Exploring Creativity through Humanoids and Dance

Robert Ellenberg
Drexel University
rwe24@drexel.edu

David Grunberg
Drexel University
dkg34@drexel.edu

Youngmoo Kim
Drexel University
ykim@drexel.edu

Paul Oh
Drexel University
paul@coe.drexel.edu

Abstract - Programming a humanoid robot to dance to live music is a complex task requiring contributions from multiple disciplines. A vocabulary of intricate limb motions must be designed to be stable and stylistically consistent. To produce coordinate these movements with music, the robot must be able to detect the appropriate beats within audio. One approach is a perceptual model of hearing, which can accurately determine beat locations, even for music without strong rhythmic content. The chosen implementation used an IIR feedback comb filter bank to determine tempo of an audio stream in real-time. A library of gestures was designed for a small humanoid robot to represent various dance motions. The software coordinates dance by choosing a random series of gestures, ensuring that a sequence won't upset the robot's balance. While playing the audio, the software initiates gestures that syncronize with detected beats. The final implementation allows user choreography and generative dance in a graphical user interface.

Keywords - Humanoid, Creativity, Robot, Dance, Rhythm

Fig. 1. Interacting with a humanoid robot

1. Introduction

In recent years, robots have taken the public stage to dance, perform musical recitals and even conduct orchestras. For instance in 2007, ASIMO was programmed to work with other humanoids cooperatively[1], and in June of 2008, could distinguish between three distinct voices[2]. In 2007, Toyota unveiled robotic violinists and trumpet players[3]. In 2008, ASIMO conducted the Detroit Symphony Orchestra [4]. These are impressive demonstrations that delight the public and have helped to spark large interest in robotics. However, many of these lack creativity. All of the performances were choreographed, or followed a strictly programmed logic. They demonstrate specific abilities very well, but fall short of achieving creativity.

The authors of this paper have a broader desire to make humanoids more human. In particular, they are interested in dance. The humanoids discussed previously are the state of the art in mechatronics, yet require careful programming and rehearsal to perform the simplest of tasks. If a robot could perform such a quintessentially human act, however, it would be much easier to identify with.

One example is the Keepon[5], a small robot that performs simple motions randomly to music. Its design is purposefully minimal, using only two yellow spheres as a "body", and only rudimentary nose and eyes for a face. The purpose of this robot is to engage children by dancing and interacting with them. Unfortunately, it does not perform beat tracking or demonstrate creativity, but the positive response shows a clear demand (Fig. 1) for more interaction and lifelike behavior in robots.

Dance is particularly challenging because it is often a real-time and creative expression of how one interprets a piece of music. Such expression can range from simple foot tapping to more complex motions like pirouettes and body leaps. The expression is also governed by music, culture, language and history. The complexity and difficulty of dance allows a rich vocabulary to express creativity that humans can recognize and understand. Like many forms of stage performance, dance embeds a mix of perception and cognition that is ultimately expressed by physical motion. Producing creative dance in a robot requires a fundamental understanding of all these subtleties[1].

Parts of the problem have already been examined. Deducing rhythm from music is fundamental to dance, and has already been demonstrated by the drum-playing robot Haile[6]. Using human-produced samples as a basis, Haile can improvise patterns and play along with human drummers. Its algorithm produces improvisations that vary depending on a desired

[1]This work was supported in part by the National Science Foundation Partnership for International Research and Education (PIRE) award OISE0730206.

parameter such as rhythmic density.

Research on virtual choreography looks at another aspect of this problem: how to represent dance in a consistent and specific notation[7]. The team developed virtual dancers that could accurately mimic human motions. This representation is crucial for sophisticated and human-like motion, though the work did not extend to creative production of dance routines.

An adult-sized humanoid would be an ideal platform to apply artificial intelligence (AI) algorithms and study creativity. Unlike Keepon, humanoids provide the anthropomorphism for people to recognize creativity. Simulations and virtual characters provide more flexibility for programming. However, the underlying physics and non-linear dynamics are not often well-defined in such simulations.

The authors are interested in applying AI-based algorithms on actual humanoids. The recent proliferation of small-sized humanoids (Fig. 2) provide a viable surrogate for such algorithms and study creativity. Section 2 describes how the platform was chosen. Sections 3 and 4 describes the methods and algorithms used. Section 5 explains the ramifications of the gathered data and the experiments. Section 6 discusses future work.

2. Experimental Infrastructure

Many small-sized humanoids have recently become available to to the U.S. public in recent years. These humanoids are usually between 12 to 20-inches tall, feature 10 to 20 degrees-of-freedom and are between 1000 to 2000 USD. These humanoids are often constructed from off-the-shelf hobby servos and are relatively easy to maintain. To be a viable platform for the authors' research, the following technical requirements were identified:

- Enough joint DOF's to mimic human motion. Naturally, a robot that lacks range of motion compared to a human will appear to be less capable as a dancer to a human observer.
- A mature and well documented programing system, which minimizes time spent coding gestures, and increases the time available for high-level coding.
- Wireless communication, which lets an external processor perform computationally intensive tasks.
- Simple mechanical construction, which minimizes downtime and maintenance.
- Sensors such as rate gyros and accelerometers for feedback control (Fig. 2).

Robots that met the design criteria of simplicity in mechanical construction were all pro-sumer robot kits. Three leading candidates emerged: the Bioloid from Robotis, Kondo's KHR-2, and the Robonova-1 from Hitec. While research has been conducted with the Bioloid [8] and the Robonova-1[9], none of the platforms has gained widespread use for humanoid research. At the start of the project, only Hitec's RoboBasic system had full documentation, an integrated development environment, and interactive posing/motor control.

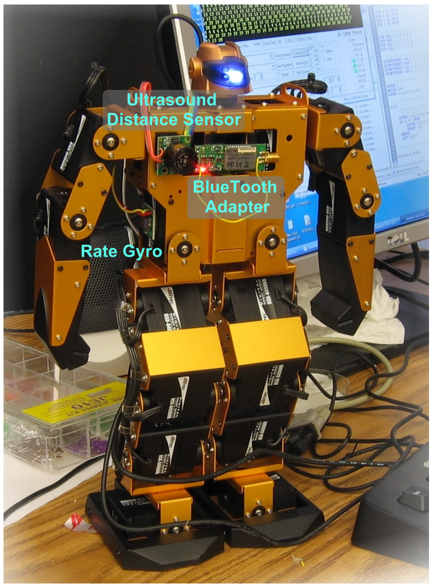

Fig. 2. Robonova miniature humanoid

2.1. *Robonova Platform*

An important advantage of the Robonova is its well-developed RoboBasic language and development environment. The low-level software can interpolate between any two positions, making motion programming very quick and easy (Fig. 3). The ability to pose the robot under power and capture its state as a frame allowed us to rapidly motions sequences. The hardware used in the experiments is Hitec's Robonova-1), chosen for its cost, simplicity, and software support. It is constructed of analog/digital servos joined with simple stamped brackets connecting the joints. The simple design allows easy maintenance and repair.

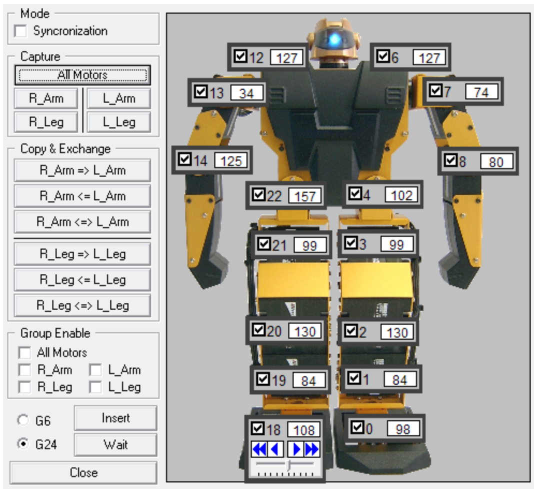

Fig. 3. Interface for posing servos to produce motion frames

The Robonova's CR3024 microcontroller runs at 7.81 MHz, with an onboard interpreter computing approximately 1200 instructions/sec. The basic code has limited features and arithmetic ability, but was sophisticated enough to read commands from a serial port and produce motion.

A laptop PC was used to perform the processor-intensive calculations of music beat tracking and gesture command. Development of audio-processing algorithms is much simpler and faster in an environment like MATLAB, where toolboxes and libraries exist to alleviate much of the burden. As such, the microcontroller had only to interpret gesture commands and control limb servos accordingly.

A basic library of about 20 gestures was programmed for the robot. An important simplifying assumption is that a fixed library of gestures will not completely restrict potential creativity. Much like how a good writer can be creative even with a fixed selection of words and grammar, the robot should still be able to express a creative arrangement of these gestures.

Arm gestures (Table I) were selected to avoid motion overlap, eliminating the need for collision control. While the Robonova's 180° elbow motion allows it to perform identical front and rear gestures, the elbows were limited to an anthropomorphic 90°.

TABLE I
ARM GESTURES

Gesture	Motion
Swing forward	Pivot arm forward at shoulder joint
Swing backward	Pivot arm backward at shoulder
Swing sideways	Lift arm straight out from body
Salute	Bring grip to chest
Sweep Across	Lift arm out, swing inwards to body
Tap	Quickly tap grip to knee

The steps chosen as leg gestures (Table II) were chosen to have a clear start and finish, while minimizing motion between states. Besides having longer execution time, large leg gestures such as Rockette-style kicks would upset the balance of the robot. Closed loop balance control could not be used effectively, so the gestures had to be carefully crafted to maintain stability. For any one leg gesture to occur, both legs had to move in order to shift the balance point over the supporting leg.

TABLE II
LEG GESTURES

Gesture	Motion
Step forward	Shift body to single support, lean forward, step foot
Step backward	Same to the rear
Step sideways	Same to the side
Foot Tap	Lift and tap foot without disturbing COM position

Maintaining stability also required a careful tuning of the balance point shift because the linear interpolation between double and single support poses may require moments at the ankle that the foot cannot produce. By reducing the speed of the transition, the accelerations required can be minimized. Due to the relatively large feet, rapid motions of the free leg can be made without upsetting the balance of the Robonova.

Compliance of ankle joints led to wobble due to momentum of the arms. To reduce this wobble, we reduced the lift and step length for each foot to reduce single support time. The fast motion required had a tendency to make a jarring foot landing, however, since the supporting ankle's position was uncertain.

Originally, the final position of leg motions resembled Fig. 4-a. This meant that after a step, the body's angular momentum was opposed by the landing foot's impact. If the foot landed flat as shown, however, the leg was fairly straight, which transmitted much of the load through the leg. Besides being unpleasantly loud, the impacts increased the settling time unacceptably. Lacking the ability to perform realtime feedback control, we instead altered the dynamics of the landing. By balancing on the static foot, the upper body was kept reasonably still during a step. Keeping the landing foot slightly lifted and tilted allowed the foot to gradually contact the ground (Fig. 4-b), improving the settling time of the The leg's momentum could be partially opposed by the flex of the ankle, which considerably improved the settling time of the leg gestures.

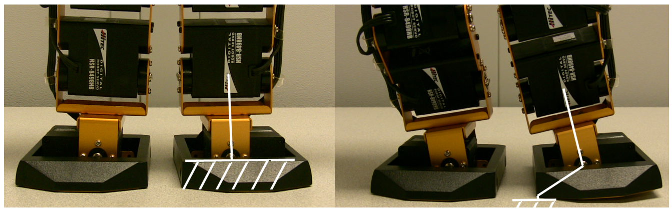

Fig. 4. a) Equal force on each foot (double support). b) Leg extended (single support)

3. COMPUTER GESTURE PRODUCTION

3.1. *Beat tracking*

The robot must also be able to extract information from music to compose a dance routine (Fig. 5). The simplest approach is to perform beat tracking to determine the tempo of the song, so that gestures execute with proper rhythm. Knowing the tempo and beat locations of the song, a series of gestures is then performed in time to this pace.

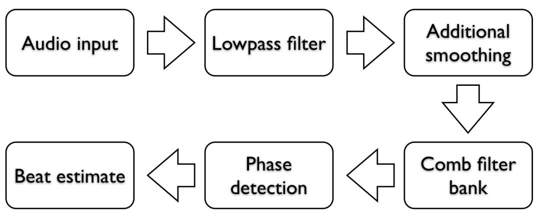

Fig. 5. Processing steps in beat tracker

A model based on human auditory perception is used to determine beat information within the audio signal, using the methods described in [10]. A comb filter (Fig. 6) delays a signal and combines the delayed version with itself and is generally implemented as an infinite impulse response filter.

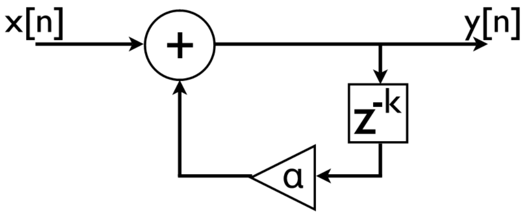

Fig. 6. Block diagram for a single comb filter

This configuration produces resonance when the filter delay matches the period of the input signal, as shown in Fig. 7. For beat detection, a bank of comb filters with varying delays allows a range of tempos to be identified. Figure 8 is a tempogram showing a typical filter output, with the peak energy corresponding to the beat tracker's estimate of current tempo. In each frame of audio (∼50 msec duration), the algorithm identifies the peak output energy, which gives the tempo and beat time predictions.

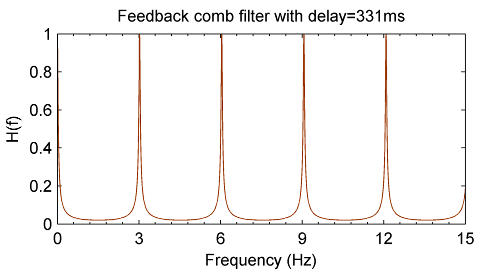

Fig. 7. Comb filter impulse response

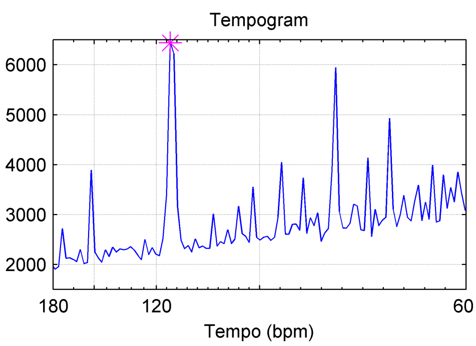

Fig. 8. Comb filter output (signal energy vs. frequency)

After an initializiation time to accumulate beat information, the system is able to predict whether any beats are anticipated within 0.55 seconds of the current analysis frame. This provides the system with sufficient lead time to produce any of the gestures within the movement vocabulary. Since the execution time of each gesture differs, the software must transmit the gesture command at the correct amount of time before the beat in order to maintain synchrony with the music. Several optimizations increased computational efficiency so that the beat tracker could operate in real-time. For example, the output of a single low-pass filter channel was sufficient to track strong beats. Since most percussion sounds peak at less than 400Hz, higher frequency melody and sound effects can be removed without loss of accuracy. Because the music database currently incorporates only popular music with strong, clear percussion, it is an acceptable tradeoff.

3.2. Graphical User Interface

A Graphical User Interface (Fig. 9) was designed to let users easily choreograph gesture sequences and demonstrate the robot's capabilities. User control allows the robot to become a basic choreography tool. Given enough control of dance motions, a humanoid robot could become a rapid-prototyping system for dance routines.

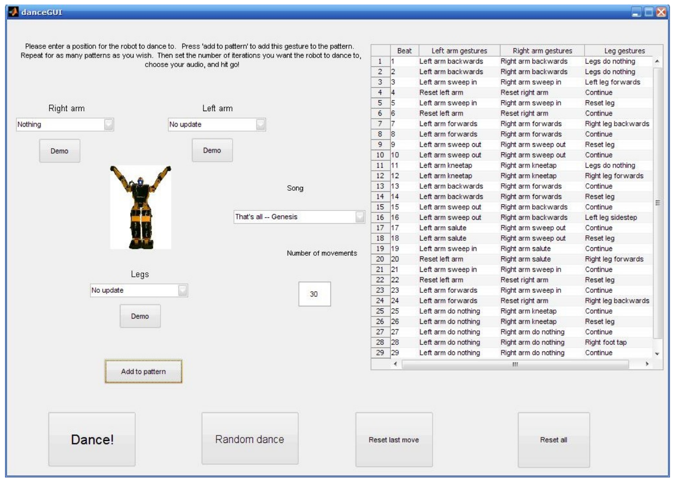

Fig. 9. Graphical User Interface for Choreography

The GUI also enables users to explore the capabilities and limits of the robot in an easy and intuitive manner. With only a few mouse clicks, the speed and balancing abilities of the robot can be displayed. This will allow the user to be educated about the robot more quickly than he or she could by reading the manual or laboriously coding a sequence of gestures for direct execution.

The prototype gestures are conveniently listed in a table; long gestures that require multiple beats are identified as such for ease of use. The user can demonstrate individual gestures with the 'demo' buttons. Another button allows the user to add the current group of gestures to the table. Finally, the user chooses the song and the play time, and begins the sequence. The complete MATLAB code and GUI is available for further reference on the project website[11].

4. Physical Gesture Production

To receive and interpret gestures, the main program loop was designed with three stages: receive commands, initiate motions, and reset state/command variables. This command cycle is updated separately for the right arm, left arm and legs. After the last movement step, the command variable is reset, and a new gesture can begin. Gesture commands to the Robonova are transmitted in groups of 3 ASCII characters (Tab. III).

TABLE III
ROBONOVA GESTURE COMMANDS

Limb	Valid Commands	Total Possible
Right Arm	'A'-'M'	13
Left Arm	'N'-'Z'	13
Legs	'a'-'z'	26

These gestures consist of linear point-to-point interpolations between stored poses. The simplest gestures require only one pose, or a cycle of 2 steps to return to the initial position. Leg gestures require shifting balance to one foot, lifting the opposite foot, and placing the foot, for a total of 3 poses. For further information, the fully commented Basic code can be found at the project website[11].

5. RESULTS

Measured loop refresh rate is plotted in Fig. 10. The peaks show the average time to complete one program loop when no commands are issued (blue) vs. a full set of 3 gestures (green). At two standard deviations, the loop times were 0.22s and .48s, respectively. The maximum theoretical dance tempo to update at every beat is approximately 125 bpm. In practice, however, the .1s-.2s minimum loop time means that timing resolution is coarse. To minimize delays, if the latency difference between neighboring gestures was less than .2s, the transmission time was averaged, and the commands sent simultaneously.

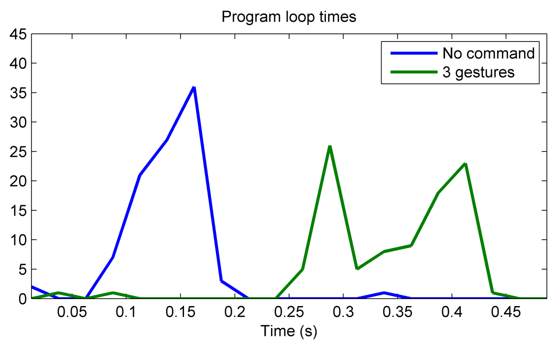

Fig. 10. Microcontroller loop time, null commands

In the worst case, new gesture commands would arrive before a slow gesture could finish, causing the overlapping command to be ignored. If that lost gesture was a return command, then future commands would not start from the default pose. Without balance feedback, the resulting motion was unpredictable. If the robot could recover its balance, then the faulty gesture did not detract from the performance. If a leg gesture was lost, however, the robot would often topple. Luckily, the solution to this problem is simply to use faster hardware.

6. FUTURE WORK

6.1. Hardware Improvements

A strong candidate platform for future work is the Hubo robot from KAIST. It has 42 degrees of freedom, inertia and contact force sensing, and two onboard x86 computers. Limb articulation, balance control and walking gait have already been demonstrated [12], which makes it an ideal platform to develop complex dance motions. As the beat-tracking algorithms improve, more information will be available to the robot. Unlike the fixed gestures of the Robonova, the Hubo could vary a gesture's speed, trajectory, and fluidity.

6.2. State-driven Dance

Sophisticate motion control moves beyond the current paradigm of fixed gestures. Considering locations of the hands and feet as separate components of a simplified state, dance moves become state transitions. With the simplest motion being an interpolation between successive states, some transitions will require intermediate states to maintain stability (such as switching supporting legs, or walking forward). An example of creativity at this level would be to order these pathways in a novel way.

More sophisticated trajectory planning, collision detection, and feedback control will even allow multiple paths between various states. A choice of a particular path could me based on qualities of the music. Since smooth human dancing requires performing motions that follow easily from previous positions, such qualities could be expressed in terms of motion properties and energy expenditure. An algorithm to "optimize" a path could consider different motion properties as its costs.

Furthermore, different trajectories between states will be necessary to add variety and subtlety to the motions. It has been shown that certain movements can indicate certain emotions to an audience[13]. This means that it is not enough for the robot to go from point A to point B, but it must do so in a manner suitable to the particular music it is dancing to. This means that the robot must learn various dance 'styles' and 'moods.'

A useful system to distill dance motions into a fundamental description is Labanotation. Developed around 1929[14] by Rudolph Laban, the system represents space and motion in a formal notation. It is commonly used for choreography to map dances[15]. Laban further specified several parameters to describe each movement within that dance space point to another in a 'happy' or 'angry' manner. Attempts to use this notation to code virtual dance [7] have shown promise, which indicates that a humanoid dancer could take advantage of the system as well.

Laban's components of a gestures dynamics are space, time, weight, and flow[15]. *Space* measures how direct a motion moves from one point to another in the dance space. One measure of directness could be the ratio of displacement to path length. Naturally, a more convoluted path to arrive at the same point in space would be more indirect, and thus have a lower directness factor. *Time* measures the suddenness and the speed of the motion. A simple time scaling on a gesture would allow the robot to adjust the execution time, without affecting the motion in space. *Weight* measures how much apparent strength is used in the motion. The challenge with this factor is essentially to quantify the difference between a light gesture (waving a feather) and a heavy one(pushing a weight) in purely kinematic terms. Like a human limb, however, an overactuated robotic arm allows optimization of the trajectory for joint torque or space efficiency [16], [17]. To produce a light or heavy motion, an appropriate virtual load could be applied to a dynamic model, and the resulting optimized trajectory to move the load would then become the dance gesture. *Flow* measures the freedom or restrictiveness in a movement, which is the most difficult to quantify, and will be left for future work.

6.3. Advanced Music Analysis

Currently, the beat tracker is designed to detect the prominent, regular beats in popular music. The system also tends to overweight higher tempos, which can be corrected using a re-weighting of the comb filter outputs as suggested in [18].

In order to incorporate the emotions expressed by the music through the robot's movements, the beat tracker must extract information beyond the tempo:

- Rhythms present in the music
- Scale/Key of the harmony and melody
- Density and variation of the musical phrases
- Choice of instruments and musical timbre

The robot will have to incorporate these features and others into its dancing. By taking advantage of music theory [15] that explains how these factors (and others) relate to mode and emotional state, the robot will be able to evoke these emotions in a fundamentally novel way.

Ultimately, since dance has such an emotional foundation [13], producing dances which are not only mechanically creative, but express an emotional message to an audience, is a strong test of a humanoid robot's sophistication. To do this would require a model of human emotions on which to build this message. To transmit such dances effectively, it would also need a theory of its audience, and be able to "understand" how to translate its message into a human language of dance.

6.4. Expert Choreography

Professional choreographers could aid the project by determining:

- Gestures mimicking those of humans
- Complete, fluid gesture sequences
- The quality of produced gestures

The choreographers could also judge the system by examining the same gestures when performed by the Robonova robot and by a human. This would show which aspects of the robot closely approximate human gestures and which could be further improved.

7. Conclusion

While the music interpretation and gesture planning established so far do not yet show creativity, the foundation of the project has been established. With a reliable means to produce rhythmic gestures of varying execution times, with associated error checking and structuring, more advanced planning can now be implemented independent from the low-level code. The division of processing between the high and low levels makes porting to other platforms quick and easy.

This bibliography was generated on February 15, 2009

References

[1] Honda, "Honda develops intelligence technologies enabling multiple asimo robots to work together in coordination." Press Release, December 2007.

[2] Honda, "Robot asimo can understand three voices at once." Press Release, June 2008.

[3] L. A. Times, "Toyota shows range with robot violinist." LA Times Business Pages, December 2007.

[4] Honda, "Hondas asimo robot to conduct the detroit symphony orchestra." Press Release, April 2008.

[5] M. P. Michalowski, S. Sabanovic, and H. Kozima., "A dancing robot for rhythmic social interaction," in *Proceedings of the 2nd Annual Conference on Human-Robot Interaction (HRI 2007)*, vol. 103, pp. 89 – 96, ACM, March 2007.

[6] G. Weinberg and S. Driscoll, "The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design," in *HRI '07: Proceedings of the ACM/IEEE international conference on Human-robot interaction*, (New York, NY, USA), pp. 97–104, ACM, 2007.

[7] K. Nahrstedt, R. Bajcsy, L. Wymore, R. Sheppard, and K. Mezur, "Computational model of human creativity in dance choreography," in *AAAI 2008 Spring Symposium*, 2008.

[8] J. C. Wolf, P. Hall, P. Robinson, and P. Culverhouse, "Bioloid based humanoid soccer robot design," in *the Second Workshop on Humanoid Soccer Robots at the 2007 IEEE-RAS International Conference on Humanoid Robots*, (Pittsburgh,USA), November 2007.

[9] I.-W. Park, Y.-D. Kim, B.-J. Lee, J.-K. Yoo, and J.-H. Kim, "Generating performance motions of humanoid robot for entertainment," *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pp. 950–955, Aug. 2007.

[10] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, pp. 588–601, October 1998.

[11] R. Ellenberg, "Robonova-1 dance code base." http://dasl.mem.drexel.edu/robertellenberg/robonova/RN1dancecode.zip.

[12] I.-W. Park, J.-Y. Kim, and J.-H. Oh, "Online biped walking pattern generation for humanoid robot khr-3(kaist humanoid robot - 3: Hubo)," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 398–403, Dec. 2006.

[13] A. P. Atkinson, M. L. Tunstall, and W. H. Dittrich, "Evidence for distinct contributions of form and motion information to the recognition of emotions from body gestures," *Cognition*, vol. 104, pp. 59–72, July 2007.

[14] A. Camurri, S. Hashimoto, K. Suzuki, and R. Trocca, "Kansei analysis of dance performance," *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 4, pp. 327–332 vol.4, 1999.

[15] E. Dörr, "Rudolf von laban: The founding father of expressionist dance," *Dance Chronicle*, vol. 26, no. 1, pp. 1–29, 2003.

[16] J. Hollerbach and K. Suh, "Redundancy resolution of manipulators through torque optimization," *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, vol. 3, pp. 308–316, Aug 1987.

[17] S. R. Maliadi, K. P. Valavanis, and M. C. Mulder, "Minimum effort factor approach for sensor-based control of a four-joint 6-dof redundant manipulator," *Journal of Robotic Systems*, vol. 11, pp. 21–40, 1993.

[18] A. P. Klapuri, A. J. Eronen, and J. T. Astola, "Analysis of the meter of acoustic musical signals," *Audio, Speech and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]*, vol. 14, no. 1, pp. 342–355, 2006.