HOKKAIDO UNIVERSITY

Title	Automatic Generation of Character Animations Expressing Music Features
Author(s)	Loi, Ka-Chon; Li, Tsai-Yen
Citation	Proceedings : APSIPA ASC 2009 : Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, 216-221
Issue Date	2009-10-04
Doc URL	http://hdl.handle.net/2115/39676
Type	proceedings
Note	APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference. 4-7 October 2009. Sapporo, Japan. Oral session: 3D Computer Graphics and Applications (5 October 2009).
File Information	MP-L1-6.pdf

Instructions for use

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

Automatic Generation of Character Animations Expressing Music Features

Ka-Chon Loi* and Tsai-Yen Li*
*Computer Science Department, National Chengchi University, Taipei, Taiwan.
E-mail: {g9506, li}@cs.nccu.edu.tw Tel: +886-2-29387642

Abstract — **In this paper, we propose to use procedural animation of a human character to enhance the interpretation of music. The system consists of a procedural motion generator which generates expressive motions according to music features extracted from a music input, and uses Dynamic Programming (DP) to segment a piece of music into several music segments for further planning of character animations. In the literature, much animation research related to music uses reconstruction and modification of existing motions to compose new animations. In this work, we analyze the relationship between music and motions, and then use procedural animation to automatically generate expressive motions for the upper body of a human character to interpret music. Our experiments show that the system can generate appropriate motions for music of different styles and allow a user to modify system parameters to satisfy his/her visual preferences.**

I. INTRODUCTION

The recent advances in multimedia technologies have enabled new types of applications combining visual and audio effects to create enhanced entertainment. It is common to see videos or image slide show be accompanied with appropriate music, possibly chosen by the computer. On the contrary, although desirable, it is less common to see 3D character animations accompanying music, probably due to the high cost of producing computer animation. As the technologies for 3D graphics are improving rapidly, real-time 3D animations are becoming more feasible. Nevertheless, we have not seen many computer animations that can be automatically generated based on the interpretation of music.

Among the possible ways of expression, dance is one of the most popular forms that can be used to interpret or enhance music. For example, some recent research focused on composing new dance motions based on existing motion libraries [11]. Dances performed by professional dancers may be difficult for regular audience to understand sometimes. Nevertheless, most people can respond to music by casual motions such as moving the hands, swinging the body, or shaking the legs. These types of motions are typically impromptu motions based on personal interpretation of music features such as rhythm, melody, etc. In this paper, we propose to analyze the features of music and model the relationship between music and motion in order to automatically generate casual expressive motions for the upper body of an animated character to enhance the interpretation of music. However, the generated motions are casual motions that are not necessarily constrained by the general rules followed by music conductors.

We organize the rest of the paper as follows. We will first review the previous work pertaining to this research. Then will give an overview of the system that we have designed in Section II. We will then describe how we segment the music for the generation of motion segments of appropriate lengths. In Section IV, we will describe our procedural animation approach to the generation of synchronized upper-body motions. Then we will present some of the results obtained in our experiments and conclude at the end with future research directions.

II. RELATED WORK

Melody is the most essential feature of music that can be remembered more easily. Delone [2] stated that the main elements of a melody include duration, pitch, quality (timbre), texture and loudness. However, relatively speaking, quality is less important because their experiments showed that the same melody can be recognized by users even when played with a wide variety of timbres, textures, and loudness. Dowling [3] discovered that melody contours are easier to remember than exact melodies. A contour refers to the profile of the melody, indicating whether the next note goes up, down, or stays at the same pitch. It was found that it was relatively easy to distinguish two melodies that have different contours.

In the literature of character animation related to music, we can find three types of research with different objectives. The research in the first category aimed to generate music according input motions. For example, Mishra and Hahn [9] proposed to generate music according to the motions performed by a human in a virtual reality setting. Nakamura et al. [10] also proposed to use motions to compose background music. The main differences between the two works were that the former generated music in real time based on music features and motion features while the latter selected background music from a music library in a post-processing step.

The second category of research used music to generate motions. The authors in [11] used a machine learning approach to analyze the relationships between motions and music by a great amount of synchronized examples. The system aimed to generate animations for new music inputs. The authors in [1] proposed to further analyze the waveform representation of an audio soundtrack in addition to the MIDI format to consider different aspects of the music for the automatic generation of synchronized musical animations. The authors in [4] also proposed to synthesize animations based

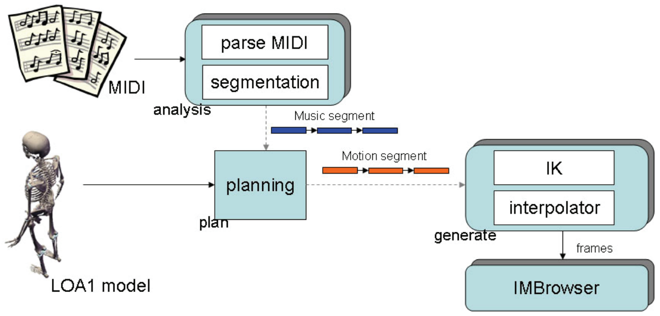


Figure 1. System Overview

on the beat analysis of motion captured data and input music. Our work belongs to this category. In this paper, we propose to generate the upper body motion from scratch by creating a mapping between the music features and procedural animations as described in the next few sections.

The third category of research studied the interaction between music and motion. For example, the authors in [7] extended the music theories in [1] and proposed *music graph*, a concept similar to motion graph for animation. A music graph consists of music clips as nodes and the possible transitions as arcs. Music is composed in real time with appropriate time scaling for each music clip according to the input motions. The authors in [13][14] analyzed dance performance to extract the correspondence between music features and motion features. Then animations are synthesized from the motion library according to a new piece of music input.

III. SYSTEM OVERVIEW

An overview of the system proposed in this work is shown in Figure 1. In this work, we focus on using the upper body to create animations for expressing music. The upper body of a human figure is composed of four parts: *right arm*, *left arm*, *head*, and *body*. The two arms are used to express melody and rhythm while head and body are used mainly to express rhythm. The source input of the system is a piece of music in the MIDI format with the melody and rhythm tracks identified. The system parses the MIDI file and segments the music into several music segments of appropriate lengths by using a DP approach. Next, in the planning module, we use the music features such as melody, rhythm, and loudness in each music segment to plan a feasible trajectory under kinematics constraints for the bodies of the human character to trace. Then, we use the inverse kinematic [16] technique to generate the motions for the human character with appropriate timing. Together with the music, the motions are then rendered and synchronized in real time in a 3D display engine, called IMBrowser, developed in our previous work [7].

IV. MUSIC SEGMENTATION

Due to the kinematics constraints of a human figure, the reachable space by hands without moving the global position of the body is also limited. Therefore, in order to express a long piece of music with the upper body, one must divide a piece of music into several music segments of appropriate lengths for the hands to trace back and forth. However, how do we perform the segmentation such that all constraints can be satisfied and certain criteria can be used to create sound visual effects? In this section, we will describe how we design the criteria to compose the objective function that is used to search for the optimal segmentation of music for a given piece of music and a human character.

Algorithm: Segment(melody)

```
Table T ;
while length from minimal to melody.length
begin
 while i from 0 to melody.length – length - 1
 begin
  while mid from i + 1 to i + length - 1
  begin
   value = normalize( T[i][mid], T[mid][i + length]);
   if value <= T[i][i + length]
     T[i][i + length].segmentAt = mid;
     T[i][i + length] = value;
  end
 end
end
```

Figure 2. Algorithm for segmenting music

A. *Music feature space*

We assume that the input of the system is a music file in the MIDI format with appropriate melody and rhythm channels. From these two channels, we can express a clip of music with a sequence of notes $s = [n_1, n_2, \ldots, n_m]$. A note n_i is described by four features [*pitch*, *duration*, *intensity*, *isBeat*]. Pitch ranges from 0 to 127 with octave as the unit; duration ranges from 0 to 64 with the sixteenth note as the unit; intensity ranges from 0 to 120. Pitch, duration, and intensity are extracted from the melody channel while the *isBeat* feature is extracted from the rhythm channel. A note in the melody channel is a beat note only if there is also a note in the rhythm channel.

B. *Objective function*

A melody contour is described by pitch and duration. In this work, we propose to map pitch and duration to the vertical and horizontal components of a hand position. Since the reachable space of a hand is limited, there exist limits for these vertical and horizontal positions. In addition, since the end of a segment is a turning point with discontinuous velocity and acceleration, it is desirable to have this end point at a beat point of the music. In order to determine the goodness of a music segment in the search algorithm to be described in the next section, we have designed the objective function as follows. We assume that a music segment s is composed of m notes and can be represented as $s = [n_1, n_2, \ldots, n_m]$. The objective function f is composed of four components:

$$f_1(s)=isBeat(s.n_m) \quad (1)$$

$$f_2(s)=pitchRange(s) \quad (2)$$

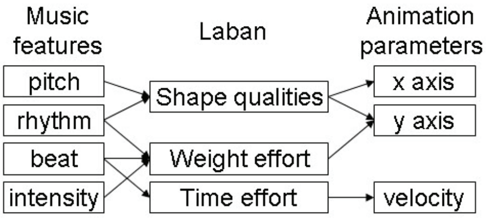


Figure 3. Mapping between music features and animation parameters

$$f_3(s)=rhythmDensity(s) \tag{3}$$

$$f_4(s)=totalDuration(s) \tag{4}$$

$$f(s)=w_1 * f_1(s) + w_2 * f_2(s) + w_3 * f_3(s) + w_4 * f_4(s) \tag{5}$$

If the last note is at a beat or a rest, f_1 return 1, otherwise 0. The function f_2 is used to determine how far the range of the pitch in this segment is from the ideal range (say, 12 half-tones). The function f_3 is used to determine how far the density of the beats in the segment is from the average density of the whole music. The fourth function f_4 is used to constrain the total duration of the segment to an ideal value as much as possible. These four component functions are normalized into the range of [0, 1] and then combined with appropriate weights to form the final objective function in eq. (5). These weights are subjective parameters that can be specified by the users with a graphical user interface provided by the system. The smaller the values returned by this function, the better the results in the search.

C. Segmentation algorithm

The problem of segmenting a piece of music is similar to the problem of multi-paragraph text segmentation, where one needs to find a way to segment a paragraph into lines such that the total amount of blank space is optimized globally [4]. This problem can be nicely solved with the classical DP approach. The DP algorithm used in our system is shown in Figure 2. We use a two-dimensional table T to record the optimal segmentation point for a given segment specified by the two indices of T. The procedure consists of three loops. The first loop is performed over the length of what a segment can have from the shortest to the longest. The second loop is performed over the starting location of a segment, and the third loop is over the possible new segmentation points to search for better segmentation. The whole DP algorithm is performed in a bottom-up fashion such that examination of longer segments can make use of the results from shorter segments. The overall complexity of the algorithm is $O(n^3)$, where n is the number of notes in the music. The final segmentation is determined by recursively examining the segmentation points (if exist) of the segments starting from T[0][melody.length]. The output of the algorithm is a sequence of segments $s_1, s_2, \ldots, s_k$, $k>0$ with an optimal value in the objective function.

V. PLANNING MOTIONS

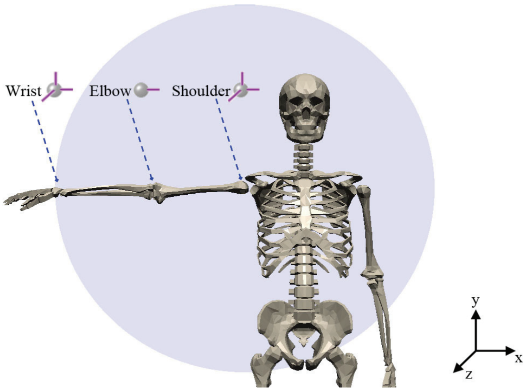


Figure 4. Reachable space and the coordinate system for the right arm

A. Segmentation algorithm

According to Laban Movement Analysis (LMA) [6], the motion of a human figure can be described with four different aspects: *body*, *effort*, *shape*, and *space*. Among these aspects, shape and effort are the ones that are more related to the expression of affect and therefore can be used to interpret music. Even though not professional dancers or conductors, most people can easily use body motions to express their interpretation of music. In these casual motions, simple rules can be deduced. For example, the rhythm of a motion is synchronized with the rhythm of music. Similarly, the strength of a motion can also reflect the loudness of music.

Therefore, we propose to map music features into three features (shape qualities, weight effort, and time effort) in LMA and then map these features into motion parameters (position and velocity) in an animation as shown in Figure 3. The melody and rhythm of music are reflected on the vertical (y) and horizontal (x) locations of the hand while beat affects the heights of via points and velocity changes along the generated trajectory. For example, according to [12], sudden changes in speed and orientation give strong indication of beat points. Larger intensity in music can also be expressed by larger vertical changes of the hand. With these principles, we attempt to plan the trajectory of the hand of a human character according to a given music segment. The motion planner is composed of three parts: setting keyframes, creating trajectory curves, and determining the speed of the hand along the trajectory as described below.

B. Keyframes

We use the right hand of a human character to express melody of a music segment as shown in Figure 4. The keyframe locations of the right hand in a music segment are determined mainly by the attributes of the notes. The horizontal axis (x) represents the time while the vertical axis (y) represents the pitch. Since the hand location is limited roughly by a sphere centered at the shoulder, we need to first determine total duration and pitch range of a music segment in order to know how to scale the keyframe locations to fit the motion for the whole music segment in the reachable space. The formula for determining these keyfarmes are shown in the equations below.

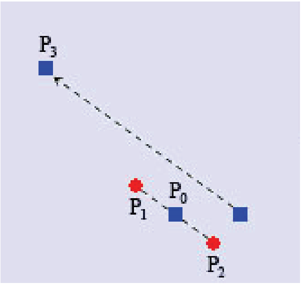


Figure 5. Illustration of how the control points P1 and P2 are aligned with the keyframe point P0 to create smooth transition at P0.

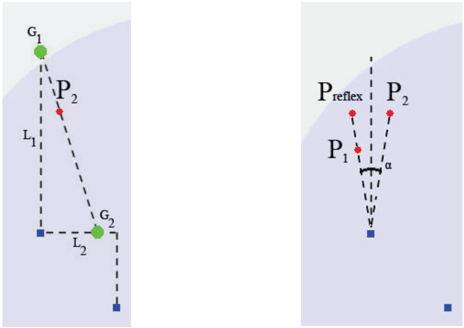


Figure 6. Illustration of how P1 and P2 are computed at a beat note

$$P(n_i) = \begin{cases} P_{init} & , \quad i=0 \\ (P(n_{i-1}).x + x(n_i), y(n_i), P(n_{i-1}).z) & \quad otherwise \end{cases} \tag{6}$$

where

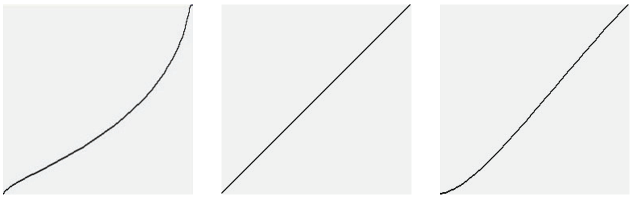
$$x(n) = n..duration / segment\_duration * X_{scale} \tag{7}$$

$$y(n) = (n_i.pitch - n_{mid}.pitch) * Y_{scale} \tag{8}$$

The direction of the x-axis is changed alternatively such that the arm can move back and forth without resetting the hand position. As shown in eg. (7), the horizontal movements (x) are scaled with X_{scale} such that the hand can span the whole reachable horizontal range during the whole music segment. For the vertical dimension, since the range of pitch for each music segment is different, we map the median pitch of the music segment to the mid-point of height (y-axis) in the reachable range. The height of the hand position for each node can then be computed relative to the mid-point, as shown in eq.(8).

D. Trajectory generation

The trajectory along the keyframes consists of C^0 continuous Bezier curves controlled by control points defined in accordance with the model of music features. According to the fact whether a keyframe is on a beat, on a rest, or neither, we construct the curve differently. If the keyframe is for a regular note (not on a beat or on a rest), we create a C^1 smooth curve passing this keyframe location. For example, as shown in Figure 5, P_0 and P_3 are the keyframe points while P_1 and P_2 are the control points controlling the curves connecting at P_0. For regular notes, P_0, P_1, and P_2 should lie on the same line.



(a) beat note (b) non-beat note (c) non-beat note after beat note

Figure 7. Speed profiles for three difference cases

Algorithm: right hand planning(music segments)

```
P_init = current wrist location
for each segment in music
begin
  plan key position for each note in segment
  if any key position is out of bound
    rescale y
  P_init = the last key position
end
KeyPos = the first key position of second segment
for each segment in music
begin
  plan trajectory between key positions
  if any part of the trajectory is out of bound
    rescale y
  KeyPos = the first key position of next segment
end
```

Figure 8. Motion planning algorithm for the right hand

If the keyframe is at a beat or a rest note, we create abrupt changes in slopes at both sides of this keyframe. In Figure 6, we show how the control points (P_1 and P_2) of the Bezier curves around the keyframe point (small blue square) are computed. First, P_2 is a linear combination of G_1 and G_2, defined by two distances L_1 and L_2. L_1 is determined by duration of the note such that the speed limit for the hand is not violated, and L_2 is a user-specified parameter. The intensity of the note determines where P_2 is located between G_1 and G_2. Second, as shown at the right of Figure 5, P_1 is located between the keyframe point and the vertical reflex point of P_2. The location of P_1 between these two points or the amount of bounce back from the reflection is specified by the user.

E. Determining speed profile

Once a trajectory is determined, we also need to determine the speed profile along the trajectory to express the time effort of the motion. We also use Bezier curves to define the desired velocity profile in the space of arc length and time. There are three types of speed profiles in our system. The first type is for beat notes as shown in Figure 7(a). We express the sense of beat not only on the trajectory discontinuity in slope but also on the change of speed at the beat point by a sudden stop after acceleration. The second type, as shown in Figure 7(b),

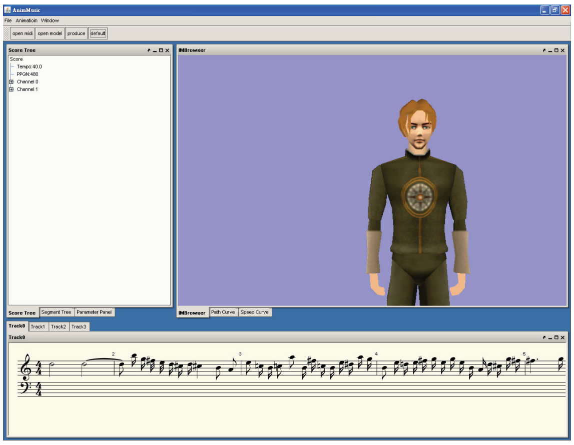


Figure 9. Graphical user interface of the system

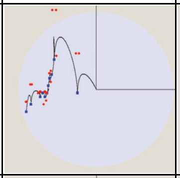
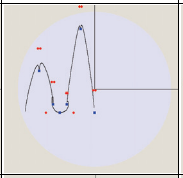
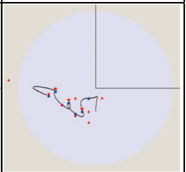
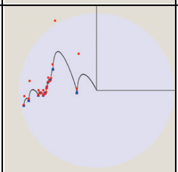
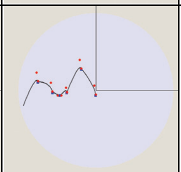
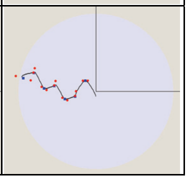
Exp. No.	1st trajectory	2nd trajectory	3rd trajectory
E1			
E2			

Figure 10. Trajectories created with two different parameter sets for three consecutive music segments

is a regular note where a constant speed is preferred in order to produce smooth motion. The third type is for a regular note whose starting speed is zero because of the prior note is a beat, as shown in Figure 7(c).

F. Motion planning algorithm for the right hand

The algorithm for generating the trajectory of the right hand according to the principles described in the previous subsections is listed in Figure 8. The input to the algorithm is music that has already been properly segmented, and the output is a sequence of unidirectional trajectories for the right hand to move back and forth in front of the character. In each music segment, if any part of the generated trajectory will cause the hand to be out of bound, the whole trajectory will be re-scaled in the y direction automatically (lines 4-7). These keyframes are then used to generate the trajectories (lines 12-14). A trajectory is also re-scaled to reduce the height if any part of it is out of the reach of the right hand.

G. Designing the motion of other body parts

While the right hand is used to express melody and rhythm of music, the left hand and other parts of the body are used to compensate the right hand. For example, the left hand is used to express long notes and staccato. When a note is longer than a quarter of the whole note, we extend the left hand outward to stress the long duration of the note. When a staccato note is encountered, the left hand is used to perform a rapid drop and sudden stop. The body and the head are also used to stress rhythm by nodding and swinging. However, in order to avoid rapid swinging or nodding, we require a minimal duration between two beat points for changing the direction of these two types of motions.

Figure 11. Snapshots of animations generated in experiment E1.

Figure 12. Snapshots of animations generated in experiment E2.

VI. EXPERIMENTAL RESULTS

The system described above has been fully implemented in Java. The graphical user interface (GUI) of the system is shown in Figure 9. In order to evaluate the effectiveness of our system, we have conducted experiments with different size of characters on different sets of parameters for different styles or user preferences. For example, in Figure 10, we show the trajectories of the first three music segments generated with two sets of parameters for the music of “Air on the G String” by Bach. In the first experiment (denoted as E1), the L_1 parameter is larger than the one in the second experiment (denoted as E2). Consequently, the magnitude of the curves in E1 is much larger than the ones in E2. The snapshots of the animation generated for this music in experiment E1 and E2 are shown (left to right and up to down) in Figure 11 and Figure 12, respectively. Note that the left hand of the character moves from time to time along with the major

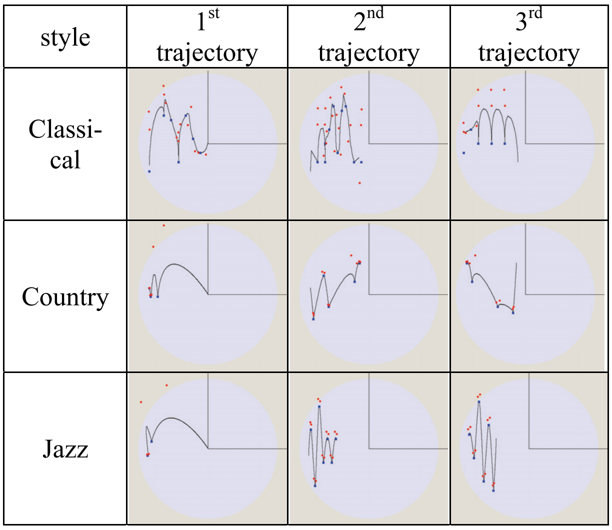


Figure 13. Trajectories created for different classes of music in three consecutive music segments

movement of the right hand. The body swings left and right synchronized to the beats in the music. We have also tried the system on various animated characters of different size and three different styles of music, including classic, country, and jazz. The trajectories of three typical segments for these three types of music are shown in Figure 13.

VII. CONCLUSIONS

In this paper, we have developed an intelligent motion generator for an animated character to use upper body motions to interpret music features. The system accepts general inputs such as a LOA1 humanoid model and MIDI music and allows a user to specify his/her preferences when generating different styles of motions. The motion generation is facilitated by a mapping between music features and the motion parameters in procedural animation. In our experiments, we have shown the effectiveness of the system through several types of music and human models.

In the current system, several parameters such as smoothness still rely on user tuning for producing more desirable results. In the future, we hope to investigate the relation between these parameters and music features such as tune and timbre to choose these parameters automatically. In addition, we would like to extend the current animation procedures to include full-body animations with global movements. Furthermore, although we believe that music can be effectively enhanced with appropriate animations, in the future we will conduct in-depth user studies to evaluate the overall effectiveness of the system and its potential applications.

ACKNOWLEDGMENT

This research was funded in part by National Science Council of Taiwan under contract no. NSC96-2221-E-004-008.

REFERENCES

[1] M. Cardle, L. Barthe, S. Brooks, and P. Robinson, "Music Driven Motion Editing: Local Motion Transformations Guided By Music Analysis," in *Proc. of the 20th Eurographics UK Conf.*, 2002.

[2] R. Delone, "*Aspects of Twentieth-Century Music*," Englewood Cliffs, New Jersey: Prentice-Hall, Chap. 4, pp.270-301, 1975.

[3] W.J. Dowling, "Scale and Contour: Two components of a theory of memory for melodies," *Psychological Review*, vol. 85, pp.341-354, 1978.

[4] T.H. Kim, S.I. Park, and S.Y. Shin, "Rhythmic-Motion Synthesis Based on Motion-Beat Analysis," *ACM Transactions on Graphics*, 20(3):392-401, 2003.

[5] D.E. Knuth, and M.F. Plass, "Breaking Paragraphs into Lines," *Digital Typography*, 1999.

[6] R. Laban, and L. Ullmann, *Mastery of Movement*, Princeton Book Company Pulishers, 1960.

[7] H.C. Lee, and I.K. Lee, "Automatic Synchronization of Background Music and Motion in Computer Animation," *Computer Graphics Forum*, vol. 24, pp. 353-362, 2005.

[8] M.Y. Liao, and J.F. Liao and T.Y. Li, "An Extensible Scripting Language for Interactive Animation in a Speech-Enabled Virtual Environment," in *Proc. of the IEEE Int'l Conf. on Multimedia and Expo*, 2004.

[9] S. Mishra, and J.K. Hahn, "Mapping motion to sound and music and in computer animation and VE," in *Proc. of the Pacific Graphics '95*, 1995.

[10] J. Nakamura, T. Kaku, T. Noma, and S. Yoshida, "Automatic Background Music Generation Based on Actors 'Emotion and Motions'," in *Proc. of the Pacific Graphics*, 1993.

[11] S. Oore, and Y. Akiyama, "Learning to Synthesize Arm Motion to Music By Example," in *Proc. of the 14-th Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2006.

[12] T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Detecting dance motion structure through music analysis," in *Proc. of IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, 2004.

[13] T. Shiratori, A. Nakazawa, and K. Ikeuchi, "Dancing-to-Music Character Animation," in *Computer Graphics Forum*, vol. 25, pp. 449-458, 2006.

[14] L. Torresani, P. Hackney, and C. Bregler, "Learning Motion Style Synthesis from Perceptual Observations," in *Proc. of the Neural Information Processing Systems Foundation*, 2006.

[15] A.L. Uitdenbogerd, and J. Zobel, "Manipulation of music for melody matching," in *Proc. of ACM Int'l. Multimedia Conf.*, 1998.

[16] IKAN(Inverse Kinematics using Analytical Methods). http://cg.cis.upenn.edu/hms/software/ikan/ikan.html