Music-Driven Motion Editing: Local Motion Transformations Guided By Music Analysis

Marc Cardle
mpc33@cl.cam.ac.uk

Loic Barthe
lb282@cl.cam.ac.uk

Stephen Brooks
sb329@cl.cam.ac.uk

Peter Robinson
pr@cl.cam.ac.uk

Computer Laboratory
University of Cambridge

Abstract

This paper presents a general framework for synchronising motion curves to music in computer animation. Motions are locally modified using perceptual cues extracted from the music. The key to this approach is the use of standard music analysis techniques on complementary MIDI and audio representations of the same soundtrack. These musical features then guide the motion editing process. It allows users to easily combine different aspects of the music with different aspects of the motion.

1. Introduction

Creating animation, with or without the use of a computer, has always been a very time-consuming process. The advent of computer-aided animation has allowed computers to perform a great deal of the effort involved in this type of work, especially for tasks such as animating human characters, dealing with collisions, and sound synchronization. Sounds are generally associated with motion events in the real world, and as a result there is an intimate linkage between the two. Hence, producing effective animations requires synchronisation of the sound and motion, which remains an essential, yet difficult, task in animation.

Computer animation packages use the concept of positional keyframes to specify points through which scene objects pass. Animators usually have a good sense of specific key positions. However in general, the inexperienced animator may have less intuition about the precise timing of the keyframes and the velocity of each object as they pass these unknown points in time. This problem is exacerbated when trying to synchronise a scene to music and sound.

Recent tools have been developed to automatically synchronise speech signals to lip movement animation [6,25]. More generally, the popular WinAmp visualizations use audio signals to seed abstract animations, whilst Lytle's [17] system generates motion curves from raw MIDI note data. These systems are effective at their targeted application area, however there is a need for a more general approach to help animators modify existing motions to music.

The goal of this work is to combine the power of advanced music analysis performed on both MIDI and audio signals with motion editing to simplify and automate the generation of synchronised musical animations. This unified framework will enable animators to try out different combinations of music/sound analysis techniques with familiar motion editing methods without having to acquire a deep understanding of either.

The paper first relates previous, usually problem-specific, approaches to synchronizing motion and music. We adopt Lytle's fundamental contribution as a premise to our investigation. We follow this with an overview of our modularised musical editing system, which leads us to present the music analysis techniques that we intend to apply on both MIDI and audio, as well as the associated methods to transform the motion. Finally, an illustrative usage scenario of our system is given along with a discussion of further research opportunities.

2. Related Work

The approach taken in [9,18] is to control music and sound generation directly from the animation. In these systems the motion parameters are tied to parameterisable musical constructs such as Timbre-trees [10], C-Sound models, or music generators [19]. The quality of the soundtrack is then directly related to the effectiveness of the sound representation. Our system takes the inverse approach, where the soundtrack is left unchanged, whilst the motion is altered.

There have been previous attempts at using musical parameters to generate novel motion. Woodgain [31] used MIDI to generate expressive drumming movements, an animated digital band followed the MIDI score in Diva [11,26], Fakespace Music [1] uses low-level MIDI note information to move objects in 3D space, and Hwang [8] represents tracks and melody progressions as orbiting planets. The Improv improvisational animation system [22] made use of Rowe's algorithmic musical analysis [28] to create *Flock of Words*, an abstract musical animation. These approaches demonstrate impelling results for their purposes; however, they do not offer a unified and generalized framework for music-to-motion mappings.

Closest to our work is Lytle's approach [17], where a common MIDI score is used to drive both sound synthesisers and an animated orchestra in *More Bells and Whistles* [16]. Their focus was on extracting low-level parameters from the MIDI score such as note pitch, note velocity, duration, pitch bend, and volume, and using them as motion curves. This constitutes a one-to-one mapping between notes data and motion. A deeper understanding of the current musical context should provide a more informed discussion as to when, and by how much, the motion should be affected. It would also be valuable to re-use and modify existing motions, such as user-defined keyframes and motion capture, rather than trying to incorporate unadapted musical motions into animations.

Finally, the systems described above chose to base their animations on either MIDI or audio representations, of the same soundtrack. Extracting note information from MIDI is straightforward, yet, a great deal of sonic information is lost in MIDI representations. What is more, extracting simple score information from audio is arduous at best. Alternatively, by combining information from both MIDI and audio version of a same soundtrack, we get the best of both worlds. The following section identifies the aspects of our system that deal with these issues.

3. System Overview

Our system extends Lytle's work, by not only using low level musical features, but also complementing this with music analysis to extract perceptually significant musical features in both the MIDI and analogue audio domains. In turn, these features are then visually translated, not by creating entirely new motion curves, but by editing existing motion curves such as keyframe animation and motion capture.

Hence, the system consists of three distinct modules:

- MIDI analysis module
- Audio analysis module
- Motion editing module

The output of the two analysis modules is used by the motion editing module to alter the final motion. Figure 1 shows the relationships between system components. Initially, a MIDI soundtrack is fed into the system where the music analysis is carried out on the raw MIDI data. In parallel, an analogue audio rendition of the MIDI soundtrack, using a software synthesiser, is produced and fed into the audio analysis module. Next, the animator constructs a rough estimate of the final animation curve defined by a set of keyframes.

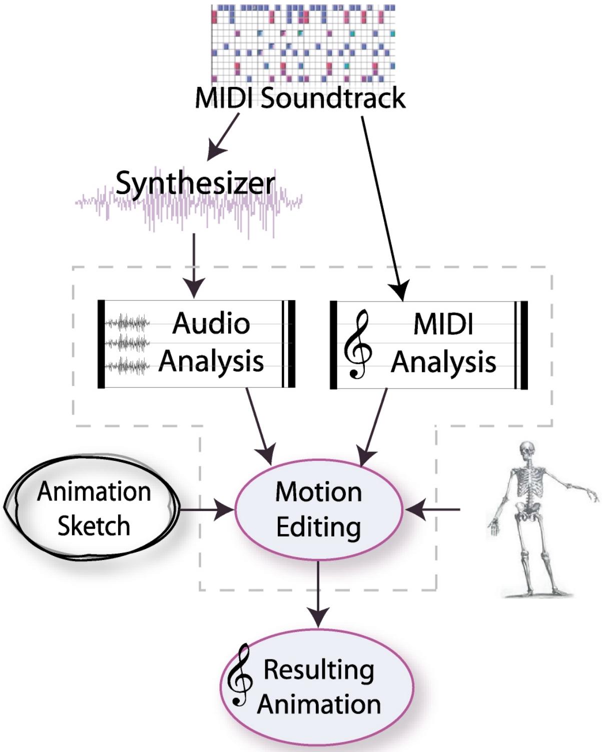

Figure 1: System Overview

This animation sketch, or motion capture sequence, is then used as a basis for the motion editing. The animator selects which music analysis parameters will modulate the chosen motion editing methods. Alterations to the motion can be layered to encompass multiple aspects of the music at once. The mapping process continues until the animator is happy with the resulting animation.

4. Music Analysis

Depending on the musical context and the targeted motion, different combinations of music features and motion editing techniques are effective. Hence, the system will provide a comprehensive selection of music analysis techniques and permit iterative testing of diverse combinations of music parameters with motion editing parameters.

Although we use standard computational music analysis techniques found in current computer music research, most of these methods need to be pre-processed or adapted before the mapping to motion editing parameters takes place. In addition to standard techniques, low-level music features (similar to Rowe's [24]) specifically designed for our needs were developed.

As mentioned in our previous work section, MIDI abstracts away from the acoustic signal level of music, up to a representation based on the concept of notes, comprised of pitch and velocity, that go on and off. Timbre, attack and envelope are examples of low-level audio aspects which cannot be captured using MIDI and yet are essential to modelling the complete music cognition experience. MIDI messages say nothing about the timbre of a sound beyond the selection of a MIDI program number.

Combining information from audio and MIDI renditions of a same musical segment will yield complementary perspectives on the analysed music. For example, performing chord recognition is simpler on MIDI streams since the notes are readily available; on the other hand, analysing frequency variations can only done on music in an analogue format. Depending on the nature of the music, a given feature might produce more accurate results than its counterpart in the other domain. The final decision as to which feature to use is left to the animator.

4.1 MIDI-based Analysis

A range of useful features can be extracted from a MIDI stream. Foremost, simply tracking the variations of in pitch and velocity (i.e. loudness) of MIDI notes, in a similar fashion to Lytle, can lead to satisfactory results. For example, loudness variations can be linked to variations in the displacement speed.

Chord information, obtained using chord recognition methods [24], is directly useable for seeding motion editing. For example, a change in chord type, e.g. minor to major, could cause a change in the motion. Identifying repetitions in chord sequences could also yield repetitions in the applied motion transformation.

Cognitive models of music can also be employed to identify perceptually important features. Pitch induction [24] finds the most salient pitch in a collection of pitches. Variations in virtual pitch should translate to significant visual changes, since pitch is so fundamental in our experience of music. Slight fluctuations in pitch by a few semitones are not interpreted into drastic changes in the motion. However, pitch variations exceeding a user-defined threshold could cause other motion filters to be selected.

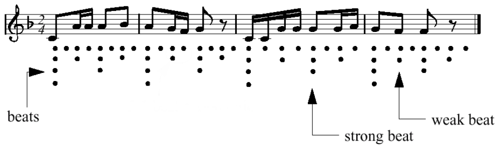

Figure 2: A Metrical Structure Hierarchy. Points, where beats of many levels align, are strong beats.

Beat perception and extraction models [24,29] enable us to extract a pulse of evenly separated beats. Some beat pulses become regularly accented with respect to their neighbours. We can identify which of them has greater structural importance by using metrical analysis [15] as illustrated in Figure 2. The intensity of the motion perturbations, originating from beat pulses, should be proportional to the beat level. Some motion alterations could be made more frequent by occurring at every mid-level beat, whilst more pronounced alterations are matched with strong beats.

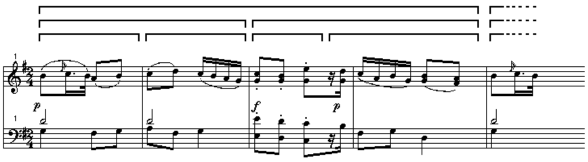

Figure 3: The brackets shows the different groupings obtained using segmentation methods

There are two organisational forces in the perception of music. The first, referred to as segmentation, is the grouping of musical events, and the other is the recognition of patterns. Since segmentation algorithms [24] identify perceptually different sections of the music, this should naturally be reflected in changes in the motion. For example, each segment can be characterised by a unique set of motion editing parameters. Consequently, the motion appears distinct from one musical segment to the other.

Music is mostly composed of patterns that are repeated and transformed. Patterns occur in all levels of music, including melody, rhythm, harmony, and texture. Once these patterns are identified using exact pattern matching or soft-pattern matching techniques [3], we can apply similar motion transforms to matching musical patterns. This will give a sense of visual echoing.

Finally, we have developed more pragmatic music analysis methods to help extract features that more formal approaches might not capture. Our ad-hoc features track changes in vertical note density, note spread across the pitch range, attack speed, note durations, inter-onset intervals, perceived loudness, and register. These features are made contextual by using the notion of Focus and

Decay [24]. The audio features, described next, can also benefit from this contextualisation mechanism.

4.2 Audio-based Analysis

The complete specification of an analogue soundtrack is distributed between the MIDI messages and the program resident in the synthesiser. A large amount of complex processing goes on at this level that has crucial influences on the final perception of the music. Hence our inclusion of audio analysis. The field of audio signal processing is large and complex, therefore this research will only deal with a small subset of this domain to limit the scope of research to a manageable level.

Because the musical score is readily available from the MIDI representation, automatic transcription systems, such as [12], are redundant here. More obvious features can be extracted from the audio such as sound bandwidth, spectrum energy and envelopes extracted using Short-Time Fourier Transforms, zero-crossing and fundamental frequency. The variations of these features can be directly mapped to variations in the motion.

Some of the above physical features are good approximations of perceptual features such as pitch with frequency and spectrum energy with loudness. Nevertheless, features such as timbre are difficult to quantify as it defines the quality of sound that allows the distinction of different instruments or voices sounding the same pitch. Auditory scene analysis methods such as [7] can be used to estimate features such as timbre.

Now that we have established which music properties we plan to use in both MIDI and audio, we will identify how we plan to modify the user-defined motion curves.

5. Motion Editing

In general, a motion-editing task may involve altering any aspect of the motion that we wish. Exactly what makes a particular motion a certain mood (happy, angry, sad), correlated to the music, or simply appealing to an animator may be difficult to capture algorithmically. The subtleties of timing, velocity, and small position perturbations add significantly to the lifelike feel of animation and its perceived relation to the music. Although a highly skilled animator may have the ability to directly specify this manually, many animators cannot or cannot afford the effort. If tightly controlled by the music analysis, current motion editing techniques offer a wide scope of editing possibilities. The user can roughly sketch out the overall shape of the motion curve, or even start with motion capture, and then specify which motion editing method to apply.

Special care needs to be taken when dealing with motion capture in our system. Motion capture emphasises a particular kind of editing operation where we begin with a essentially correct motion, and attempt to change it with regards to the music so as to preserve much of what we began with. In general, the changes we make are relatively small with respect to the overall action. This is important for two reasons: one, because the kinds of operations we will want to perform on the motion capture will be quite different than when we are sculpting a motion from a few keyframes; and two, it emphasises that as we change the motion capture, we often are interested in preserving other aspects of that motion as well.

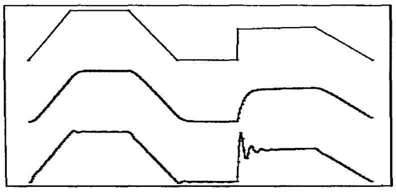

Figure 4: Two different IIR filters are applied to the top motion curve

We are currently focussing on a select range of motion editing techniques which we will briefly mention here; nonetheless, additional techniques can be later appended to the system if need be.

Motion signal processing, introduced by Bruderlin [2], uses a variety of standard signal processing tools to edit motion curves. As noted by Unama [30], scaling different frequency components can affect the emotional content of motions. Filter banks can be used to divide a motion signal into a number of components, which can be manipulated independently according to music parameters, and then reassembled to create a new motion signal.

Infinite impulse response (IIR) filters can smooth, overshoot, or add wiggle to motion as first shown in the Inkwell system [14] (See Figure 4). This can be used to give more or less expressiveness to motions as noted by [14]. Furthermore, non-linear filters such as the wave-shaping filter [2] can be used to introduce extra undulations. For example, if we apply identical motion filters for recurrent musical themes that were located using musical pattern matching, then each time the main melodic theme plays, movements of similar nature will execute. In a similar fashion, multi-resolution editing techniques, which manipulate each different level of motion details separately, enable different levels of the motion to be simultaneously linked to a variety of musical parameters.

Additive motion techniques enable us to blend two or more [23] motions together. Depending on the musical context, different foreground motions can be blended into the main motion. For example, stochastic noise functions

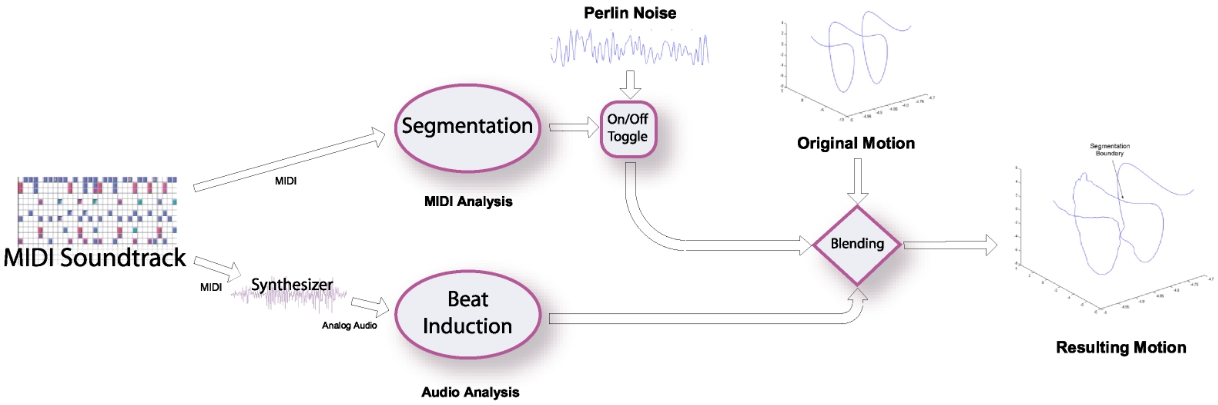

Figure 5: Overview of system usage

can also be added to the main motion in order to give it more texture as described in [21]. This noise enhances the animation and has the potential to create novel and organic-like musical animations. In motion warping [2], a specifically generated displacement map is blended in without disturbing continuity and global shape in the original motion. For example, each beat could add a point to the displacement curve. The strength of the beat would decide the deviation of the added point on the displacement map. The resultant animation would jump at each beat pulse.

Given a short animation sequence, we would loop certain parts of motion whilst gracefully making the transition between them. Some instances of the looped motion are modified to better accommodate the fluctuations of the music. The techniques of motion cyclification [27] are particularly appropriate here to make motion segments loopable. Furthermore, time warping [2] specific motion instances resizes them to better conform to the musical cues. To avoid repetitiveness and to make each motion instance more connected to the current musical context, perturbations, such as the ones described above, can be applied.

6. Example System Usage

The following section details a typical example of how our system is used. First, the animator inputs a MIDI soundtrack along with sketched keyframe animation. He then decides which music analysis features to link to which motion editing modules. In this simple case, the audio rendition of the music is used as a basis for beat induction, while segmentation is carried out on the MIDI as illustrated in Figure 5.

For each strong beat detected in the audio, the X=0 position of the perturbation curve, visible in Figure 6, is aligned with the beat time event. At which point, it is blended into the main motion. Hence, negative values of X represent time before the beat, whilst positive values of X represents time after the beat event. This enables anticipation and overshoot, which are conventional animation effects [13], to occur in the perturbed motion. Observe that our chosen perturbation curve is a 2D curve and the original motion a 3D curve in Figure 7. Consequently, the amplitudes of the perturbation curve are used to scale the normalized Frenet Frame's normals found for each point of the original motion. We actually use the quasi-normal, introduced by Coquillard [4], since the standard normal along a 3D path can be non-continuous or null at some points.

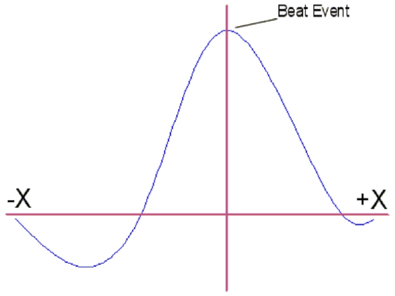

Figure 6: Perturbation induced by a musical beat where the beat event is located at X=0.

In parallel with the beat perturbations, the detected segmentation boundaries are used to segment the initial motion. To mark the transition from one musical segment to the other, we toggle the blending of Perlin noise on and off. The presence and absence of Perlin noise will give a distinctly different texture to the motion from one musical segment to the other. Hence the first segment starts out with added noise, whilst the next does not, and so on. Figure 7 shows the stages necessary to produce the final motion sequence.

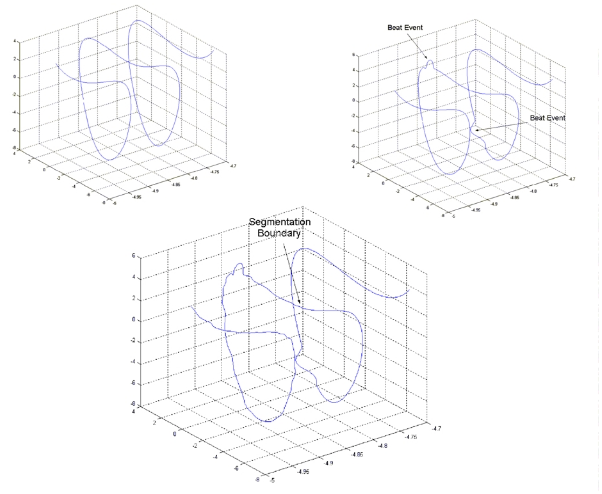

Figure 7: (top-left) A 3D plot of the original motion curve; the path starts on the left-hand-side and finishes on the right-hand-side. (top-right) Plot of perturbed motion using the beat data. (bottom) Plot of resultant motion to which noise was blended in up until the segmentation boundary.

7. Future Work and Conclusion

Linking animations to their corresponding music is an important part of an effective animation. We have outlined a system for integrating motion and sound. The key to this approach is to use various music analysis techniques on both MIDI and audio to guide the motion editing process. This allows users to easily combine different aspects of the music with different aspects of the motion.

Initially, focus of the work was on implementing a selection of MIDI-based features of the music. Metrical analysis, chord recognition, ad-hoc features, simple segmentation and exact pattern matching has been investigated. As yet, beat induction is the only feature supplied by the audio analysis. A number of motion editing techniques, as well as a flexible GUI, need to be implemented in order for us to extensively test multiple combination of music and motion parameters. This will then help us eliminate unconvincing combinations.

Finally, we hope to provide a set of specially designed motion transformations in order to take advantage of human motion capture. For example, a shoulder swinging or head nodding motion could be added to the animated character.

10. References

[1] Mark Bolas Christian Greuel. Sculpting 3d worlds with music. *IS and T/SPIE Symposium on Electronic Imaging: Science and Technology*, February 1996.

[2] A. Bruderlin and L. Williams. Motion signal processing. *SIGGRAPH 1995*, August 1995.

[3] Emilios Cambouropoulos. Extracting significant patterns from musical strings. *In Proceedings of the AISB'99 Convention (Artificial Intelligence and Simulation of Behaviour),* 2000.

[4] S. Coquillard, A control-point-based sweeping technique, *CAD*, 1997

[5] P. Desain and H. Honing. Computational models of beat induction: the rule-based approach. *Journal of New Music Research*, pages 1–10, 1995.

[6] T. Frank, M. Hoch, and G. Trogemann. Automated lip-sync for 3d-character animation. *15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*, August 1997.

[7] David Gerhard. Audio Signal Classification. *Phd thesis*, School of Computing Science, Simon Fraser University, 2000.

[8] Jounghyun Gerard and Jane Hwang. Musical motion: A medium for uniting visualization and control of music in the virtual environment. *Int. Conference on Virtual Systems and Multimedia*, 1999.

[9] J. Hahn, H. Fouad, L. Gritz, and J. Lee. Integrating sounds and motions in virtual environments. Sound for Animation and Virtual Reality, *SIGGRAPH 95 Course n.10 Notes*, 1995.

[10] J. K. Hahn, J. Geigel, Jong Won Lee, L. Gritz, T. Takala, and S. Mishra. An integrated approach to motion and sound. *The Journal of Visualization and Computer Animation*, 6(2):109–124, 1995.

[11] J. Huopaniemi, L. Savioja, and T. Takala. Diva virtual audio reality system. *Proc. Int. Conf. Auditory Display (ICAD'96)*, pages pp. 111–116, 1996.

[12] K. Kashino and H. Tanaka. A sound source separation system with the ability of automatic tone modeling. *Proceedings of the 1993 International Computer Music Conference*, pages pp. 248–255, 1993.

[13] J. Lasseter, Principles of Traditional Animation Applied to 3D Computer Graphics, *SIGGRAPH*'87, 1987

[14] Peter C. Litwinowicz. Inkwell: A 2 1/2-d animation system. *SIGGRAPH 1991 Proceedings*, 25:pages 113–122, July 1991.

[15] H. Longuet-Higgins and C. Lee. The perception of musical rhythms. *Perception*, 1982.

[16] W. Lytle. More bells and whistles [video]. In *SIGGRAPH'91 film show*, 1994.

[17] W. Lytle. Driving computer graphics animation from a musicalscore. *Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers*, 1990.

[18] S. Mishra and J. Hahn. Mapping motion to sound and music in computer animation and ve. *Invited Paper, Proceedings of Pacific Graphics*., 1995.

[19] Jun ichi Nakamura, Tetsuya Kaku, Tsukasa Noma, and Sho Yoshida. Automatic background music generation based on actors' emotion and motions. *In First Pacific Conference on Computer Graphics and Applications*, 1993.

[21] Ken Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1:pages 5–15, March 1995.

[22] Ken Perlin and Athomas Goldberg. Improv: A system for scripting interactive actors in virtual worlds. *Computer Graphics*, 30(Annual Conference Series):205–216, 1996.

[23] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5), 1998.

[24] Robert Rowe. *Interactive Music Systems*. The MIT Press, 1993.

[25] A. L. M. Zs Ruttkay. Chartoon 2.1 extensions; expression repertoire and lip sync. *Technical Report INS-R0016, Information Systems (INS)*, 2000.

[26] L. Savioja, J. Huopaniemi, T. Lokki, and R. Vnnen. Virtual environment simulation - advances in the diva project. *Proc. Int. Conf. Auditory Display (ICAD'97)*, 1997.

[27] F. Silva, Motion cyclification by time x frequency warping. *In Proceedings of SIBGRAPI'99, XII Brazilian Symposium of Computer Graphics and Image Processing*, pages 49–58, 1999.

[28] Eric Singer Robert Rowe. Two highly integrated real-time music and graphics performance systems. *ICMC*, Thessaloniki, 1997.

[29] D. Temperly, The Perception of Harmony and Tonality: An Algorithmic Perspective, University of Columbia, *PhD thesis*, 1996.

[30] Munetoshi Unuma, Ken Anjyo and Ryozo Takeuchi, Fourier Principles for Emotion-based Human Figure Animation. *SIGGRAPH '95*, 1995.

[31] Adam Woodgain. Visualizing expressive movement in music. *Proceedings of San Francisco IEEE Visualization'96*, 1996.