Dance Motion Control of a Humanoid Robot Based on Real-Time Tempo Tracking from Musical Audio Signals

Naoto Nakahara[1], Koji Miyazaki[1], Hajime Sakamoto[1], Takashi X. Fujisawa[1], Noriko Nagata[1], and Ryohei Nakatsu[2]

[1] Kwansei Gakuin University, School of Science and Technology
2-1 Gakuen, Sanda, 669-1337 Japan
Nakahara-N@kwansei.ac.jp
[2] National University of Singapore
21 Lower Kent Ridge Road, 119077 Singapore
idmdir@nus.edu.sg

Abstract. This paper proposes a system that controls and generates a humanoid robot's dance motion in real-time using the timing of beats in musical audio signals. The system tracks changes in tempo and calculate the integration value of a decibel by analyzing audio signals in real-time. It uses the information to add changes to the robot's dance motion. Beat intervals and the integration value of decibels are used to change the tempo and range of the robot's dance motion respectively. We propose a method to synchronize dance motion of robot with musical beat, changing the robot's dance motion interactively according to the input value.

Keywords: Robot, Dance Motion, Beat Tracking, Music Understanding, Human Computer Interaction.

1 Introduction

Music and dance have had a strong relationship since ancient times. People started to move their bodies as music was played. Dance was used as a way to express feelings and communicate with each other.

There is numerous research on retrieving information from musical audio signals, but few applied the information to contents such as humanoid robots. There is some research that applies musical information to robot motion, but those contents dealt with comparatively slow movements and the information was presented one-sided from robot to human. We think that a bi-directional information exchange is an important factor for human-robot interactions. It is important to synchronize robotic motion with music, utilizing faster motion speed, to be able to perform more complex movements.

We implemented a system that tracks the timing of beats from audio signals that are input by either .wav file format music files or a keyboard. The system can change and control tempo and range of robotic dance motion by using the information extracted from audio signals in real-time. As the music is played by the user, the robot

S. Natkin and J. Dupire (Eds.): ICEC 5709, pp. 36–47, 2009.

synchronizes its dance motion with music's tempo and beat. The user can control the tempo and range of motion of the robot by tapping keys at different speeds and strength.

2 Related Work

Research done by Goto [1], [2] is famous for implementing a real-time beat tracking system called "BTS." BTS can track the timing of beats from musical audio signals. This model utilizes a multi-agent structure, where each agent predicts the time period between beats (inter beat interval) and the time of next beat with different parameters. Since this model tracks the timing of beats in music that have a roughly constant tempo, it takes a long time or simply can not track tempo changes. Research done by Yoshii [3] proposes a method to synchronize a humanoid robot's (ASIMO [4]) steps with musical beats. The musical signal is inputted from a microphone on ASIMO's ear and extracts beat intervals from that signal. Then ASIMO tries to synchronize his steps to that music. The step interval is limited to between 1000 and 2000 ms, and can not synchronize fast moves to the music. The movement is limited to the basic movement of a human, which is a stamping motion and does not change. Shiratori researched a method that allows a dancing robot the ability to observe and imitate human dance performances, making the movement more natural [5]. They extracted a sequence of primitive motion acquired from motion data. A method to convert captured human motion to a feasible robot motion automatically was proposed. In addition, they proposed a method to generate new combinations of motion by extracting the timing of beats and the dynamics of music along the time axis and the mapping sequence of primitive motion. These calculations are done offline, not real-time.

Although research pertaining to synchronizing motion of humanoid robots and music has been done, the motion of the robot is still limited to slow and basic motions of humans, or does not work in real-time. An existent real-time beat tracking model from audio signals takes a long time or can not track tempo changes. In addition, input is only received from a music file where information is presented one-sided, from robot to human. To realize an interactive robot dancer, the system must synchronize the robot's comparatively fast dance motion to musical audio signals and change motions according to the input signal in real-time. To track tempo changes in music faster than existing real-time beat tracking models are required when synchronizing robotic dance motion to audio signals that are inputted by a user, for example from a keyboard.

We propose a method to synchronize robotic dance motion with musical audio signals and change the robot's motion interactively in real-time according to the input audio signal. A method to track tempo changes faster than a past model is also proposed.

3 Retrieving Tempo and Beat from Musical Audio Signal

At the musical audio signal analysis stage of our system, we implemented a beat-tracking model to track the basic timing of beats in music audio signals. Although

beat tracking models proposed so far dealt with songs that have a roughly constant tempo, we implemented an algorithm which can track tempo changes in audio signals faster than the existent model. The system runs on a cluster of a duel processor, 2.16GHz operation machine.

In the frequency analysis stage, musical audio signals are continuously transferred into spectrograms by applying FFT. In our implementation, the FFT size is 1024 samples, the shifting interval is 256 samples, and input signals are sampled at 22050Hz. Therefore, the spectrogram is obtained every 11.6msec, and frequency resolution is 21.53Hz.

3.1 Finding Onset-Components

An important clue to finding the beat timing is based on the general knowledge of music; that at most times the instruments are sounded at the timing of the beat. This is the basic view of the beat tracking algorithm. Algorithms based on this make different predictions to output the best prediction of the next beat interval. If the Formula.1 is fulfilled, Formula.3 is used to calculate the power of onsets on a specific frequency. If Formula.1 is not fulfilled, the power of the onset component is considered as zero. The value of *prevPower* is given by Formula.2. Fig.1 shows an image of an onset component. An onset component is a frequency component that is likely derived from an onset. *p(t, f)*is the power at time *t* and frequency bin *f*, and *d(t, f)* is a power of the onset component at *t* and *f*. Unit of time *t* is 11.6msec, which is time resolution of FFT and the unit of frequency *f* is 21.53Hz which is the frequency resolution of FFT in the present implement.

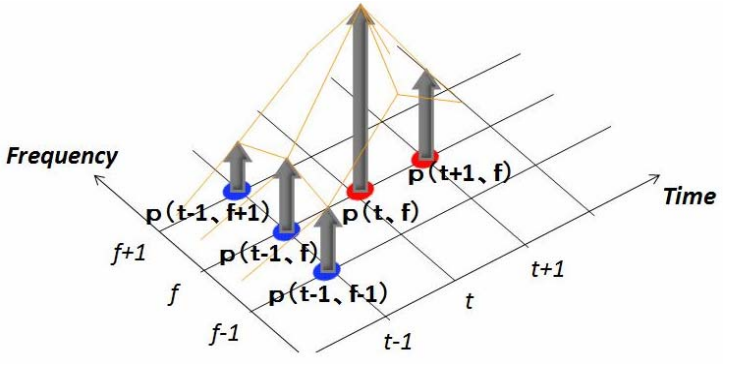

Fig. 1. Extracting onset component

$$\min(p(t,f), p(t+1,f)) > prevPower \tag{1}$$

$$prevPower = \max(p(t-1,f), p(t-1,f\pm1)) \tag{2}$$

$$d(t,f) = \max(p(t,f), p(t+1,f)) - prevPower \tag{3}$$

Onset components powers are classified into seven frequency ranges (0-125Hz, 125-250Hz, 250-500Hz, 0.5-1kHz, 1-2kHz, 2-4kHz, and 4-11kHz). All of the components powers are summed according to the classified range, described as seven onset component powers. These seven powers are recorded every 11.6msec along the time axis and described as the onset component's power vector. Each power vector is smoothed by the Savitzgy-Golay smoothing algorithm along the time axis **[6]**. The onset time is given by its time that gives the local peak of the onset power vector, and they are described as onset time vectors. Furthermore, seven onset time vectors are transferred into three types of vectors according to frequency focus type, which is low, middle, and all.

3.2 Auto-Correlation and Cross-Correlation

As a next step, onset time vectors are used as clues to predict IBI(inter beat interval), and the next beat time. Fig.2 shows an image of predicting IBI and next beat time. IBI is the inner beat interval which is the time between two beats. The IBI of the audio signal is obtained by calculating auto-correlation of onset time vectors. Next, beat times are predicted by calculating cross-correlation using onset time vectors. The system uses different parameters when calculating auto-correlation and cross-correlation to get different interpretations. Calculations of auto-correlation and cross-correlation on several different parameters and the calculation of interpretation are implemented as different agents in the system. Each agent has different parameters, such as auto-correlation window size and onset time vector frequency focus type. When the auto correlation window size becomes small, the agent's sensitivity to temporary changes will improve, but stability of the prediction will be low.

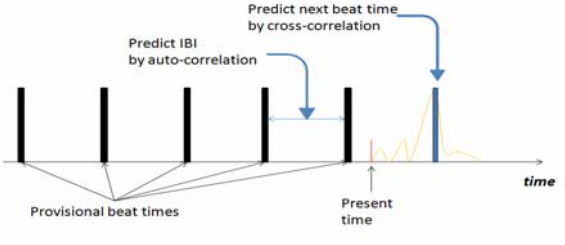

Fig. 2. Prediction of inter-beat interval and the next beat time by calculating auto-correlation and cross-correlation using onset vectors and provisional beat times

3.3 Integration of Agents

The difficult part of tracking tempo and beat is that all instruments are not always sounded at the timing of the beat and that appropriate parameters for calculating auto-correlation differs by music. To cope with the ambiguity of music, the system uses multi-agent predictions and evaluations. Each agent has functions that evaluate their own prediction. All agents are grouped according to their predicted next beat time. Then, evaluation values of each agent are summed in the group to obtain each group's evaluation value. The group that has the highest evaluation value is selected as the

primary group. The agent that has the highest evaluation value in that group is considered the most reliable agent. The system outputs the prediction of the most reliable agent as a final output.

4 Robotic Motion Control

At a robotic motion control stage of our system, robotic motion is generated and controlled. Basic functions and commands to control and synchronize with music are sent from a PC program to the robot's micro computer.

4.1 Humanoid Robot "Tai-chi"

Humanoid robots that are used in the system are "Tai-chi." Tai-chi is a humanoid robot that is made by Nirvana Technology. Tai-chi's height is 37cm and weight is 2.2kg. Specifications of Tai-chi are shown in Table.1

Fig. 3. Humanoid robot "Tai-chi"

Table 1. Specification of Tai-chi

Size/Weight	37cm/2.2kg
Degree of flexibility	21 (12/legs, 8/arms, 1/head)
CPU	SH2/7046 50MHz
Motor	KRS-2346ICS PDS-947FET
Battery	NiCad battery RCP-33 7.2V 1100mAh

4.2 Dance Key Pose Database

To preserve the basic dance motions of the robot, the system maintains key poses of dance motions. A robotic motion editor is used to produce key poses for the robot.

Image of the motion editor is shown in Fig.4. Tai-chi has 21 joint motors and the motion editor is used to preserve and control each angle of the motor which represents one key pose of the robot's dance motion. Users can easily change the angle of the motors with scrolling bars on the motion editor screen. Also, users can check what kind of key pose is generated with those motor angles by sending motor angles to the robot, or simply watching the computer graphics of the robot on the screen to simulate the robot's key poses and dance motion. The robot' dance motion key poses are described as 21motor angles; these values are saved as text files. Those text files are maintained in the system in the dance motion key pose database.

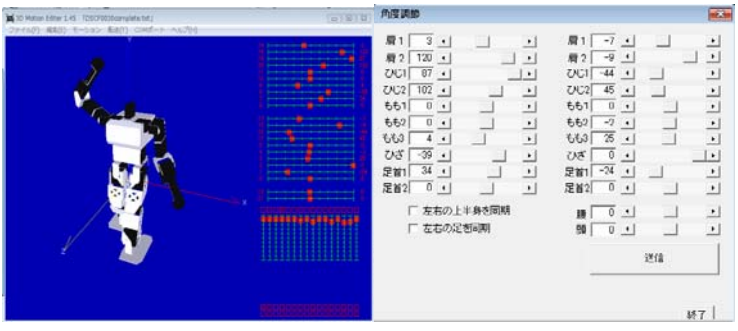

Fig. 4. Motion Editor of Tai-chi

4.3 Producing Dance Motion

An image of produced dance motions is shown in Fig.5. Dance motion is produced by using motor angles of key poses that are loaded from text files in the database. First, 21 motor angles that describe key pose are sent from the PC to the micro computer in the robot, the robot then takes that pose. Next, another 21 motor angles and specific times are sent to the micro computer. This time is the time between one key pose and another in msec. When these values are sent to the micro computer on the robot, it calculates and interpolates each angle of the motors to take the next pose in that specific time from the present key pose. By continuing this process, the robot's dance motion is generated and played. Also, by changing combinations of key poses and selecting different key poses every time, the system can generate many types of dance movements.

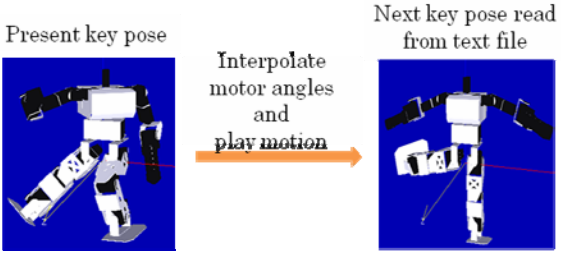

Fig. 5. Playing dance motion

5 Synchronizing Robotic Dance Motion with Music

An algorithm to synchronize robot's dance motion with music is shown in this section. This is realized by sending information from the musical audio analysis stage to the robotic dance motion control stage. The timing of a beat is known as the most fundamental and important factor of the music; it is also important when synchronizing dance motion with it.

5.1 Connection between the Musical Audio Signal Analysis Stage and the Robotic Motion Control Stage

In our present system, the musical audio analysis stage is implemented with Max-MSP; robot dance motion control stage is implemented with Visual C++. Both stages are in the same computer and are connected with TCP/IP. Control signals from the robot dance motion control stage in PC are sent to the robot through USB. Although in our present system both stages are in the same computer, it can be separated to different computers. For example, the music audio analysis stage in one computer can be connected with the robotic dance motion control stage in another computer using a standard TCP/IP network.

5.2 How to Synchronize Robotic Motion with Music

To be able to synchronize with the beat, the system should be able to determine the beat before it arrives. The timing of the next beat is predicted from audio signals at the musical audio signal analysis stage and it is transferred to robot dance motion control stage.

How to synchronize robotic motion with music is shown in Fig.6. When to send information and what kind of information to send is important in synchronizing robotic dance motion with music. When to send information from the musical audio analysis to the robotic motion control stage is decided before the next beat time comes. In the musical audio analysis stage, time between beats and the next beat time is predicted and calculated as final output. When the next predicted beat time comes, IBI (time period between adjacent beats) is sent from the musical audio analysis to robotic motion control stage. When the robotic motion control stage gets IBI from the present music, it selects the next key pose and reads the 21 motor angles from a text file. Also, the target time for the robot to transfer to next key pose from the present key pose is calculated. Target time is calculated by adding the IBI to the time when the information is transferred to the robotic motion control stage. Then, the 21 motor angles and the target time for the robot to transfer to the next pose is sent to the micro computer on the robot through a USB line. When the micro computer receive the information, it sets up the target time and next key pose motor angles, and calculates and interpolates each angle of motors as the time passes to transfer to that key pose at target time from the present pose. The robot's dance motion is generated and played by continuing this process.

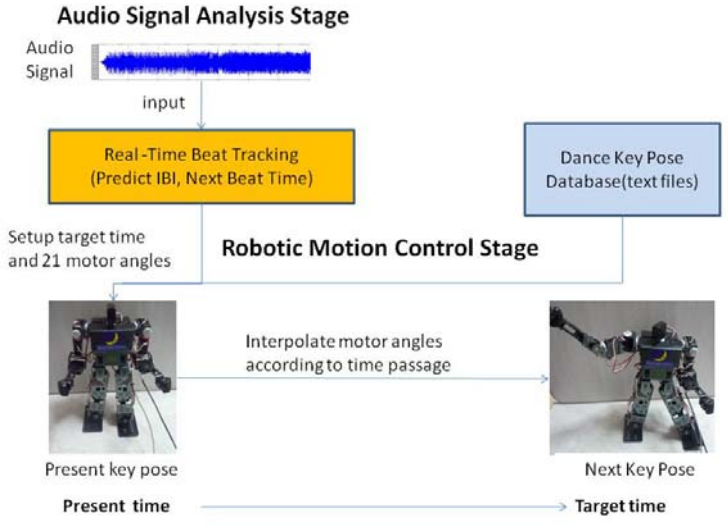

Fig. 6. Synchronizing robot's motion with music

6 Interaction with Robot Using Keyboard

The system works in two modes which are the "appreciation mode" and the "interactive mode." In the appreciation mode, .wav file format music files are inputted and the music has roughly constant tempo. Since the input signal has a roughly constant tempo, the system does not have to consider tempo changes. In the "interactive mode," audio signals that are obtained by the user's input from the MIDI keyboard are analyzed as input. Since the user might change the speed of key tapping, the system has to consider tempo changes and synchronize the robot's dance motion. Users can select the timbre of instruments such as piano and drums and play some chords on the keyboard or a tap of a key in a certain tempo. Although the system uses a MIDI keyboard as an input device, it does not use any MIDI information. MIDI signal is transferred to raw audio signal by connecting output jack and input jack on a PC with a line, and the signal from the input jack is analyzed as input of the interactive mode. The integration value of the decibel value of input signal is used to decide the strength of key tapping. Robotic motion is selected and changed according to the tempo and integration value of the decibel of the audio signal.

6.1 Tracking Musical Tempo Changes

As explained in section 3.2, the calculation of the auto-correlation and cross-correlation is used to predict the IBI of the music and next beat time. If the input is the musical audio signal from a music file, the tempo is roughly constant and does not have to consider tempo changes, so is important to maintain the prediction of the IBI as stably as possible. To make the stability higher, the system produces a history of IBI that are predicted by calculating the auto-correlation of onset time vectors.

Auto-correlation is calculated every 11.6msec and the system maintains every prediction of IBI as a history of the signal analysis. Using the history of the IBI, the most frequent value of the IBI is predicted by calculating the auto-correlation output from the musical audio analysis stage. The system uses IBI history to prevent the output of temporally wrong predictions of IBI and lower the sensitivity of changes in prediction by calculating auto-correlation. In the interactive mode, the input is an audio signal that is obtained from user's keyboard which allows the change of tempo to be assumed. To track the tempo changes in audio signal, the system must be sensitive to changes, so another algorithm is needed. If the input is from the user's keyboard, the audio signal analysis stage must maintain stability in predicting the IBI and be sensible to tempo changes. To be sensible to tempo changes, the system resets the history of the IBI if the IBI prediction is calculated and auto-correlation fulfills the following two conditions. One is that the predicted IBI differs by ±50 msec compared with the frequent IBI in the history. Another is that the predicted IBI is within ±50 msec compared with the previous predicted IBI. The system decides that the tempo of the musical audio signal has changed if these two conditions are fulfilled five times in a row. By using this algorithm, the system can maintain stability to temporal tempo changes and decrease incorrect predictions as well as track main tempo changes in the audio signal.

6.2 Motion Change of the Robot by the Integration Value of Decibels

In an interactive mode, audio signals from the keyboard are used as input. Users can control the tempo of the robot's dance motion by changing the tempo of tapping the key. Also, the user can control the range of dance motion by changing the strength of key tapping. This is realized by calculating the integration value of the decibel of audio signals from the keyboard and selecting key poses that correspond to that value. Motor angles of four key poses are stored in one text file as one unit. Balances of the robot through transitions of key poses in units are checked by the motion editor. Last key pose in an unit is set as a standing pose, which is a neutral pose of the robot, so that the robot can maintain the balance through the transition to another motion unit. Each dance motion text file in the database is classified into three groups according to the range of motion. In the present implementation, there are five units of motion for each group, which are fifteen units and sixty key poses in total. The group the motion text file belongs to is decided by the range of value changes of each motor angle. When the integration value of the decibel is obtained from the audio signal, the system determines which group of key pose text files to select from. The value is rescaled between 0 and 1. Possibility of appearance of each group is determined by sending the value to fuzzy functions and one group is selected using that possibility. Fuzzy functions are used to emulate the ambiguity of humans when selecting dance motion. The system finally selects the next text file of dance motion to play from that group. If the system selects the same group as last time, next text file is selected randomly in the same group. By continuing this process the user can control the range of the robot's dance motion using a keyboard and changing the strength of key tapping.

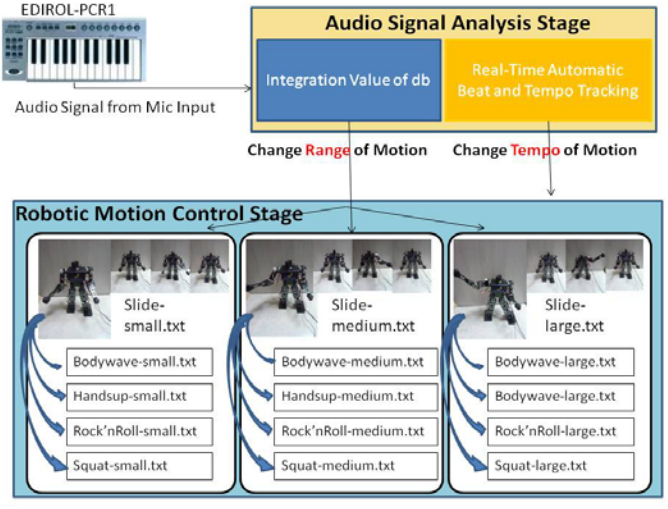

Fig. 7. Changing motions of the robot by the integration value of decibel and tempo of audio signal

7 Evaluation

7.1 Beat Tracking Result

Ten songs were used to evaluate audio signal analysis part of our system. We retrieved the time of beat in music by using the software called "AudaCity"**[7],** which can play and display the waveform of the music at the same time. Then we compared the time of each beat with the time of beat retrieved from beat tracking part of the system. Thirty seconds, from twenty seconds to fifty seconds of each song were used in the evaluation. Table2 is a list of ten different songs, and an average error of every beat in a song. Average of all ten song's average errors were 36.1msec.

Table 2. Result of the Beat Tracking

Title	Artist	Type	BPM	Average Error
Love me do	The Beatles	Rock	147	37msec
Rock 'N' Roll Star	Oasis	Rock	139	32msec
Holiday	Green Day	Rock	147	26msec
Movin' on without you	Hikaru Utada	Pop	122	39msec
Together	EXILE	Pop	112	54msec
Another Story	Mr.Children	Pop	91	36msec
FAKE	Mr.Children	Pop	125	32msec
Dream Fighter	Perfume	Techno-pop	135	19msec
BANZAI	MISA	Trance	145	42msec
Nobody Knows	DARK-ONE	Trance	145	45msec

7.2 Dance Motion Made by the System

First we used a music file as an input. Without using the motion change method, the robot just danced to the beat of the music in a single motion, but by using the algorithm proposed above, various combinations of motions were played according to the change of the input signal, instead of just playing the same motion many times. The motion change algorithm helped to reduce the monotony of dance motions.

Next, we used the keyboard as an input device and played some chords on it. The robot started to dance on the tempo of the tapping of the keyboard. If the keyboard is tapped weakly, the dance motion of the robot changed to small motion, and if tapped strongly, the dance motion changed to large motion. If we changed the tapping of the keyboard slower to faster or faster to slower, the robot tracked and changed the tempo of the dance in 5 to 10seconds.

8 Conclusion

We developed a system that controls and generates a humanoid robot's dance motion in real-time using the timing of beats in musical audio signals. The beat is extracted from the musical audio signal in real-time, allowing the measurement of intervals between beats and the prediction of the next beat. By taking history of IBI, and clearing the history if certain conditions are fulfilled, the system maintains stability of prediction and tracks tempo changes faster than past real-time beat tracking models. We also proposed an algorithm to change the tempo and the range of dance motion of a robot interactively by using keyboard as an input device and changing the tempo and strength by key tapping. By using this algorithm, the system can interactively change the dance motion of robot according to the input and reduce the monotony of generated motions.

References

1. Masataka, G., Yoichi, M.: Real-time Beat Tracking for Drumless Audio Signals –Chord Change Detection for Musical Decisions, Speech Communication, 311–335 (1999)
2. Masataka, G.: An audio-based real-time beat tracking system for music with or without drum-sounds. Journal of New Music Research 30(2), 159–171 (2001)
3. Takaaki, S., Atsushi, N., Katsushi, I.: Detecting Dance Motion Structure through Music Analysis. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 857–862 (2004)
4. An Official Site of ASIMO, `http://www.honda.co.jp/ASIMO/`
5. Kazuyoshi, Y., Kazuhiro, N., Toyotaka, T., Yuji, H., Hiroshi, T., Kazunori, K., Tetsuya, O., Hiroshi, O.: A Biped Robot that Keeps Steps in Time with Musical Beats while Listening to Music with Its Own Ears. In: International Conference on Intelligent Robots and Systems, pp. 1743–1750 (2007)
6. Abraham, S., Marcel, J.E.G.: Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Analytical Chemistry 36(8), 1627–1639 (1964)
7. An Official Site of Audacity, `http://audacity.sourceforge.net/`

8. Shinichiro, N., Atsushi, N., Kazuhito, Y., Hirohisa, H., Katsuhi, I.: Generating Whole Body Motions for a Biped Humanoid Robot from Captured Human Dances. In: IEEE International Conference on Robotics and Automation (2003)
9. Dixon, S.: A Beat Tracking System for Audio Signals. In: Proc. of Diderot Forum on Mathematics and Music, Vienna, Austria (1999)
10. Simon, D.: A Lightweight Multi-Agent Musical Beat Tracking System. In: AAAI Workshop on Artificial Intelligence and Music (2000)
11. Shinozaki, K., Oda, Y., Tsuda, S., Nakatsu, R., Iwatani, A.: Study of dance entertainment using robots. In: Pan, Z., Aylett, R.S., Diener, H., Jin, X., Göbel, S., Li, L. (eds.) Edutainment 2006. LNCS, vol. 3942, pp. 473–483. Springer, Heidelberg (2006)
12. Wama, T., Higuchi, M., Sakamoto, H., Nakatsu, R.: Realization of Tai-chi Motion Using a Humanoid Robot. In: Rauterberg, M. (ed.) ICEC 2004. LNCS, vol. 3166, pp. 14–19. Springer, Heidelberg (2004)
13. Foote, J.: Content-Based Retrieval of Music and Audio. In: Multimedia Storage and Archiving Systems II, Proceedings of SPIE, pp. 138–147 (1997)
14. Scheirer, E.D.: Tempo and beat analysis of acoustic musical signals. Journal of Acoust. Soc. Am. 103(1), 588–601 (1997)