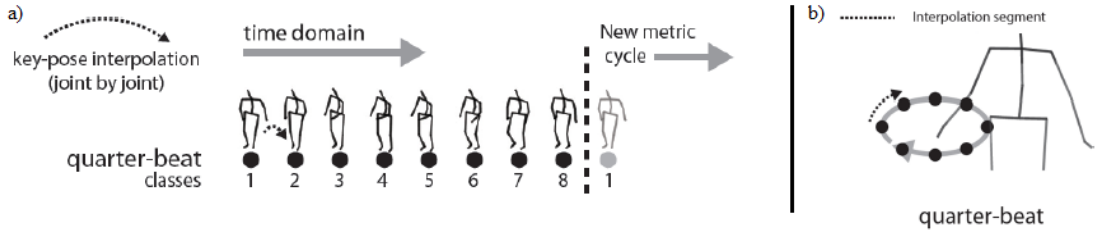


Figure 3.9: Key-poses Interpolation (adapted from [ONG+11]. a) Key-poses interpolation. b) Joint interpolation.

them, by interpolating all of their joint angles into continuous trajectories (figure 3.9b)). For generating the joint trajectories between every two key-poses, we used a sine function to interpolate the joint angles θ_i of the first pose towards the joint angles θ_f of the second, and performed this transition within an imposed amount of time (time domain presented in figure 3.9a)), given by δ.

As such, every of the robot's joints follow a sine-like trajectory between every two key-poses, from θ_i to θ_f:

$$f(t) = A \times sin\left(\frac{\phi_f - \phi_i}{\delta} \times t + \phi_t\right) + \alpha, \forall t \in [0, \delta] \tag{3.22}$$

Where A is the amplitude (eq. (3.23)), and α is the defined offset (eq. (3.24)):

$$A = \frac{\theta_f - \theta_i}{sin(\phi_f) - sin(\phi_i)} \tag{3.23}$$

$$\alpha = \phi_i - A \times sin(\phi_i), \tag{3.24}$$

where θ_i and $\theta_f \in [-\pi, \pi]$, δ is the duration of the key-poses' transition in seconds, and ϕ_f is the initial key-pose joints' phase and ϕ_i the finals'.

3.6.1 Generating Expressive Robot Dancing Sequences

Since the expressive aspects of dancing motion are implicitly related to its musical qualities (beat-synchrony, and others) and its underlying variability, constrained by the dance pattern *per se*, we tried to replicate both these aspects into the generated robot dancing sequences. In respect to beat-synchrony, this was achieved by determining the dance motion frames where the determined key-poses occur, according their specific metrical

classes (given by the former spatiotemporal dance representation), and so calculating for each pose p its correspondent duration of transition to the next pose $p+1$:

$$\delta_p = \frac{(frame_{p+1} - frame_p)}{freq}, \tag{3.25}$$

where $freq$ is the frame rate (100 frames per second). This calculation is done to find the correct time between the poses, and the time between the key-poses is used as the duration of the transition between two poses.

As the process of synthesizing key-poses uses the random selection of rotations, and the motion is done by the cyclic repetition of this poses (figure 3.9a)), we will obtain variability in the final generated dance motion. The final dance sequence is then built by concatenating successive cycles of key-poses, synthesized with variability at quarter-beat resolution, according to the dance TGA representation.

3.7 Conclusions

In this chapter were presented the methods necessary for analyzing, adapting and generation of the dance motion fundamental key-poses. The methods for generation and size adaptation presented are flexible and simple to use, enabling the easy generation of key-poses with the desired body model. Further morphology adaptation was necessary, and the methods to do so were presented. The extracted and adapted key-poses act as a intermediate representation, from which angles were extracted in order to enable the humanoid robot to reproduce the same key-poses. With this extraction, XML Slot Behaviors were built in SimSpark simulator, and as this extraction may not be exact, refinement of the key-poses was applied by means of Tabu Search optimization. The optimization of the key-poses helped enhancing the similarity of the humanoid key-poses to the original ones. After this step, the humanoid dance motion is generated by sine-interpolation, ordered according to the used dance representation model. Further testing of this methods should be done concerning mainly the similarity of the key-poses and the similarity of the final humanoid dance motion with the original human dance. Specifically, it's also important to understand the impact that the morphological adaptation has in the dance in terms of similarity. In every case, it must be ensured that all the applied methods don't produce any unwanted alteration in the body model. Ultimately, the effect of introducing musical synchrony and variability in order to generate expressive robot dance performances must be also evaluated.

Chapter 4

Experiments and Results

This section presents the performed experiments and the respective obtained results. The methods presented in Chapter 3 were tested in terms of pose similarity with the original captured dance, and overall musical expressiveness of the final generated robot dance. The presented tests and results aim to validate all the proposed methods, by evaluating and presenting visual insights of the several performed adaptations and motion retargeting towards the resultant robot dance, while overcoming all the morphological differences and kinematic constraints imposed by the used humanoid body model in comparison to the original human-based captured.

In terms of overall similarity with the original dance, we performed individual tests over the produced key-poses by each stage of the whole process (synthesis, adaptation, retargeting, and refinement), always comparing these with the ones generated by the just previous stage, and with the original captured ones. This evaluation, done by means of numerical measurements or visual comparison, of the generated key-poses at each process stage, allows measuring the individual effect of each method in terms of gains and losses of similarity in comparison to the original dance, that we aim to replicate.

As a complement, all methods' results were individually evaluated in terms of morphological consistency to check if they were not compromising the fixed geometry of the humanoid body model, in the process. Again, this is performed by recurring to specific numerical measurements and visualization of physical details, that aim to confirm the correct disposition of all joints and segment lengths according to the target humanoid body model.

Finally, in order to evaluate the degree of musical expressiveness of the generated robot dance, which includes the effect of introducing variability and the degree of beat-synchrony, and given the higher subjectiveness of these aspects, we additionally performed an on-line user-evaluation with some video trials followed by proper questions

into a Likert-scale questionnaire [Mog99]. In order to corroborate the former quantitative results over the presented degree of similarity with the original dance, the subjects were also inquired about similarity aspects, concerning specific stages of our method.

By following [ONG+10], all tests were performed with the same MoCap data of samba dance. Considering the results reported in [ONG+10], all the performed tests described in this Chapter (except the first, reported in Section 4.1), were generated with a TGA dance representation defined at a quarter-beat resolution, introducing variability in the key-poses' synthesis process (i.e. "variability-4" parametrization), since this parametrization seems to greatly represent the original samba dance.

4.1 Key-Poses synthesis based on random angles

In order to evaluate the proposed method, for synthesizing key-poses based on the random choice of every body segment's rotations, at different metrical resolutions, and in order to be comparable with the [ONG+10]'s method (which distinctively selects random coordinates for every joint positions), we also interpolated the resultant key-poses by means of a cubic spline interpolant. Identically, we also measured the similarity between the synthesized and original dance sequences, and additionally measured their similarity against the sequences synthesized with [ONG+10]'s methods, by recurring to the correlation coefficient [DVMC10], which quantifies the linear time-domain correlation between two signals s_1 and s_2. Considering these signals as the joint motion trajectories of the compared pair of sequences, with N frames each, the correlation coefficient $r_{s1,s2}$ was defined as follows:

$$r^{i}_{s_1,s_2} = \frac{\sum_{n=1,j=1,d=1}^{N,J,D} \left[\left(s_1(n,j,d) - \bar{s_1}\right)\left(s_2(n,j,d) - \bar{s_2}\right)\right]}{\sqrt{\sum_{n=1,j=1,d=1}^{N,J,D} \left(s_1(n,j,d) - \bar{s_1}\right)^2 \sum_{n=1,j=1,d=1}^{N,J,D} \left(s_2(n,j,d) - \bar{s_2}\right)^2}}, \tag{4.1}$$

where J is the total number of joints (total of 20 joints in the considered MoCap/synthetic body model), D is the considered spatial dimensions (3 dimensions), $\bar{s_1}$ and $\bar{s_2}$ are the mean frames across all joints and dimensions for s_1 and s_2, and i is the index of the evaluated sequence. The maximum value for the correlation between $s1$ and $s2$ is $r_{s_1,s_2} = 1$, occurring when the signals are identical.

By also following [ONG+10], and in order to evaluate the reliability of the proposed method against the same conditions, we also synthesized 6 dance sequences, with $30s$ (3000 frames at $100Hz$) each, using different parameterizations on the used TGA dance motion representation, concerning different metrical resolutions and variability conditions. Besides, we also measured the dimensionality ($Dim(J*S*T)$) of the synthesized sequences, which endeavors to measure the number of spatiotemporal arguments used to describe the full-body 3D trajectories of the whole sequence (where J is the number of

joints of the body model, S is the number of spatial arguments used in the TGA spherical representation, and T is the number of metrical classes used to represent the dance sequence, at the considered metrical resolution); and the level of reduction (*Reduction*) introduced by the use of such representation model (with different parameters) in comparison to using the whole joint trajectories information supplied by the respective MoCap data. This method, and all these tests and evaluation were already described in a journal paper [ONG+11] submitted to the EURASIP Journal on Audio, Speech, and Music Processing - Special Issue on Music Content Processing by and for Robots.

The correlation results are then presented in table 4.1, for the same parameterizations evaluated in [ONG+10], where r_{o_1,o_2} is the correlation coefficient between two excerpts of the same captured original sequence, off-set by one metrical cycle (one measure); $r_{c,o}$ is the correlation coefficient between the synthesized sequence based on key-poses generated using random joint coordinates and the original sequence [ONG+10]; $r_{r,o}$ is the correlation coefficient between the synthesized sequences using key-poses synthesized based on random joint rotations and the original sequence; and $r_{c,r}$ is the correlation coefficient between the same dance motion sequence synthesized using key-poses generated using random joint coordinates and key-poses generated using random joint rotations.

Table 4.1: Comparison of the Correlation coefficients between the different key-pose synthesis methods, in relation to the spatiotemporal dimensionality (*Dim*) and level of reduction (*Reduction*) of their respective representation models. [ONG+11]

Test	$\mathbf{r_{o_1,o_2}}$	$\mathbf{r_{c,o}}$	$\mathbf{r_{r,o}}$	$\mathbf{r_{c,r}}$	**Dim**(JxSxT)	**Reduction**
original	0.818	–	–	–	20x3xnFrames	0
random-2	–	0.420	–	–	20x(3+1)x2 = 160	0.38xnFrames
variability-1	–	0.516	0.540	0.890	20x(3+1)x2 = 160	0.38xnFrames
variability-2	–	0.774	0.782	0.923	20x(3+1)x4 = 320	0.19xnFrames
variability-4	–	0.809	0.802	0.965	20x(3+1)x8 = 640	0.09xnFrames
fixed-4	–	0.822	0.813	0.993	20x3x8 = 480	0.13xnFrames

Although such correlation results allowed to compare the different synthesized dance sequences against each other, and finally against the original captured dance, it don't validates the requirement of implicitly keeping the geometry of the body model fixed in the process of synthesis. In order to evaluate this aspect, on both methods of synthesis, we compared the morphology of the synthesized body models against the original one. As such, we compared both original and synthesized body sizes, by comparing the sum of all their respective body segment lengths, calculated as follows:

$$Size_{total} = \sum_{s=1}^{19} l_s, \tag{4.2}$$

where l_s is the length in *mm* of the segment s. The body segments considered unite the joints presented in figure 4.1 below.

This comparison, for both key-pose synthesis methods, is presented in table 4.2, the table presents the mean body size for the generated poses and the mean deviation. The value of the obtained total size is a mean of the obtained value of eq. (4.2) for all the eight key-poses.

Table 4.2: Comparison of the total body size expected with the obtained (values in *mm*).

Generation Method	Expected Size	Obtained Size Mean	Obtained Size Mean Deviation
Random Points	4697.777	4695.422	19.287
Random Rotations	4697.777	4697.777	0

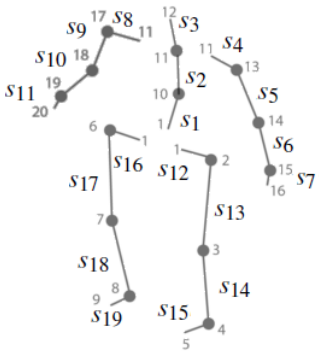


Figure 4.1: Body model segments (represented by $s_{segmentnumber}$), joints and kinematic chains.

After accomplishing the size validation, it's important to visually and numerically test if all the calculated joint positions are inside their respective joint spherical distributions, following the given TGA representation model. Figure 4.2 presents a visual plotting of exemplar key-poses synthesized at "variability-4" with random rotations, and the respective joint spherical distribution. It's important to refer that some joints, at some metrical classes, don't present spherical distribution because their radius were reduced to zero during the motion analysis phase with the application of the LDA. This situations are pointed with small sphere with a cross inside in figure 4.2 (left shoulder of both poses on the left side).

On the other hand, figure 4.3 presents the same synthesized key-poses superimposing both random method's results against the same key-poses synthesized without variability (at "fixed-4"), allowing a better comparison of the key-poses generated from each method.

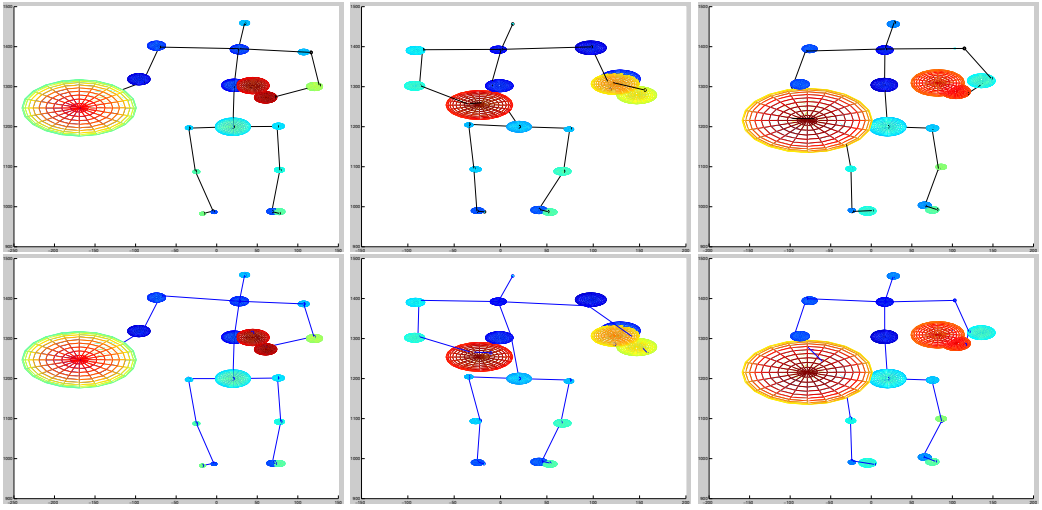


Figure 4.2: Visualization of three synthesized key-poses, at "variability-4", in order to evaluate if all the calculated joints are inside of their respective TGA spherical distributions. a) Method based on random points (*bottom row*). b) Method based on random rotations (*upper row*).

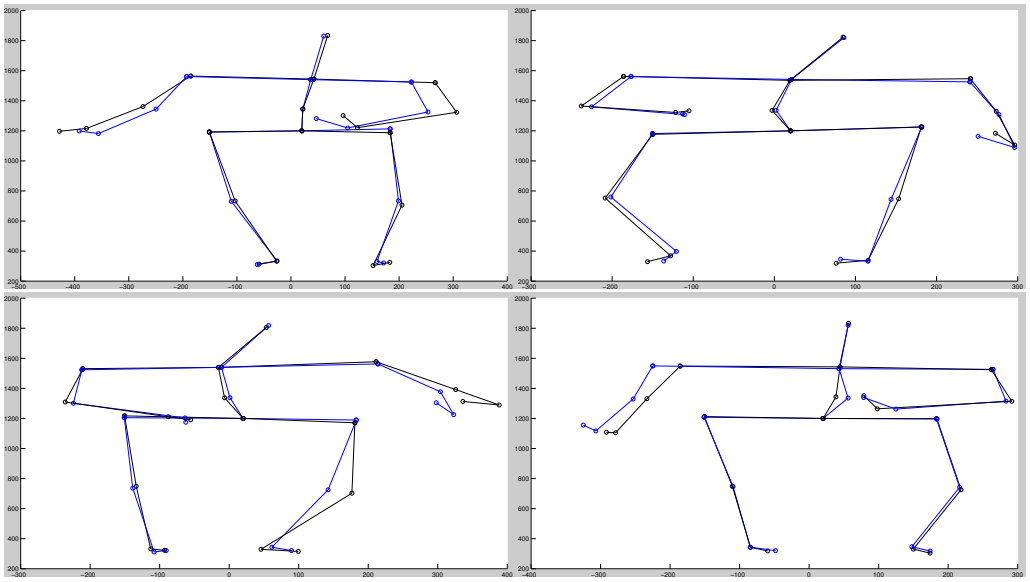


Figure 4.3: Visual comparison of some key-poses examples synthesized, at "variability-4", with: method based on random points (*black line*), method based on random rotations (*blue line*). Joints are represented by a circle.

4.2 Motion Morphological Adaption

4.2.1 Different Segments Lengths

In order to assure the correct adaption of the body, a size comparison using eq. (4.2) was applied. The results of this measurement, using body models with either half (0.5x) and double (2x) sizes (by multiplying all segment lengths by these same factors) are presented in table 4.3. As a complement, the visual comparison of the different generated bodies

was also presented in figure 4.4.

Table 4.3: Comparison of the total body sizes for different scaling factors and NAO's morphology.

Scale	Expected Size	Obtained Size
1x	4697.777	4697.777
2x	$4697.777 \times 2 = 9395.554$	9395.554
0.5x	$4697.777 \times 0.5 = 2348.889$	2348.889
NAO	1573.00	1573.00

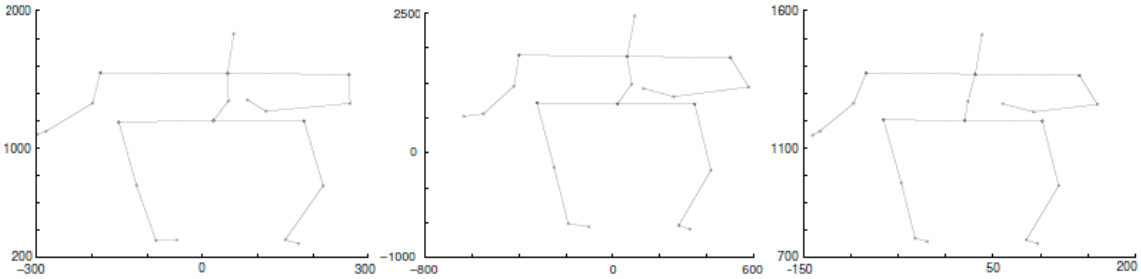


Figure 4.4: Key-poses synthesized with variability (at "variability-4") for different body scales (axis measures in *mm*): a) Original captured human body (*left*). Body segments scaled 2x (*middle*). Body segments scaled 0.5x (*right*).

In order to replicate the robot NAO's humanoid morphology, a final adaption test was made by resizing all original segment lengths to the NAO's. Both expected and obtained body sizes (by applying the proposed resizing method) are also presented in the last line of table 4.3. The respective NAO's segment lengths were extracted from the NAO box model [Sim], as illustrated in figure 4.6.

4.2.2 Different Number of Joints

The application of the method presented that allows to erase a joint was only tested for one numerical parameter, the total body size. Since even without the erased joints, the body model must have the desired segment lengths, and the total body size was kept equal to the presented in last line of table 4.3.

Further testing was done with visual comparison, as presented in figure 4.7. In this visual examination it was expected that the connection between the two joints near the erased one were connected by a single segment. It also was taken in account that this segment should be closer to the erased joint spherical distribution and the segment should start and end inside the extremity joints' spherical distributions (D_j^m and D_{j+2}^m). According to figure 4.7, this method was applied to the spine section of the model (figure 4.1 illustrates the joints), transforming the segments linking the joints in the hip (joint 1) to the spine (joint 10) and this to the neck (joint 11) into a single segment directly connecting

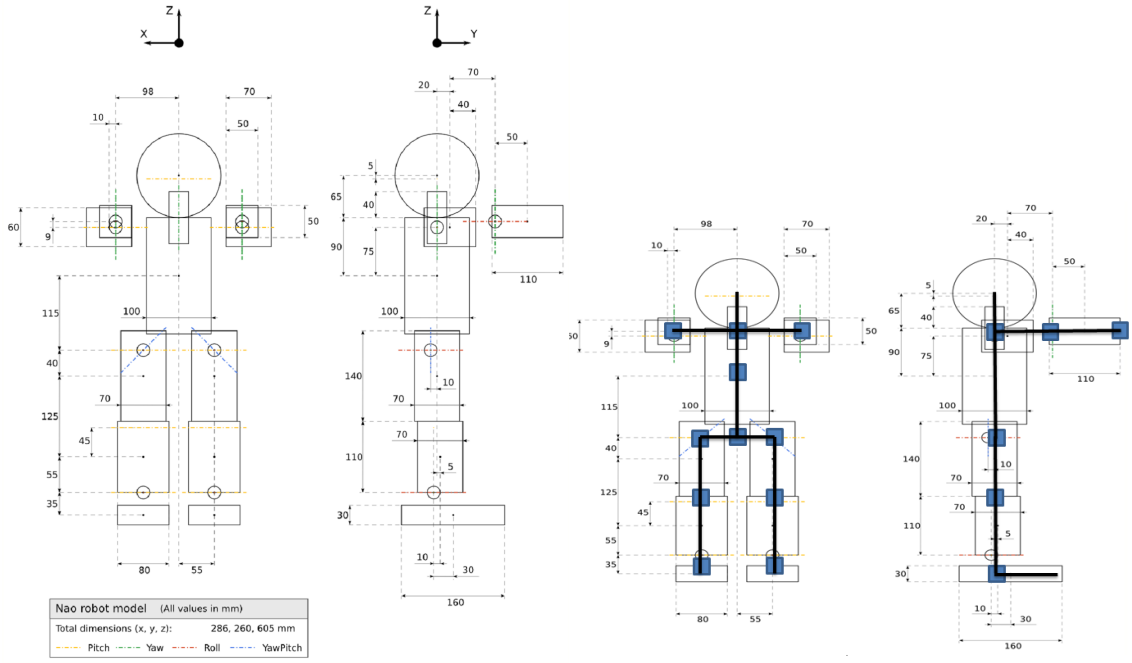


Figure 4.5: a) Simulated NAO box model with the respective size of each body part (adapted from [Sim]) (*left*). b) NAO's box model with blue lines representing the adopted body model and squares in the considered joints (the size of the hands was considered equal to the feet) (*right*).

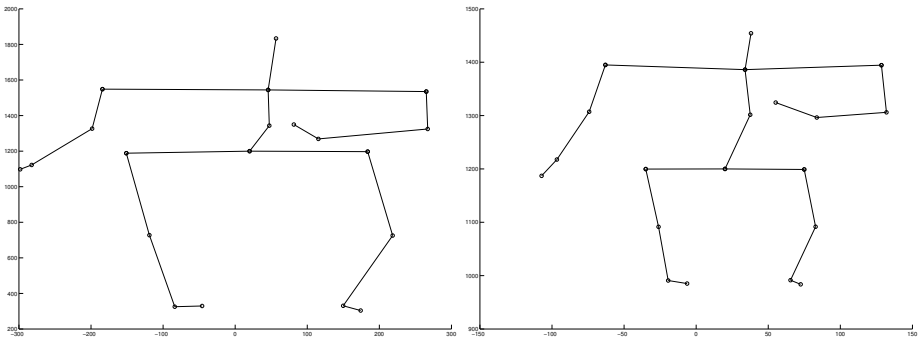


Figure 4.6: a) Key-pose generated using body with NAO's segment lengths (*left*). b) Key-pose generated using the original MoCap human model (*right*).

the hip to the neck. The figure confirms that even with the application of such operation, the total body size was kept equal to the desired.

4.2.3 Additional physical restrictions

In order to evaluate the application of the physical restrictions to the generated poses only visual evaluation was used. This visualization focused in the hip section, where the method was applied, in order to confirm if both segments in that section were collinear, as required. For such discretion, the respective hip segments and their joints' spherical distributions are presented in figure 4.8.

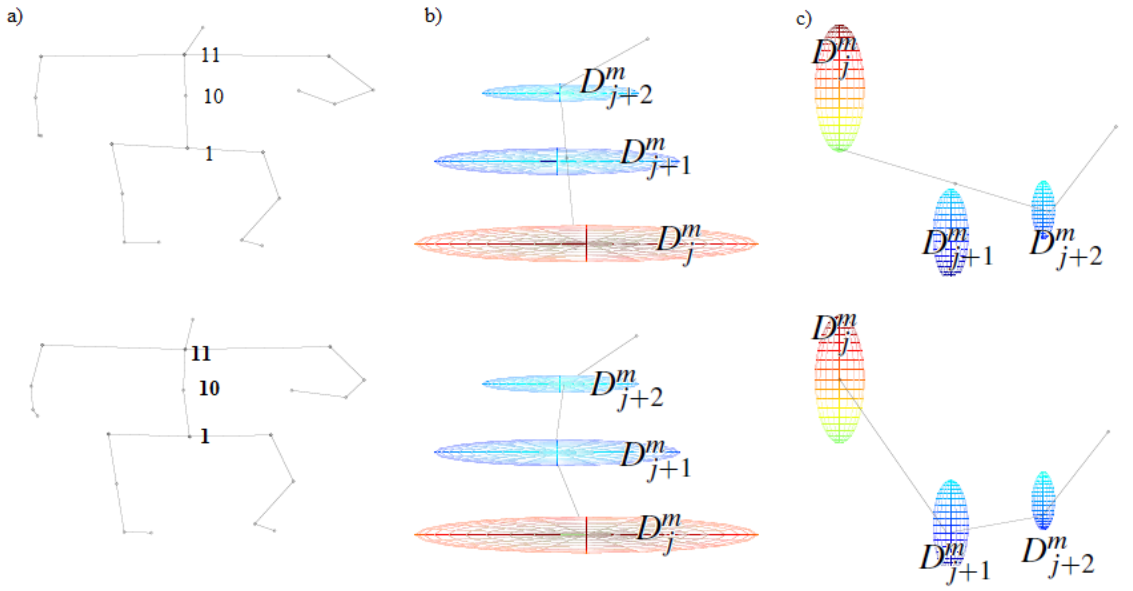


Figure 4.7: Comparison of a key-pose with the spine middle joint (*bottom images*) and without the spine middle joint (*top images*). a) Synthesized key-pose (*left*); b), c) Detail of the spine joints, with their spherical distributions, in different view angles (*middle* and *right*).

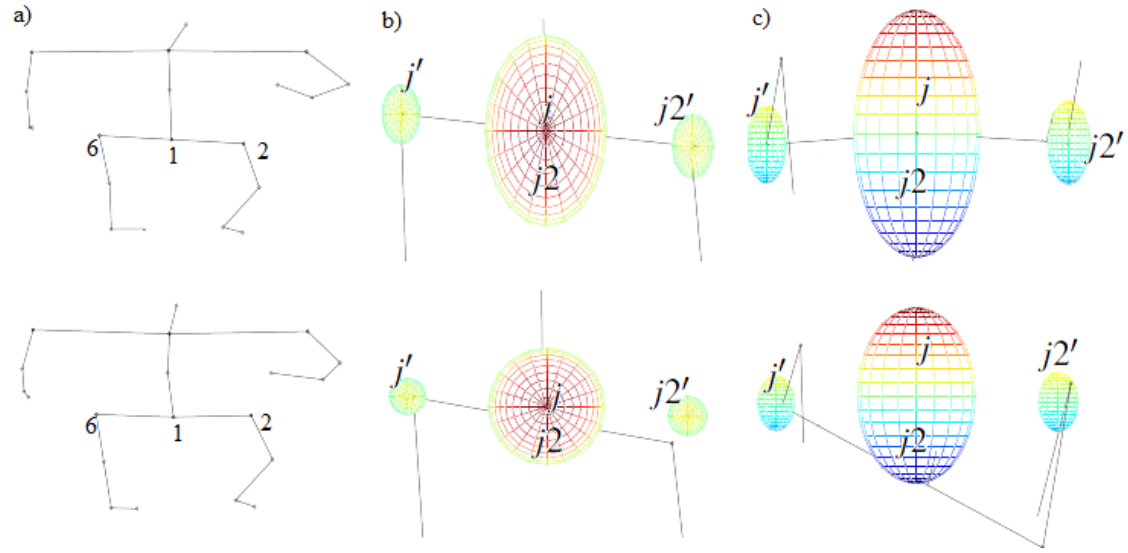


Figure 4.8: Comparison of a pose with normal hip segments (*top images*) and with the adapted hip segments(*bottom images*). a) Full pose (*left*); b) c) Detail of the hip joints, with spherical distributions, in different angles of vision (*middle* and *right*).

4.3 Key-Poses Retargeting

For evaluation of the method presented in 3.4, the similarity function 3.21 (also used for the key-pose refinement) will be applied:

$$Sim_{pose,robot} = \sum_{j}^{n} \sum_{i=j}^{n} \left| d_{pose_{i,j}} - d_{robot_{i,j}} \right| \tag{4.3}$$

The usage of this similarity function enables the comparison of the pose reproduced in the humanoid using the extracted angles, and the poses generated. The appliance of this function is done in order to compare the synthesized and adapted key-poses joints positions, $d_{pose_{i,j}}$ in (4.3), with the actual humanoid joints position that result from the angles passed onto the robot, $d_{robot_{i,j}}$ in (4.3).

To perform this evaluation, eight synthesized and adapted poses were generated. The joints positions for this poses were extracted and saved, and this joints positions were used across all the comparisons done. This positions will be further referred as r, and will be the $d_{pose_{i,j}}$ factor of our similarity evaluation.

The last step of this generation was the generation of the angles between the body parts, in order to obtain the joints spatial position, for this three sets of angles were considered and so three similarity comparisons were performed:

- Angles were extracted from the synthesized and adapted key-poses $s_{r,poseAdapted}$.
- To compare the former evaluation to a "worst-case scenario", the same similarity evaluation was applied to compare a neutral robot pose, with all joint angles equal to zero $s_{r,poseNeutral}$.
- Finally, angles were extracted from key-poses that weren't resized or adapted, $s_{r,pose}$

For the first eight generated poses, the similarity values are presented for the arms pose, for the legs and the total value (table 4.4). The values for the Legs are calculated by applying the similarity function only to the joints positions of the legs, the Arms value is done the same way but only considering the joints positions for the arms, and the Total value is done by applying the similarity function to all the joints position of the body.

It's important not only to use this similarity comparison to prove the value of the present methods, but also to have a real notion of the gain in terms of information, since it may also be possible to obtain the desired pose by refinement of a neutral pose.

Besides the numerical similarity evaluation a the visual comparison of the poses was also used. This comparison was made between the generated key-poses and the reproduction of those key-poses in the humanoid robot (figures 4.9 and 4.10), all the presented key-poses suffered adaptation.

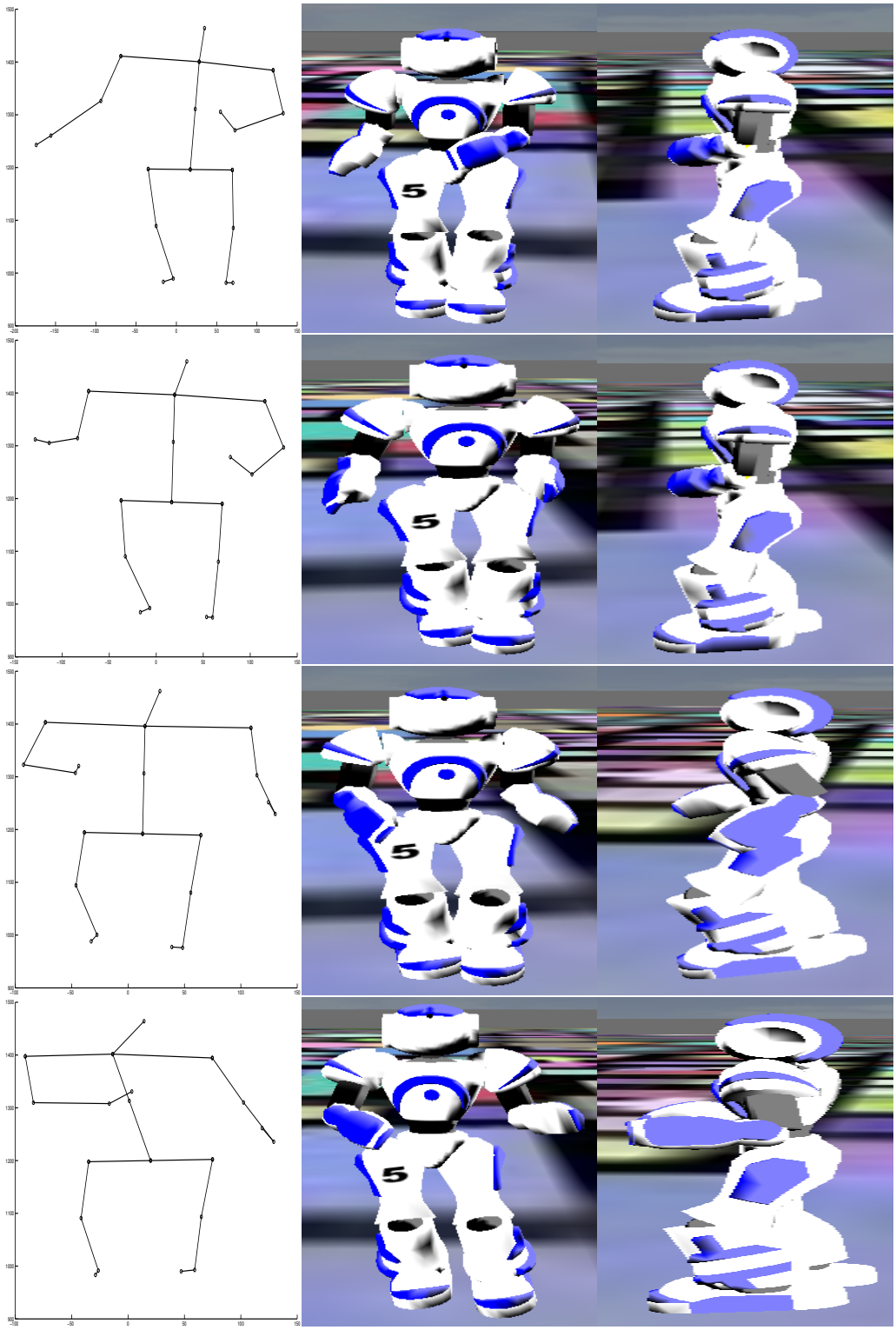


Figure 4.9: Visualization of key-pose 1 (*top*) to key-pose 4 (*bottom*) (each row represents a new pose), synthesized at "variability-4": a) Synthesized-adjusted (*left*); b), c) Retargeted to simulated humanoid NAO, in frontal (*middle*) and lateral (*right*) views.

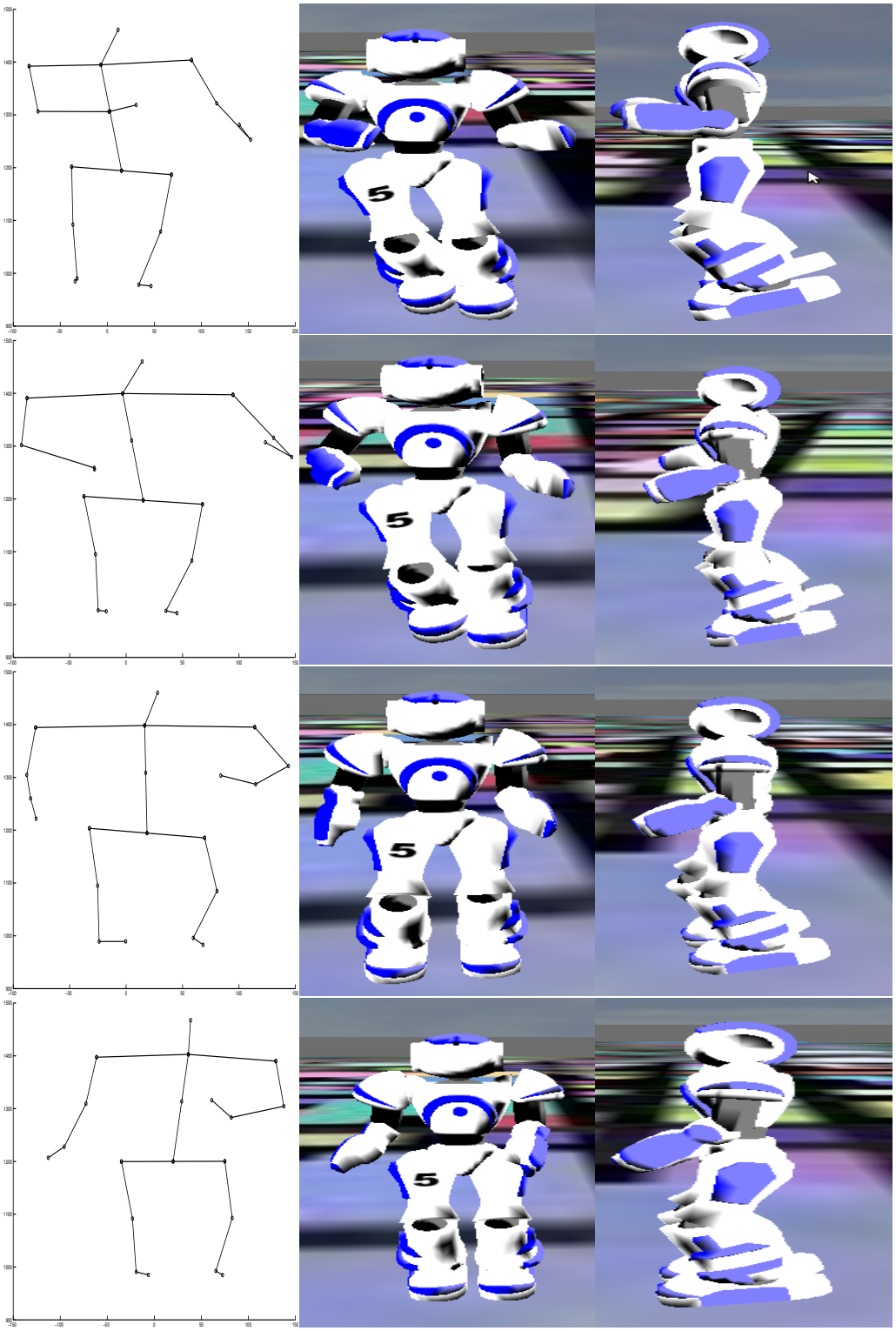


Figure 4.10: Visualization of key-pose 5 (*top*) to key-pose 8 (*bottom*) (each row represents a new pose), synthesized at "variability-4": a) Synthesized-adjusted (*left*); b), c) Retargeted to simulated humanoid NAO, in frontal (*middle*) and lateral (*right*) views.

Table 4.4: Similarity comparison of the generated poses.

	$s_{r,poseAdapted}$			$s_{r,poseNeutral}$			$s_{r,poseNormal}$		
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

4.4 Key-Poses Refinement

In order to evaluate the key-pose refinement, via optimization, the measures used in the last section were reapplied to determine the gain in terms of similarity. As already stated, the refinement was applied by recurring to the Tabu Search algorithm [Glo86], since it got the best and faster results from all the algorithms tested from [Rei10]'s framework. This algorithm was applied with the same set of parameters for every experience, as presented in table 4.5. The refinement was performed over each one of the eight retargeted key-poses, by iteratively testing different joint angles and for each test measuring the fitness of the generated robot key-pose against the synthesized-adjusted one, by means of the similarity function given by eq. (4.3).

Table 4.5: Parameters used in the Tabu Search configuration.

Parameter	Value
Number of Experiments	5
Threads	1
Minimum Change for Angles	-6.0
Maximum Change for Angles	6.0
Minimum Change for Deltas	-0.4
Maximum Change for Deltas	0.4
Number of Iterations	300
Tabu List Size	1000

The optimization process was performed over the generated key-poses given as starting conditions the retargeted key-poses (as described in Section 3.4) and also a neutral pose (with all body joint angles set to zero), which acted as the "worst-case condition". This was performed in order to measure, in terms of optimization iterations, and consequent computational time, the improvement or gain of applying a previous retargeting of

the key-poses into the humanoid robot body model instead of simply applying the optimization starting from neutral (or random) poses towards the synthesized key-poses. The best obtained similarity values for the whole 300 optimization iterations were compiled in table 4.6.

Table 4.6: Similarity comparison of the refined key-poses.

	$s_{r,poseAdapted}$			$s_{r,poseNeutral}$		
Pose	Arms	Legs	Total	Arms	Legs	Total
1	225	322	1489	274	612	2024
2	200	337	1435	359	549	3144
3	298	340	1576	283	569	2605
4	164	323	1473	364	460	1906
5	320	492	2028	461	647	3912
6	251	516	1998	259	540	2242
7	176	432	1855	362	828	2357
8	207	314	1247	213	546	1989

The key-poses were individually refined, so the variation of the similarity function values over the several performed iterations is presented in figure 4.13.

Finally, a visual comparison of all the key-poses in one complete metrical cycle (*i.e.* eight key-poses, at "variability-4"), after applying such refinement, against the previous synthesized-adjusted ones, in presented in figure 4.11 and figure 4.12.

4.5 Subjective Evaluation

In order to have a more broad evaluation, and specifically evaluate the degree of musical expressiveness evidenced by the resultant robot dance, the several steps of the work were presented and evaluated by 136 people by means of video demonstrations followed by an user-survey (inquiry presented in A). The inquiry showed videos with excerpts of the dance in the different phases of the work: human dance motion, avatar dance motion created from the interpolation of the generated and adapted key-poses, and the robot dance motion synthesized by interpolation of the refined key-poses.

The inquiry had two sections, one for the evaluation of the similarity and another for evaluation of the expressiveness and variability of the robot dance motion.

The first part had five questions that aimed to evaluate the dance similarity between different parts of the work:

- Quantify the similarity between the real human dance motion and the avatar dance motion (by interpolation of the synthesized-adjusted key-poses);
- Evaluation of the similarity between the avatar dance motion and the robot dance motion;

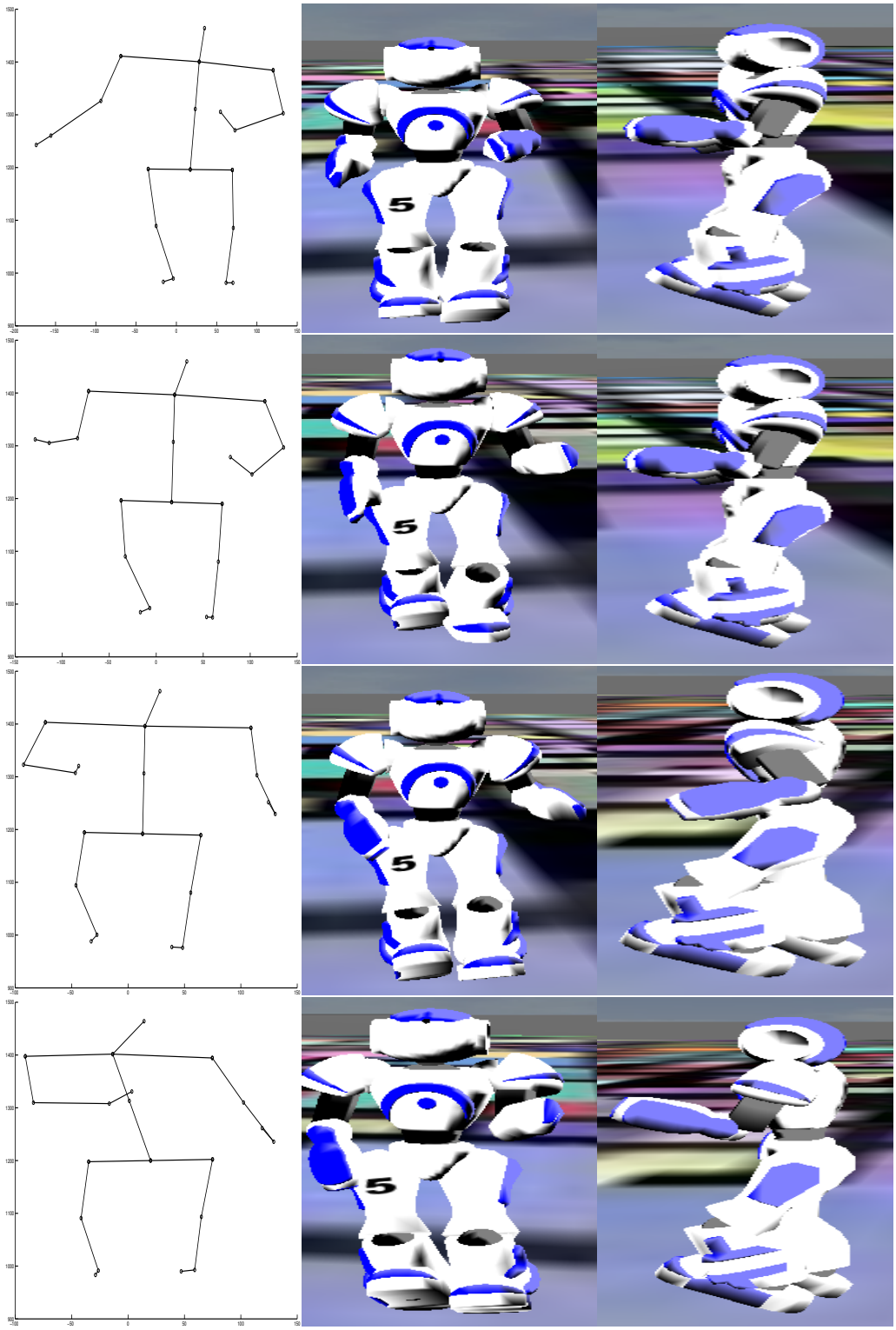


Figure 4.11: Visualization of key-pose 1 (*top*) to key-pose 4 (*bottom*) (each row represents a new pose), synthesized at "variability-4": a) Synthesized-adjusted (*left*); b), c) Refined from the retargeted to simulated humanoid NAO, in frontal (*middle*) and lateral (*right*) views.

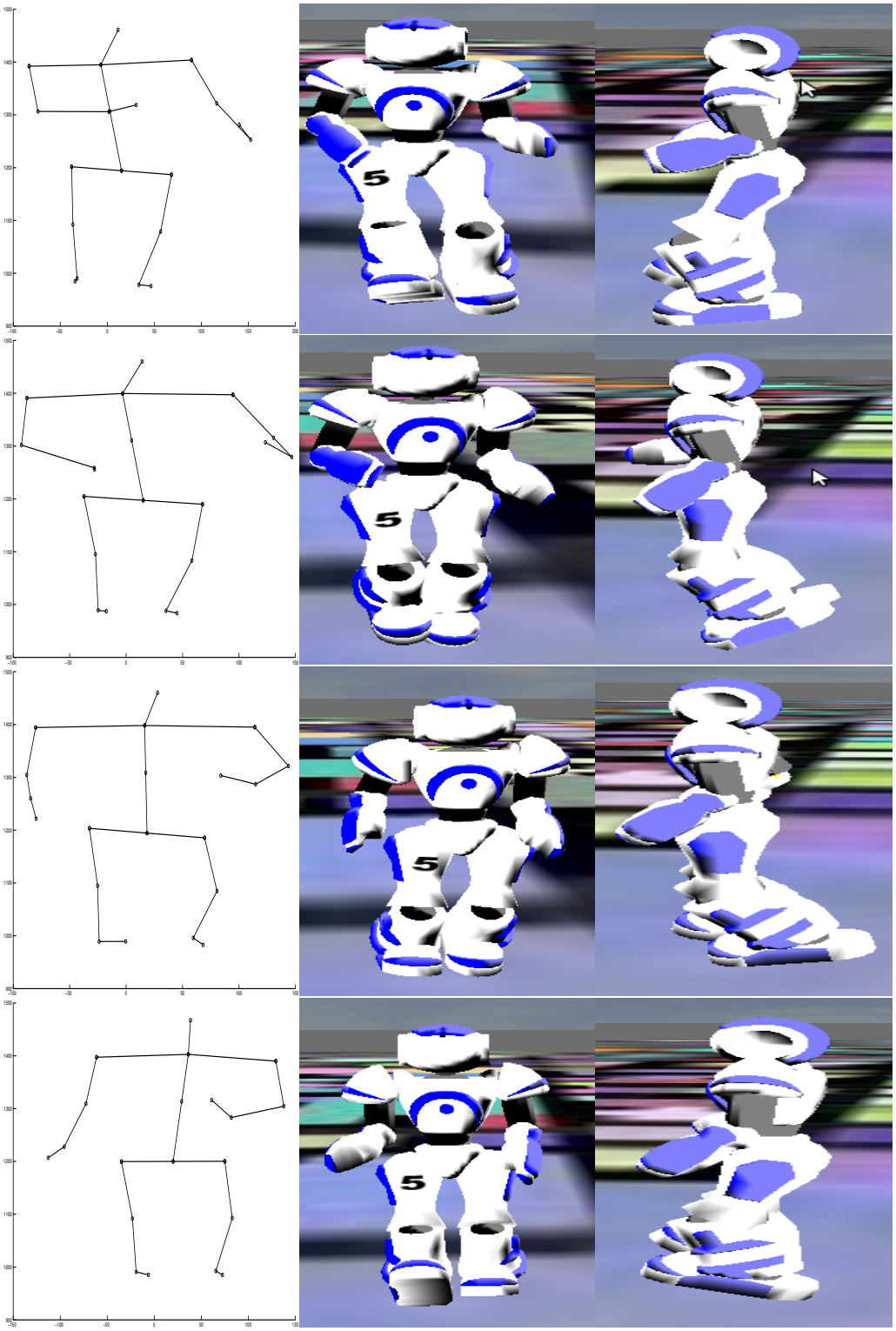


Figure 4.12: Visualization of key-pose 5 (*top*) to key-pose 8 (*bottom*) (each row represents a new pose), synthesized at "variability-4": a) Synthesized-adjusted (*left*); b), c) Refined from the retargeted to simulated humanoid NAO, in frontal (*middle*) and lateral (*right*) views.

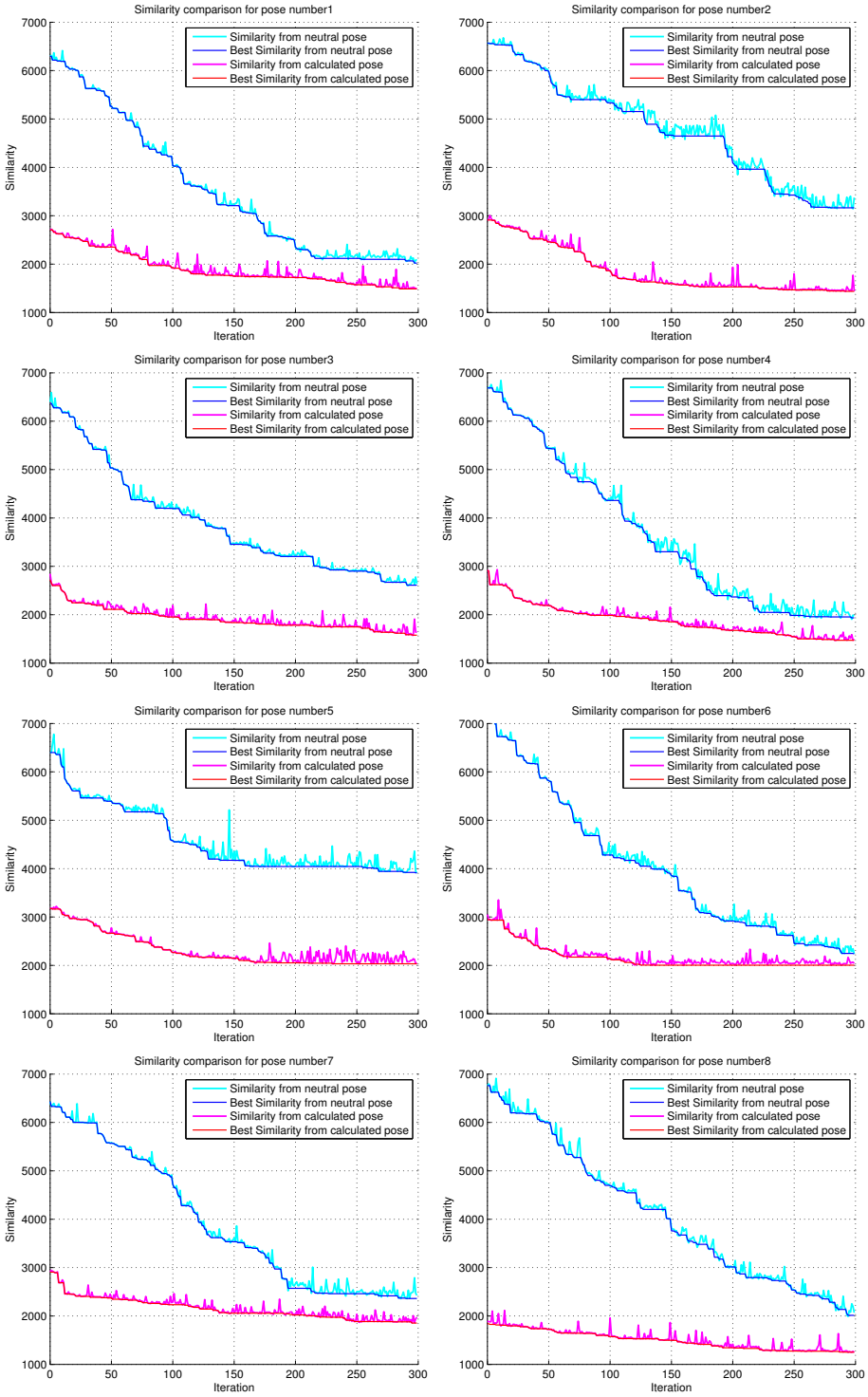


Figure 4.13: Comparison of the optimization process for eighth key-poses (full metrical cycle), synthesized at "variability-4". The degree of similarity, calculated from eq. (4.3), was measured between the key-pose under optimization and its synthesized-adjusted equivalent (*i.e.* optimal response).

- Evaluation of the similarity between the real human dance motion and the robot dance motion.

For the last comparison of the motion, two extra evaluations were specifically performed to evaluate the similarity of the leg motion and arm motion. For this section three videos were produced with small excerpts of the dance motion in analysis in each question. For the real human dance motion was used a small video of a female dancer performing the same style of Samba ("Samba-no-pé"). For the avatar dance was used a video from the dance motion generated by spline-interpolating the key-poses synthesized by the random rotations method, and adapted to the robot morphology (synthesized-adapted). And for the robot dance motion, a video was recorded of the robot NAO, in the simulation environment SimSpark, reproducing successive cycles of eight randomly generated (to translate variability), and optimized, key-poses (at "variability-4").

The second part had only two questions that aimed to evaluate the musical synchrony and degree of expressiveness evidenced by the final robot dance. For these questions, a simple video with two parts was used: in one part the robot dance without any variability in the motion (by concatenating successive cycles of the same eight key-poses) and in the second part the robot dance motion with variability, both also generated at "variability-4". This was done to understand if the variability that the methods introduce is perceptible in the final robot dance motion, and if such is relevant to improve the expressiveness of the robot dance performance. The degree of musical synchrony was also subjectively evaluated over the same video.

All the questions were answered in a Likert-scale, constituted by 5 levels of agreement, where 5 represented the maximum agreement and 1 the worst. For the several comparisons the obtained results are presented as frequency distributions in figures 4.14, 4.15 and 4.16. Besides, in table 4.7 and table 4.8, are presented the mean and standard deviation of the results for each question.

Table 4.7: Degree of Similarity. Inquiry responses average and standard deviation for the various evaluations

Evaluation	Mean	Deviation
Dancer vs Avatar	3.3	0.7
Avatar vs Robot	2.8	0.6
Robot vs Dancer (Total)	2.5	0.7
Robot vs Dancer (Arms)	3.2	0.7
Robot vs Dancer (Legs)	2.2	0.8

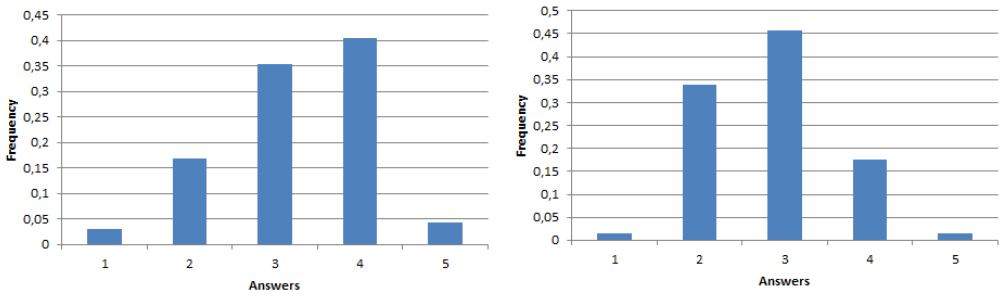


Figure 4.14: Distribution of the responses to the inquiry: a) Comparison of the similarity between the dancer motion and the avatar motion (by interpolation of the synthesized-adjusted key-poses) (1 means no similarity and 5 means equal) (*left*). b) Comparison of the similarity between the avatar motion and the robot motion (1 means no similarity and 5 means identical) (*rigth*).

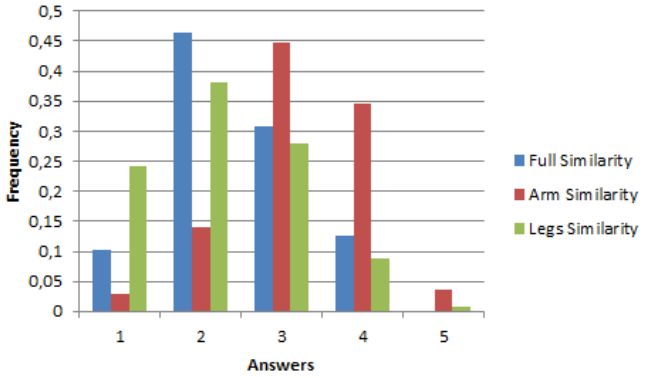


Figure 4.15: Distribution of the responses to the inquiry: Comparison of the similarity between the dancer motion and the robot motion (1 means no similarity and 5 means equal). This comparison is also done in components evaluating the arms and legs separately.

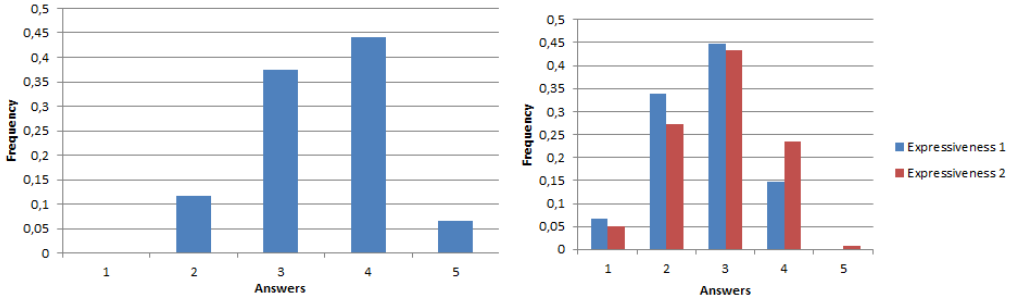


Figure 4.16: Distribution of the responses to the inquiry: a) Expressiveness of the motion in the two exerts of the robot dance motion, one with variability, in red, and another without variability, in blue, (1 means no expressiveness and 5 means very expressive) (*right*). b) Evaluation of the degree of evinced musical synchrony (1 means no synchronism and 5 means fully synchronized) (*left*).

Table 4.8: Degree of Musical Expressiveness. Inquiry responses average and standard deviation for the various evaluations

Evaluation	Mean	Deviation
Motion-Music Synchronism	3.5	0.6
Expressiveness without variability	2.7	0.6
Expressiveness with variability	2.9	0.7

4.6 Discussion

4.6.1 Key-Pose Synthesis with Variability

In terms of comparison of the key-pose synthesis methods, table 4.1 shows that the correlation between the method based on random coordinates and random rotations is almost equal to 1, indicating this way that both the synthesized dance sequences are almost identical. The comparison of the correlation of both methods with the original dance sequence is also almost equal. This results seem to validate that the methods produce equally similar dance motions to the original sequence. On the other side, table 4.2 demonstrates the advantage of the method based on random rotations, since the body model is equal to the expected, contradicting the body size's difference to the original imposed by the random coordinates method. Besides, even if its by a small difference, the method to generate key-poses based on random rotations is more accurate. With the visual comparison presented in figure 4.2, mainly in the comparison of the first pose (*upper and bottom left*) and figure 4.3, its visible that there is a slight difference in the shoulder section between both methods. Further visual comparison showed that the difference in the body total size exists only in the shoulders and hips. The results validate the proposed method, showing that is more accurate than the previous method without significant losses in the dance sequence similarity.

4.6.2 Key-Poses Morphological Adaption

The method for resizing the body model by scaling and translation of the joints spherical distributions seems effective, presenting the desired results. Even at different sizes, the synthesized key-poses seem to keep the same posture across all body. The method is robust and allows to easily mold the spherical distributions, in size and spatial position, fitting them to a different body, without compromising the shape of the resulting dance. In terms of visual evaluation, figure 4.4 shows that even with different scaling the bodies presents similar poses. This is also confirmed by figure 4.6, when building a body with the robot NAO size. The MoCap's and NAO's bodies have significant differences in terms of segment lengths, but present a good degree of similarity in the representation of ever key-pose. The obtained body segment lengths for the NAO body, based on figure 4.5,

seem consistent with the simulation body, but the 3D dimensionality of the real robot's body parts may impose self-collision problems in future work.

The methods for "erasing" a joint, in this case applied to the spine, and the application of the physical restriction, in the hip section, introduce some changes in the posture, but this changes seem to be localized in the affected areas by the work of this methods. Such changes should be expected, since the application of this restrictions to a certain body part effectively changes the posture of that body part. Either way, it's important that the remaining body parts still reproduce the posture with great degree of similarity, which seems to be confirmed by figure 4.7 and figure 4.8. While figure 4.7 confirms a good solution to the problem, it also seems that this methods impose a loss of the variability in the construction of the spine, mainly because this method behaves in a greedy way, trying to find a good solution which reduces the alternatives for choosing a random rotation in the spine segments. From figure 4.8 it's clear that the application of these 2 methods has its disadvantages, since it was possible to generate valid hip segments, for the presented hip spherical distribution, but as the application of the method to generate the spline moves the initial point (joint 11) it makes impossible to generate valid segments (*i.e.* segments whose joints fit their respective spherical distributions) for the hip section. As the two methods work separately, they only try to archive its own goals and not a better global solution, eventually making it only an optimal/good solution to the spine or hip problems.

In terms of the key-pose angle extraction, for the robot joint rotations, the overall results seem good, in terms of visual and numerical evaluation, especially for the arms. Yet, for the legs, the results aren't so good, mainly due to the hip rotation in the transverse plane (that gives the feet and legs orientations), which seems to not always be correctly calculated. This error is also aggravated because of the angular limits of the correspondent humanoid joint, avoiding certain types of positions for the legs.

Table 4.4 shows better similarity results for the extracted angles from the adapted key-poses that 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%.

4.6.3 Robot Key-Poses Refinement

Using [Rei10]'s framework and the configuration presented in table 4.5 to refine the key-poses, the similarity value was decreased by almost half in most cases. As suggested by figure 4.13, the application of the same refinement over the neutral poses only was able to obtain the starting similarity values from the extracted angles (figure 4.13). This demonstrates the advantage of the presented method for extracting the angles for the key-poses as a pre-refinement step, since better results are obtained faster than by simple optimization over a neutral pose. By analysis of the refined key-poses, figure 4.11 and

figure 4.12 suggest that the difference in the arms, between the retargeted and refined key-poses, isn't very perceptible and in the numerical evaluation the improvement of the similarity for the arms is low. These results for the arms are in line with the results from the angle extraction, where the arms also were very similar to the original pose. In the legs, the changes are more clear, visually and numerically, but in some poses the gain in similarity isn't too clear in the visual evaluation.

4.6.4 Subjective Evaluation

Finally the subjective evaluation mainly recurs to visual evaluation of the resultant dance motion, and as the inquiry reflects the opinion of a larger number of persons it helps to understand the possible audience opinion and evaluation over the overall robot dance performance.

The questions were made to individually evaluate the several applied methods. Firstly we inquired the subjects about the degree of similarity between the real dancer and the avatar motion (by interpolation of the synthesized-adapted key-poses), in order to evaluate the reliability of the analysis and representation of the original dance motion, and the synthesis and adaptation of the key-poses. In this evaluation, the mean of the answers was positive and most answers were also positive (see figure 4.14*a)* and first line of table 4.7), which showed good acceptance of the synthesis and adaption of the key-poses, resultant motion via key-poses' interpolation, in comparison to the real Samba dance. These results helped to support the already obtained results for the correlation between the original dance motion and the dance motion generated from the random rotations key-pose synthesis method (presented in table 4.1).

The evaluation of the similarity between the avatar and the robot, that mainly evaluates the retargeting and refinement phases, had the mean results almost on the border from the negative to the positive (second line of table 4.7). The results are also mostly distributed between the 2, 3 and 4 values of the scale (figure 4.14 *b)*), and this shows that there is some resemblance between the avatar and the robot dance motion for the great majority of the inquired people. This visual evaluation suffers from the physical and aesthetic differences between the avatar figure, that is composed of lines (stick figure), and the robot figure, that is more dense and complex. These differences may prejudice the visual comparison. The evaluation of the dancer versus the robot wasn't as positive as the other results, mainly because of the already referred differences in the leg sections, since the arms had a fairly positive evaluation (figure 4.15).

The evaluation of the full body is in the border from positive to the negative (third line of table 4.7). These results suggest that the leg and hip section, being so important in the Samba dance, and being so different from the human to the robot, impose several differences in the final dance motion that reduce the similarity level. Ultimately, the

inquiry results aren't optimal, but show that there is a fair degree of similarity between the original dance motion and the robot dance motion, the results also are interesting because the video of the dancer used in the comparisons is different from video of the motion capture data dance. Since the original dance motion, from the motion capture data used in this thesis, was performed without embellishments [ONG+10], and the video presented in the inquiry had quite a few embellishments, it potentially prejudiced the comparisons. Even so, is good to reinforce that all the comparison results from the inquiry were still positive, even if we are comparing the robot dance motion with a human dance motion different from the one from which this was built.

Concerning the evinced degree of musical expressiveness, in terms of beat-synchrony, the results seem to be good for the vast majority of the inquired people (see figure 4.16*b)* and table 4.8), showing that the interpolation method in combination with the beat-timings given by the dance TGA representation is a valid approach, for maintaining the original dance motion-music synchrony.

Finally the evaluation of the evinced expressiveness between the robot dance motion with variability and without variability shows that the difference is not very visually perceptive, since most of the answers evaluated the 2 excerpts as being almost equal or exactly equal (see figure 4.16*a)* and last two lines of table 4.8). This suggests that most of the inquired subjects didn't noticed any difference between the two excerpts, which is justified because of the small differences in the same poses from successive metrical cycles, presenting only some varied angles, and those presenting a too low degree of variation (1 to 10 degrees per joint). We can then argue that the poses couldn't present significant variations to be visually detected, and as so further studies need are needed to evaluate the effect of introducing variability in the overall expressiveness of the resultant robot dance.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The creation and generation of humanoid behaviors isn't a simple, or fast, process. This thesis presents a form of transforming human capture dance motion into humanoid motion.

The study presented in this thesis is mainly focused in cyclic dance motions (popular dance styles), where we can find a close musical-motion relation. Initial work was made over a new key-pose synthesis method, that allows the construction of key-poses with variability. This synthesis method, based on random rotations, proved to be as good as the original method based on random coordinates, and, contrarily to the former, this method didn't compromised the fixed geometry of the human/humanoid body model.

The presented methods for the adaptation of the key-poses achieved the expected results without compromising the key-poses shape or the body model morphology. The resizing method was implemented in a "universal" way and provided valid results, showing that it can adapt poses to any desired humanoid body model. The other morphological adaption methods were specifically required by the target humanoid robot morphology, and, as the former, they achieved the desired results, without introducing errors in the key-pose body model. Yet, although these methods also aim to solve generic morphological adaptation problems, their application to other humanoid robots should still be explored and tested. The morphological adaptation, applied to the target humanoid robot NAO, proved to be valid at maintaining the key-poses shape and expression while successfully transforming them to the target morphology.

As for the retargeting, interesting results were also obtained, with an overall good degree of visual similarity between the original key-poses and the humanoid equivalents. This similarity was greater in the arms, while the legs suffered from the several difficulties found at the robot's hip section, and from the lack of joints and DoFs presented by the body and hip section. Also, some aspects about the dimensions of the humanoid body

parts seem to present difficulties in the visual comparison between the robot reproduction of the key-poses and their former stick figure representation. The adopted retargeting method presented an efficient and simple approach for extracting the robot joint angles from the synthesized-adapted key-poses, allowing fast computation and easier creation of humanoid robot behaviors.

Finally, for the key-poses' refinement, a metric to evaluate the similarity between two poses was presented. This metric was used as a fitness function to our optimization problem, and as a form of comparison to evaluate the degree of similarity of the achieved robot dance compared to the original. The presented similarity function envisions to quantify the deviation between two poses by comparing the relation between all their body joints in terms of spatial distances. Even if it lacks an absolute zero value to have a scale sensation in this evaluation, it numerically demonstrated the gain in similarity when there was also a perceptive visual gain. The optimization allowed to obtain more similar (*i.e.* refined) key-poses, in the humanoid robot, to the original ones.

The final experiment, in the form of an user-survey, subjectively evaluated all the developed work by means of several videos with the results of some of the presented methods in this thesis. This aimed to obtain feedback over the similarity and musical expressiveness evinced by the resultant robot dance performance. In terms of similarity with the original dance, the results seem to validate the approach, even if the dance excerpts weren't considered exactly equal. In terms of musical expressiveness, the approach for replicating the original dance rhythmic synchrony seemed to be validated. Yet, further studies are needed to correctly measure the effect of introducing variability in order to improve the overall expressiveness of the generated robot dance.

Ultimately, the Samba dance motion data used in this thesis provided several challenges, and the final generated robotic dance motion may had suffered from the lack of hip movement and some physical constraints of the target robot, which are of most important to this specific dance style. Even facing this difficulties, the final robot poses, generated from the several proposed methods, replicated with good similarity the original poses, and consequently the original Samba dance.

5.2 Future Work

The most obvious and necessary work is to ensure the humanoid biped balance. In order to generate and fully perform the dance motion, humanoid balance must be imposed and self-collisions must be avoided. The upgrade of the refinement's fitness function may help to solve the balance problems via optimization. This could be achieved by changing the

evaluation function as follows:

$$Sim_{a,b} = \left(\sum_{j}^{n} \sum_{i=j}^{n} \left| d_{pose_{i,j}} - d_{robot_{i,j}} \right| \right) * \frac{1}{time} \tag{5.1}$$

, where *time* is the amount of time where the robot had balance. As such, the greater the time the lower the evaluation function, and, consequently, the optimal the generated motion would be. This would help to find, by optimization, a trade-off between similarity and balance. Self-collision avoidance can also be done by optimization, by ensuring that the distance between the body parts is higher than the sum of there size in the correct distance.

Balance maintenance can also be dynamically achieved by adjusting the ZMP trajectories [OSKI10], [ZHD+04], [KKYO09], constraining the total vertical inertia force to be equal to zero, and so keep the robot balance. The calculation of the center of gravity, and/or center of pressure, [Ste07], [SA09], can also be used to ensure the humanoid balance.

The method to determine the intersection between two spheres presented in [ONG+10] and [ONG+11] may also be upgraded in order to allow faster computation of the method. This may prove extremely useful for the application of the presented work in real time. This improvement may prove to have advantages in the search for the variation of the rotation between two body segments.

Further work may also be done in order to attempt to merge, or create a direct bond, between the method to generate the key-poses based on random rotations and the angle extraction, or retargeting of the dance. This would increase the efficiency of the process and would take full advantage of the method developed to generate the key-poses based on random rotations. Further analysis over morphological adaptation, and appliance to other robotic humanoid morphologies, may help to validate and complete the developed work. This can also be achieved by applying this technique to other dance styles presenting dance patterns intimately correlated to musical metrical qualities.

Ultimately, the proposed techniques may be extended and applied over different dance motions and even over different kinds of motion, allowing the simple creation and reproduction by a humanoid robot of different kinds of basic human behaviors. Besides, the presented techniques for angles extraction and motion refinement, may be also useful for the easy creation of user-defined humanoid poses.

References

[dA03] Edilson de Aguiar. Character animation from a motion capture database. Master's thesis, University of Saarland, 2003.

[Dic] Lee Dickholtz. Meta motion. Online at: http://www.metamotion.com/. Consulted on 17th January 2011.

[DVMC10] J. Dauwels, F. Vialatte, T. Musha, and A. Cichocki. A Comparative Study of Synchrony Measures for the Early Diagnosis of Alzheimer's Disease Based on EEG. *NeuroImage*, 49(1):668–93, January 2010.

[Ebe06] Russell Eberhart. Particle swarm optimization particle swarm optimization the inventors : Particle swarm optimization particle swarm optimization. *Science*, 18(17):1–12, 2006.

[GD05] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer Music Journal 29(1)*, pages 34–54, 2005.

[GHB+08] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The NAO Humanoid: A Combination of Performance and Affordability. *Computing Research Repository (CoRR)*, abs/0807.3:10, 2008.

[Glo86] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[GWSF06] Fabien Gouyon, Gerhard Widmer, Xavier Serra, and Arthur Flexer. Acoustic cues to beat induction: A machine learning perspective. *Music Perception 24(2)*, pages 181–194, 2006.

[Hea] Healthbase. Online at: http://www.healthbase.com/resources/. Consulted on 25th May, 2011.

[Hol75] John Holland. *daptation in Natural and Artificial Systems*, volume Ann Arbor. University of Michigan Press, 1975.

[KGV83] S Kirkpatrick, C D Gelatt, and M P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KKYO09] 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.

[Kla03] Anssi Klapuri. Musical meter estimation and music transcription. In *Cambridge Music Processing Colloquium*, pages 40–45. Citeseer, 2003.

[KPS03] 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.

[LJJ96] F. Lerdahl, R. Jackendoff, and R.S. Jackendoff. *A generative theory of tonal music*. The MIT Press, 1996.

[LOGPR08] João Lobato Oliveira, Fabien Gouyon, and Luís Paulo Reis. Towards an interactive framework for robot dancing applications. In *ARTECH – 4th International Conference on Digital Arts*, pages 52–59, 2008.

[LOGPR09] João Lobato Oliveira, Fabien Gouyon, and Luís Paulo Reis. Robot dance based on online automatic rhythmic perception. In *DSIE'09 – 4th Doctoral Symposium on Informatics Engineering*, 2009.

[LR07] Nuno Lau and Luís Paulo Reis. High-level coordination methodologies in soccer robotics, robotic soccer. *Itech Education and Publishing*, pages 167–192, 2007.

[Mog99] N. Mogey. So you want to use a likert scale. *Learning Technology Dissemination Initiative*, 1999.

[Nat] Naturalpoint: Optitrack. Online at: http://www.naturalpoint.com/optitrack. Consulted on 5th February, 2011.

[NL10] 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.

[NNIY02] 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.

[NNY+03] 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.

[NOI+05] 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.

[OCFT+08] F Ofli, C Canton-Ferrer, J Tilmanne, Y Demir, E Bozkurt, Y Yemez, E Erzin, and A M Tekalp. Audio-driven human body motion analysis and synthesis. *Engineering*, pages 2233–2236, 2008.

[ON91] Nihat Özkaya and Margareta Nordin. Fundamentals of biomechanics: equilibrium motion and deformation. *Springer*, 1991.

[ONG+10] João Lobato Oliveira, Luiz Naveda, Fabien Gouyon, Marc Leman, and Luis Paulo Reis. *Synthesis of Dancing Motions Based on a Compact Topological Representation of Dance Styles*. IEEE, Taipei, Taiwan, 2010.

[ONG+11] João Lobato Oliveira, Luiz Naveda, Fabien Gouyon, Marc Leman, Luis Paulo Reis, and Paulo Sousa. A Spatiotemporal Method for Synthesizing Expressive Dance Movements of Virtual Humanoid Characters. *submitted to Special Issue on Music Content Processing by and for Robots from the EURASIP Journal on Audio, Speech, and Music Processing*, page (To appear), 2011.

[OR04] Oliver Obst and Markus Rollmann. Spark – a generic simulator for physical multiagent simulations. In *Multiagent System Technologies Proceedings of the MATES 2004*, volume 3187, pages 243–257. Springer, 2004.

[OSKI10] 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.

[Par94] R. Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception*, 11(4):409–464, 1994.

[PGLLPR09] Hugo Picado, Marcos Gestal, Nuno Lau, and Ana Maria Tomé Luís Paulo Reis. Automatic generation of biped walk behavior using genetic algorithms. *IWANN (1)*, pages 805–812, 2009.

[Pic08] Hugo Rafael Picado. Development of behaviors for a simulated humanoid robot. Master's thesis, Universidade de Aveiro, 2008.

[Rei10] Luís Rei. Optimizing simulated humanoid robot skills. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2010.

[RL01] Luís Paulo Reis and Nuno Lau. Fc portugal team description: Robocup 2000 simulation league champion. *RoboCup 2000: Robot Soccer World Cup IV*, pages 29–40, 2001.

[RNKI05] 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.

[RNKI06] 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. IEEE, 2006.

[RRL11] Luís Rei, Luís Paulo Reis, and Nuno Lau. Optimizing simulated humanoid robot skills. *Proceedings of the 11th International Conference on Mobile Robots and Competitions*, pages 84–89, 2011.

[SA09] B. Stephens and C. Atkeson. Modeling and control of periodic humanoid balance using the linear biped model. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 379–384. IEEE, 2009.

[Sch10] Adriana Schulz. Character animation from motion capture data. Technical report, Insituto Nacional De Matemática Pura e Aplicada, 2010.

[Shi06] Takaaki Shiratori. *Synthesis of Dance Performance Based on Analyses of Human Motion and Music*. Phd thesis, University of Tokyo, 2006.

[Sim] Simspark comunity. Online at: http://simspark.sourceforge.net/wiki. Consulted on 1st June, 2011.

[SKNI07] 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. IEEE, October 2007.

[SNI06a] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Dancing-to-Music Character Animation. In *EUROGRAPHICS*, volume 25, pages 449–458, September 2006.

[SNI06b] 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.

[SOR11] Paulo Sousa, João Lobato Oliveira, and Luis Paulo Reis. Humanized robot dancing: Humanoid motion retargeting based in a metrical representation of human dance styles. *EPIA 2011*, 2011. To appear.

[Ste07] Benjamin Stephens. Integral control of humanoid balance. In *IROS 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems.*, pages 4020–4027. IEEE, 2007.

[TB10] Petri Toiviainen and Birgitta Burger. *MoCap Toolbox Manual*. University of Jyväskylä: Jyväskylä, 2010.

[Zat98] Vladimir Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics, 1998.

[ZHD+04] 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.

Appendix A

Subjective Evaluation: Inquiry

Bellow are presented the questions of the used inquiry. The questions were divided in two different groups (as explained in 4.5): a similarity evaluation group (figure A.2) and a expressiveness analysis group (figure A.1).

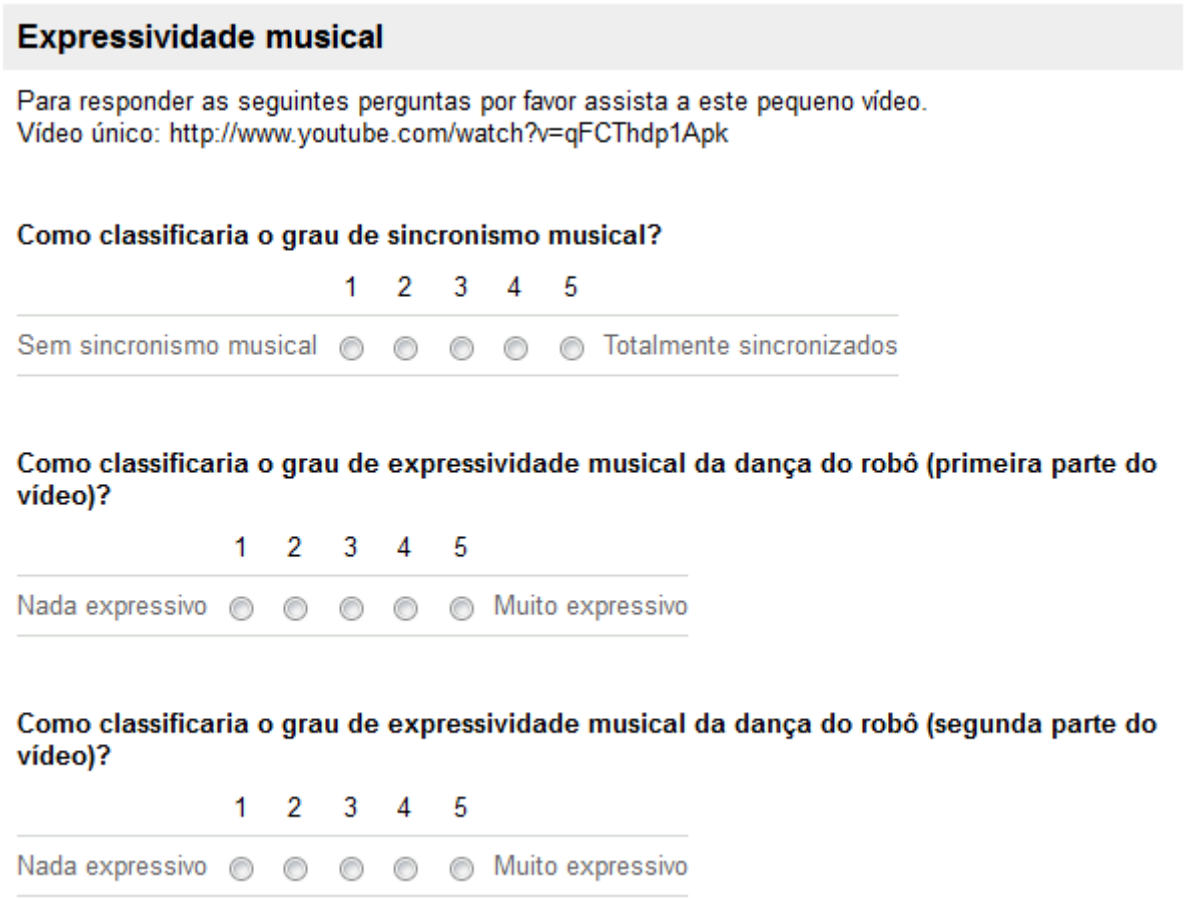


Figure A.1: Inquiry questions about expressiveness evaluation.

Similaridade

As seguintes perguntas visam analisar a similaridade, ou parecença, entre diferentes excertos de dança.

Como classificaria a semelhança da dança efectuada pela bailarina e pelo avatar? *
Vídeo: http://www.youtube.com/watch?v=n8vPRpiCilk

	1	2	3	4	5	
Não existe semelhança.	○	○	○	○	○	Idênticos

Como classificaria a semelhança da dança efectuada pelo avatar e pelo robô?
Vídeo: http://www.youtube.com/watch?v=8nf4lE86nFg

	1	2	3	4	5	
Não existe semelhança.	○	○	○	○	○	Idênticos

Como classificaria a semelhança da dança efectuada pela bailarina e pelo robô? *
Vídeo: http://www.youtube.com/watch?v=BqZF7z4GE48

	1	2	3	4	5	
Não existe semelhança.	○	○	○	○	○	Idênticos

Em relação aos braços unicamente, como classificaria a semelhança dos movimentos do robô com a bailarina? *

	1	2	3	4	5	
Não existe semelhança.	○	○	○	○	○	Idênticos

E em relação às pernas? *

	1	2	3	4	5	
Não existe semelhança.	○	○	○	○	○	Idênticas

Figure A.2: Inquiry questions about similarity evaluation.