Proceedings of the 2nd International Symposium
on Adaptive Motion of Animals and Machines,
Kyoto, March.4-8, 2003

Recognition and Generation of Leg Primitive Motions for Dance Imitation by a Humanoid Robot

Shinichiro Nakaoka[1], **Atsushi Nakazawa**[2], **Kazuhito Yokoi**[3] **and Katsushi Ikeuchi**[1]

[1]The University of Tokyo, Tokyo, Japan, nakaoka@cvl.iis.u-tokyo.ac.jp
[2]Japan Science and Technology Corporation
[3]National Institute of Advanced Industrial Science and Technology

Abstract

We have attempted to realize a dancing humanoid robot which can imitate dance performances by a human. It is impossible to realize a performance including leg actions on a robot by direct conversion of motion data captured from a dancer. Constrains of robot legs and the dynamic balance must be considered. To solve this problem, this paper proposes a framework based on primitives of leg actions. First, a sequence of primitives is recognized from the captured human motion. Then motion data for a robot are generated from the primitive sequence. The process considers both the constraints and the dynamic balance. This framework brings high flexibility to adapt the performance to various stage conditions or recomposed choreography. Generated robot motions were tested in OpenHRP dynamics simulator. A robot successfully imitated dances for both the upper and lower body.

Figure 1: Jongara-Bushi

1. Introduction

Our project has attempted to develop a total technology to preserve traditional folk dances which have been disappearing for the lack of successors[3]. Currently, our target dances are Japanese folk dances such as *Jongara-Bushi* as Fig. 1.

To preserve dances means to reproduce the dance performances again. A typical solution for reproduction is CG animation or virtual reality. For example, Yukawa et al.[6] have proposed 'BUYO-FU' system, which enables composition of choreography and reproduction by CG animation. However, these methods are insufficient because 'watching a dance' is not 'watching a CG animation'. Performances by an actual dancer which brings stronger impression than CG animation or virtual reality are necessary. In this study, a humanoid robot is used to realize actual performances.

Pollard et al. [4] have developed a method to import human dance motion into a robot. Motion data acquired through a motion capturing system are converted to joint angle trajectories of a robot, and the trajectories are modified to be feasible ones within constraints of the robot. However, their study does not deal with leg motion. A body of their robot is fixed to a stand. On the other hand, this study attempts to realize whole body performances by an actual robot, including leg actions. In generation of leg motion, to use motion data which are directly converted from captured motion is not easy, because leg motion of a robot is too restricted to apply such data. Leg structure of present robots easily causes self collision or overrun of movable ranges. Avoiding those situations, leg motion must also consider balance keeping.

We attempt not to develop a robot just as a dance recorder and player, but to develop a robot which can recognize abstract information of dance by observation and imitate dances on the basis of recognition results. This approach brings high flexibility to performances. For example, performances can be adapt to various stage condition, recomposed choreography sequences, or interaction with a human.

Inamura et al. [1] have proposed a framework of imitation, called *mimesis loop*. Their method can deal

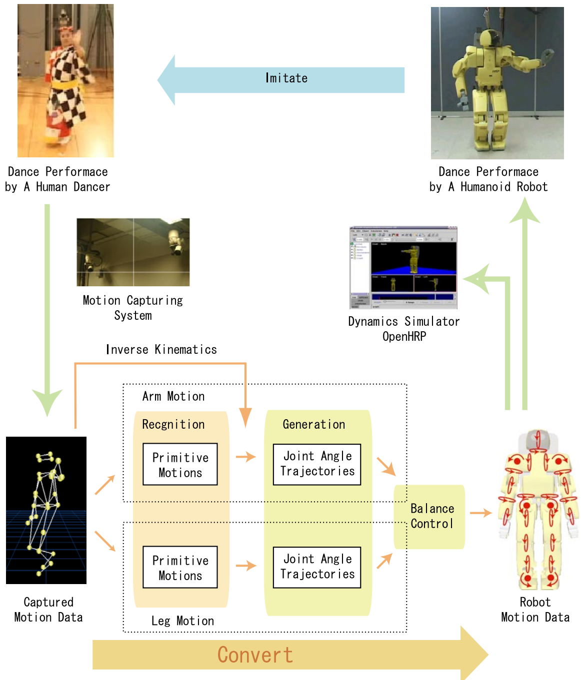

Figure 2: Overview of the System

with general kinds of motions, but detailed feasibility of actual robot performance is not focused on. On the other hand, our method is specialized in dance performance and focuses attention on generating feasible motions on an actual robot.

In this paper, we proposes a framework based on primitives of leg actions. The primitives are precedently defined and human dance performance is recognized as a sequence of primitives. Then robot motion is generated only from information of the primitive sequence. In this framework, feasible motions are simply generated than adapting direct converted data from raw motion data.

2. System Overview

A robot motion is converted from a human dance performance through the process in Fig. 2.

First, dance motion is acquired as digital data from a human performance by means of a motion capturing system. We use VICON, an optical type motion capturing system. Time series positions of 30 joint markers are acquired at the rate of 200 frames per second. Currently, our target dances are Japanese folk dances such as Jongara-Bushi and Aizu-Bandaisan.

Since captured motion data cannot be directly imported into a robot, the data must be converted to joint angle trajectories of a robot. This process mainly consists of two parts: recognizing dance motion from captured motion data and generating motion data for a robot. By the recognition process, symbolic representation of the dance is extracted. The representation is composed of *primitive motions*, which is a minimal unit of dance motion.

In the present system, conversion processes are different between upper body motion and leg motion. Upper body motion is generated by using both a raw motion data and extracted primitives [3]. First, joint angles are directly converted from raw motion data by inverse kinematics. Then the data is modified to adapt constraints of a robot. The extracted primitives are used to acquire better expression in this time. On the other hand, leg motion is generated only from the information of the extracted primitives. Finally, upper body motion and leg motion are integrated and a waist trajectory is modified to keep the dynamic balance. Detailed process for leg motion is described in the following sections.

As the final step, a robot performs a dance according to the generated joint angle trajectories. We use HRP robot platform which has common control interface between virtual robots for simulation and actual robots[2]. The validity of the generated motion data is tested by simulation on a virtual robot. Currently we use humanoid robot HRP-1S[5].

3. Recognition of Leg Primitive Motions

3.1. Primitive Definition

In our target dances, three basic leg actions are observed as primitive motions: standing (STAND), stepping (STEP), and squatting (SQUAT). Figure 3 shows these actions. Each primitive has parameters which are required to recreate leg motion for a robot. Although there may be other basic actions such as jumping or spinning in other dances, we currently focus attention to above three actions for present targets.

STAND represents the motion that both legs support the body and keep balance. This primitive has parameter of waist height and standing period.

STEP represents one stepping motion. To be precise, one foot is upped and downed on a floor while the other foot is supporting the body. The former foot is *swing foot* and the latter foot is *support foot*. To express various step actions, a period of primitive is expressed by medium time at which a swing foot takes the highest position, and final time at which the swing foot lands on the floor again. Primitives has states at

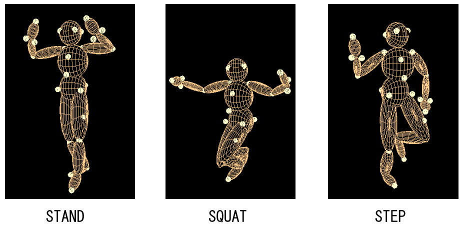

Figure 3: Motion primitives of the legs

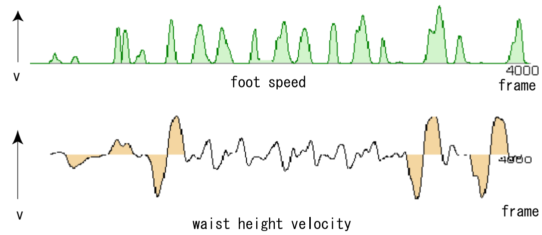

Figure 4: Graph of foot speed and waist height velocity

Primitive	Parameters
STAND	- standing period - waist height
SQUAT	- period (medium time, final time) - the lowest waist height at the medium time
STEP	- which is a swing / support foot ? - period (medium time, final time) - position and orientation of the swing foot at the medium time and the final time - waist orientation at the final time

Table 1: Parameters of Leg Primitives

these times: Position and orientation of a swing foot and the waist. Those properties are described as relative value from the support foot. The primitive does not require the initial state so that it can be adapted to any initial poses.

SQUAT represents one squatting motion, down the waist and up it again. As well as STEP, SQUAT has medium time at which the waist takes the lowest position, and height of the waist at that time.

The definitions of three primitives are summarized in table 3.1.

Note that values of the primitive parameter which concerns with the length must be normalized into some standard human model. The normalization enables unified description and adaptation to various robots.

3.2. Extraction of Primitives

To extract STEP primitives, the speed graph of a foot (Fig. 4) is analyzed. This graph is basically a sequence of bell shaped curves. During one unit of the curves, the foot moves and stops. This movement is regarded as stepping motion. Hence segments of STEP primitive are extracted as a unit of the curve. Medium time is extracted by finding the highest position of a swing foot in the segment. Note that small sliding movements of a support foot appear as small curves in the graph so that those curves should be eliminated. Sins a step does not always lift up a swing foot, height of a foot is useless to judge whether a foot is supporting or swinging.

After each STEP segment is extracted, states at the key times are extracted from captured foot markers as relative values from a support foot.

To extract SQUAT primitives, a velocity graph of waist height (Fig. 4) is analyzed. In this graph, squatting action appears as a set of a concave curve and a convex curve, that is the movement to down the waist and up it again. The extraction process has to find this set of curves. As well as STEP, a small area which is nothing but a small swinging should be eliminated. Waist height at the lowest position is extracted as a medium state.

The segment of STAND primitive corresponds to the frames where the speed keeps approximately zero for a certain period in both foot speed and and waist height velocity.

Figure 5 shows an extracted primitive sequence in Jongara-Bushi.

4. Generation of Leg Motion for a Robot

Leg motion for a robot is generated from a primitive sequence, which is extracted from a original dance motion through the recognition process described in above section.

First, position and orientation trajectories of feet are generated. Then, initial joint angle trajectories are calculated from the feet trajectories by inverse kinematics of leg. After leg motion and arm motion are integrated, the waist trajectory is modified to satisfy the dynamic balance.

4.1. Generating Initial Motion

For each primitive in the acquired sequence, feet trajectories which represent an action of the primitive is

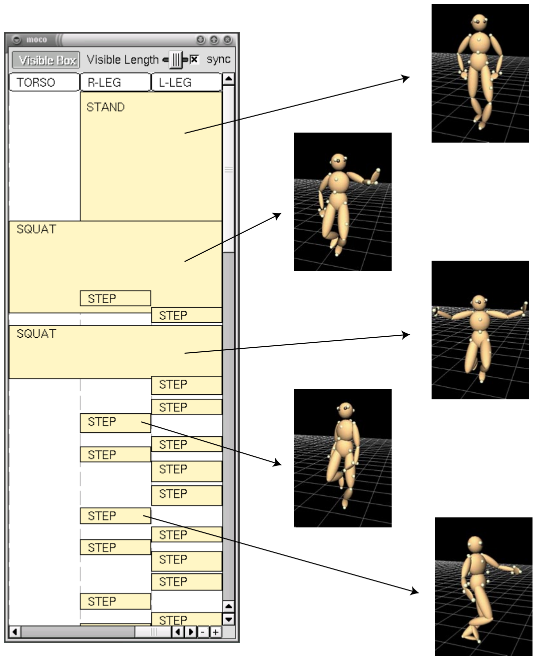

Figure 5: Extracted primitive sequence and poses in some primitives

generated. To be precise, the values of foot trajectory are the position of the ankle joint and the orientation of the sole. These values are expressed on the waist coordinate for inverse kinematics of leg.

For STEP primitive, the trajectories are basically created by interpolating initial, medium, and final states of position and orientation. The initial state is the state just before entering the new primitive motion. The medium state and the final state are determined from the parameters of STEP primitive.

There is the case that the medium state and the final state are modified according to the robot constraints. The modification is caused by contact condition, the constraint of movable range, and self collision.

Although human can contact a sole freely with a floor, a present robot must contact a sole to lie flat against the floor. Otherwise, a foot cannot support the body stably. Thus the final foot orientation in acquired STEP primitives must be constrained to be horizontal, and the final position must be level with the floor.

Movable ranges of robot legs is usually narrower than that of human. If a state of a primitive is beyond the ranges, it must be restricted within the ranges.

Since a robot like HRP-1S has fat legs, self collision of legs easily occurs. Compared with human, self collision is significant problem for a robot. Collisions must be detected and eliminated in generation process.

For SQUAT primitive and STAND primitive, feet trajectories are generated by the similar process as STEP.

After feet trajectories are generated, joint angle trajectories are calculated by inverse kinematics.

4.2. Dynamic Balance

A target robot has the ability to take poses in the motion generated through the above process. However, if the actual robot tries to perform the motion on a floor the robot must be unable to keep balance so that it falls down.

Leg motion is generated under the assume that all the area of a foot sole contacts with the floor when it is supporting the body. In other words, a sole does not rotate during that time. In terms of dynamics, this assume is satisfied when the point at which the moment to the robot body is zero exists in the area of the sole surface. In this time, the sole does not rotate. The point is called ’zero moment point (ZMP)’ and the area is called *supporting area*. If a robot is supported by both feet, supporting area corresponds to the convex area which consists of both soles.

Given the physical model of a robot, a trajectory of ZMP can be calculated from motion data for the robot, under the assume that the supporting area is infinite. If zmp moves out of an actual supporting area, the motion is impossible to perform because the actual motion must imply rotation of the supporting sole at that time, so that the sole moves away from the ground and the robot falls down.

In this study, a desired ZMP trajectory which is always in the supporting area is prepared first. Then the motion is modified to realize the trajectory.

4.3. Desired ZMP Trajectory

The condition that ZMP is inside a supporting area must be satisfied, but it is not sufficient condition for actual stability. If a supporting area remains on one state, ZMP should remain a stable point in the supporting area, but supporting state changes with steps. Stability of motion depends on a ZMP trajectory with state transitions. In this study, we applied the following criteria.

- In STEP period, ZMP must locate at center of a supporting sole.
- In STAND period, ZMP moves from a previous position to the next supporting position by third order polynomial equation. Initial velocity and

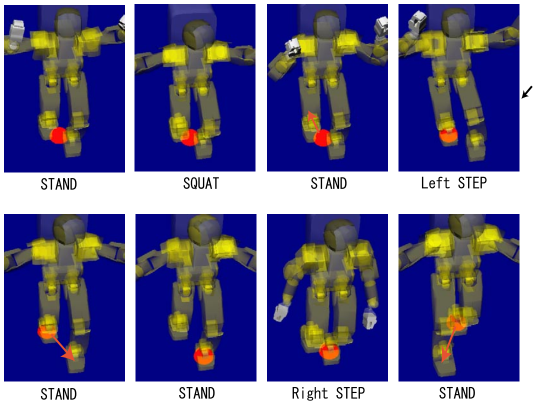

Figure 6: A motion with support state transition. A markers on the feet shows ZMP.

accelerations and final ones are kept zero.

- If period of STAND is long, transition is separated into three steps: (1) from a previous position to center of supporting area, (2) stay there and (3) move to the next supporting position.

- If period of STAND state is short, ZMP movement speeds up and robot motion becomes unstable. Adequate transition time is required for stable motion. In this case, ZMP movements is expanded so that it starts in the previous state and extends into the next state. Acceleration and deceleration of ZMP is done in those states.

Figure 6 shows a sequence of support state and ZMP trajectory.

4.4. How to realize desired ZMP

Given a desired ZMP trajectory, a motion must be modified to realize it. Nishiwaki et al. [**?**] proposed a method to solve this problem On discrete system, supposing all the segments are restricted to be translated horizontally in the same distance, the following equation is acquired.

$$x^e_{zmp}(t_i) = \frac{-hx^e(t_{i+1}) + (2h + g\Delta t^2)x^e(t_i) - hx^e(t_{i-1})}{g\Delta t^2}$$

where x^e_{zmp} is a difference between calculated ZMP and desired ZMP, x^e is a translation distance to realize desired ZMP t_i is time at frame i, h is height of center of mass, Δt is time per one frame. This equation is about x-axis, and similar equation applies to y-axis.

This equation is expressed by information of consecutive three frames. This type of equations are solved as tridiagonal simultaneous linear equations.

This method cannot figure out a result which completely follows the desired ZMP trajectory in one calculation, because the constraint that all the segments translate parallel in the same distance is actually impossible. However, by iterating calculation, a converged result is acquired.

5. Results

In recognition, primitive sequences are successfully extracted as Fig.5. In this section, we examine validity of generated robot motion.

5.1. Appearance of Generated Motion

Figure 7 shows a captured dance motion and a generated robot motion for HRP-1S in Jongara-Bushi. Animation here is not dynamics simulation, but just replay of motion data.

It seems that the result can be regarded as good imitation of the human performance. The most remarkable difference is trajectories of a whole body. Since movable range of the robot legs is restricted, the robot cannot always follow a rapid turn or a wide step in the original motion.

Although a method to evaluate similarity or skill of performances is necessary, we currently have no proper criteria. Evaluation such as distance comparison of trajectories of some body parts makes no sense because body type is different between a robot and a human, and this applies among skilled human dancers.

5.2. Feasibility in Dynamics

Compared with appearance, validity of dynamics is clearly evaluated by whether a robot accomplishes a performance or falls down. First, a motion is tested on the dynamics simulator. If the robot can stably perform a dance from beginning to end on simulation, a performance by an actual robot is experimented.

Before simulation, ZMP behavior should be checked. In a initial motion without balance control, ZMP movement is noisy and it cannot remain in a supporting area. After balance modification is applied, ZMP movement becomes quite stable. However, when the calculation to follow desired ZMP is iterated, there is a case that erratic dropping motions appear. The balance modification of each frame refers to only neighbor two frames, unreasonable initial waist trajectory

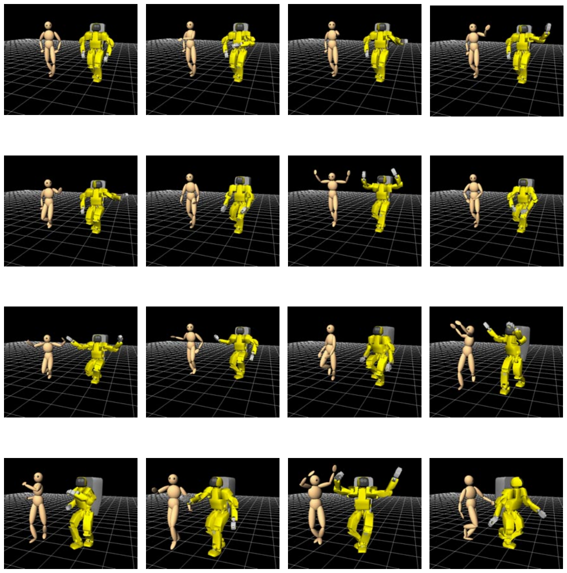

Figure 7: Original motion and generated robot motion

may falls into local erratic dropping. Initial motion generation should be improved to solve this problem.

Then dynamics simulation is tested with the controller of HRP-1S [5]. Although the controller has facility to keep balance against small disturbance, initial motion cannot accomplish a performance. After balance modification is applied, the robot successfully performs a whole dance without falling down as Fig. 8. However, a support foot slides and motion becomes unstable when the robot widely rotates the waist to turn. Balance modification does not consider horizontal (yaw) element of moment, which is not concerned with ZMP. Balance modification should be improved to solve this behavior.

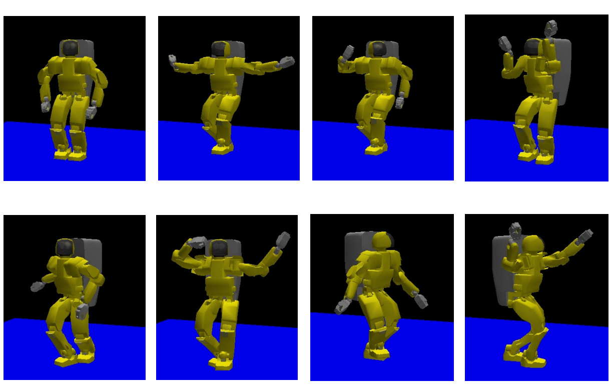

Figure 8: Dynamics simulation with leg steps

6. Conclusion and Future Work

This paper proposed primitives of leg actions for dance imitation for a robot On the basis of the primitives, human dance performance is recognized as simple symbolic description, and feasible robot motion is generated from the recognition result. This paper describes a method to create a desired ZMP trajectory for balance modification. Imitative dance performance with leg actions by a humanoid robot is realized on dynamics simulation.

We are going to experiment performances on an actual robot. The generation method should be improved to acquire more stable performance. Current problems are local erratic dropping and yaw element in balance modification. Additionally, evaluation criteria of performance skill should be defined and a robot should acquire high-skilled performances on the criteria.

References

[1] Tetsunari Inamura, Iwaki Toshima, and Yoshihiko Nakamura. Acquisition and embodiment of motion elements in closed mimesis loop. *Proceedings of International Conference on Robotics and Automation*, 2002.

[2] Fumio Kanehiro. Virtual humanoid robot platform to develop controllers of real humanoid robots without porting. *Proceedings of International Conference on Intelligent Robots and Systems*, 2001.

[3] Atsushi Nakazawa, Shinichiro Nakaoka, and Katsushi Ikeuchi. Imitating human dance motions through motion structure analysis. *Proceedings of International Conference on Intelligent Robots and Systems*, 2002.

[4] Nancy S. Pollard, Jessica K.Hodgins, Marcia J. Riley, and Christopher G. Atkeson. Adapting human motion for the control of a humanoid robot. *Proceedings of International Conference on Robotics and Automation*, 2002.

[5] Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Shuji Kajita, and Hirohisa Hirukawa. A honda humanoid robot controlled by aist software. *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2001.

[6] Takashi Yukawa, Takaaki Kaiga, Kazuo Nagase, and Hideo Tamamoto. Human motion description system using buyo-fu (in japanese). *Information Processing Society of Japan*, 2000.