Haptical Music Player

Alfonso Balandra[1,a)] Hasegawa Shoichi[1,b)] Mitake Hironori[1] Sato Makoto[1]

Abstract : This project aims to translate the most basic music elements from a music MIDI file into an enjoyable and coherent haptic vibration. To achieve this goal a simple entertainment setup was developed. This setup consist of 3 main parts synchronized with the musical piece; a MIDI music player, a simple musical structure animation and an event based haptic vibration. The notes' pitch, timing and duration are used as parameters to generate a real time haptic vibration. In order to do that, an event based haptic vibration is calculated depending on the note's pitch and duration; using a sine damping function. The haptic vibration is displayed on a haptic device, resembling the tactile sensation of the note's sound, all of this in synchrony with the music animation and the MIDI player sound.

1. Introduction

Is not surprising, that almost any normal human can appreciate a musical piece, even without any kind of musical formation. The basic elements that compose any song −like rhythm, harmony, and melody− are integrated by our hearing and brain to create an enjoyable experience. But there is also an inner beauty in the basic musical elements that can be appreciated in the song structure and chords, by people with more musical knowledge; like a fugue or a cannon.

While a simple music animation helps the user to see the structure of the music, the vibration shows a relation with the sound and the properties of the notes that are being played. The main elements of the project: the animation, the vibration and the sound help people novice in music to identify and understand the notes' properties in a music piece. To achieve this goal, an special kind of MIDI player was developed. A MIDI player capable to synchronize, in real time, a music structure animation and a haptic vibration according to the note's timing, pitch and duration.

2. Purpose Clarification

The main purpose of this project is to exchange the music listening experience though haptic vibration; giving the user the opportunity to enjoy and understand the basic elements of music. Consequently our efforts are focused on finding a novel way to represent coherently the note's pitch, timing and duration into a resemble haptic vibration.

While the user hear the musical piece, the note's pitch, timing and duration are used as parameters to compute a event based haptic vibration. Then the resulting force from the calculations is displayed into a haptic device. Consequently, our efforts are focused on finding a mathematical method that translates the basic elements of music into a coherent and enjoyable haptic vibration. In simpler words we are trying to haptic vibration.

3. Personal Motivation

Personally taking, the main motivation to develop this kind of idea came from the Musical Animation Machine [4], a project maintained and developed by Kevin Kelly. The Musical Animation Machine enabled me to understand music in a most basic level without musical or instrument playing knowledge. So, the next step for me was not also to listen and understand the music note patterns though animation, but also exchange the experience feel-

1 東京工業大学, Tokyo Institute of Technology, Suzukakedai Campus, Yokohama 226-8502, Japan
a) poncho@haselab.net
b) hase@pi.titech.ac.jp

ing the corresponding vibration of every note.

Also other motivation was that the main idea seem to be very novel. There are not previous research, internet projects or artistic works that had tried to translate a piece of music in to haptic vibration. There are efforts to improve the experience in music of video games using better hardware [1] [2] or studying the human perception of music [3]. But until now, we haven't found any work that specifically had tried a resembling mapping of music elements into haptic vibration.

4. Methodology

A virtual entertainment environment is setup to improve the music vibration rendering. This environment is composed of 3 main modules: a normal MIDI music player, the 3D music animation and the haptic vibration rendering.

4.1 MIDI Music Player

To play the music a simple MIDI music player was implemented. To speed up the implementation the Open Midi Project C++ library was used. The MIDI files are parsed, handled and played using this library. Also this library provides the necessary mechanisms to synchronize the animation and the haptic rendering, which is necessary to setup the entertainment environment.

The preference was given to MIDI file format because the song information is discrete. In a MIDI file the basic song information directly available for reading or writing. So is not necessary to use any audio signal processing algorithm to separate the tracks and notes of the musical piece.

4.2 3D Music Animation

The music animation of the project is based on the Kevin Kelly's project the Music Animation Machine [4]. The developing of the music animation follow early Kelly's animations example.

To display a animation while the music plays is an important factor, but not fundamental part of this project. The reason of using a music visual animation, along with the music and the vibration, is because it provides information about the musical piece to the user. Information like the song notes' pitch, duration and timing; and also the different tracks that compose the song. Otherwise, if any visual animation is not displayed; the users with insufficient musical knowledge feel confused when they try to match the music with the haptic vibration by their own. The music animation is important tool to enable the most inexperienced users to match the musical events with the haptic events.

The visual music animation was made using OpenGl. Following Kelly's example, the notes of the MIDI track are represented using 3D rectangles. The rectangle length represents the exact length of the note in seconds. The rectangle's position in ***(x)*** represents the note position in the song's time line, the ***(y)*** coordinate represents the note's pitch; high pitch notes are higher than the lower pitch notes, and the ***(z)*** coordinate is used to place the notes of the song's different tracks; in this way the notes of one track will not be occluded by the other. Also multiple colors are used to to represent the different tracks of the song, in other to identify the notes of each track *(see Figure* 1).

Also the concept of present time is very important for this animations. The present events, that are being played by the MIDI player, are always shown at the middle of the screen. So the notes that are going to be played are on the right of the screen, the notes that are being played are in the middle and the notes that had been played are in the left side of the screen. Also the color of the notes changes to yellow, when these are played. To synchronized the animation with the music player the MIDI clock ticks are used to match the movements of the OpenGL camera with the music.

4.3 Hapic Rendering

The haptic interaction with the user is passive, this means that the vibration is displayed to the user, but the user doesn't have any interaction with the music animation. So in this particular case the haptic vibration is modelled using an event-based haptic model, but the haptic pointer velocity is not considered; because the interaction happens only in one direction. *(see Equation 1).*

$$f(t) = -keyamp \cdot e^{(-decay \cdot t)} \cdot sin\left(t \cdot keyfreq\right) \quad (1)$$

The model, shown in the *see Equation 1*, was created by Kuchenbecker [6]. It can be used to simulate differ-

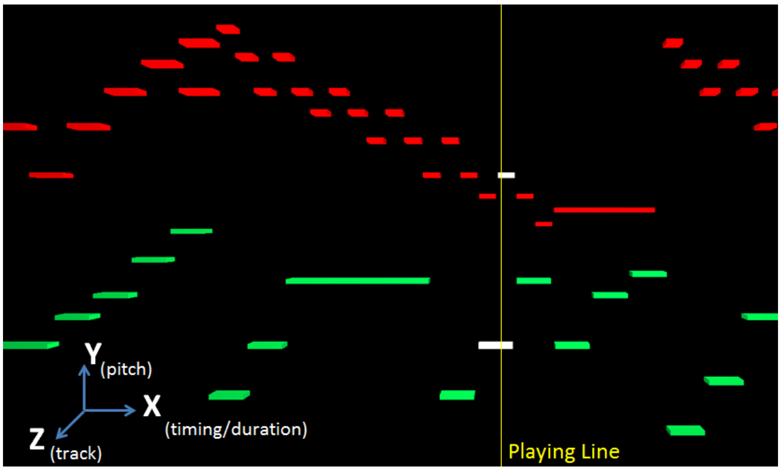

Fig. 1 Music animation while playing a song with 2 tracks.

ent kinds of materials, from soft materials like wood, to hard materials like aluminium. Essentially we use the event-based model to display a lower frequency vibration for the low pitch notes, and a higher frequency vibration for the high pitch notes. In consequence using a very simple direct proportion we map the note pitch to the sine wave frequency and the amplitude of the damping function, and then using the note duration we calculate the decay rate of the function.

To transform the note pitch into haptic vibration, first the lowest and highest pitch note in the track are found. Then, the number of unique notes played between the lowest and highest note in the track are counted. With these parameters we divide the distance between the lowest note and the highest note by the number of unique steps *(see Equation 2)*.

$$steps = \frac{(note_{max} - note_{min}) + 1}{unique\_notes} \quad (2)$$

Then using a simple direct proportion, between the note pitch and the number of normalized steps in the track, a step number for every note pitch is selected with a normalized distance between them *(see Equation 3)*. This step normalizes the spaces between minimum and the maximum pitch in the track.

$$note_{step} = \frac{unique\_notes \cdot (note_{key} - note_{min})}{(note_{max} - note_{min}) + 1} \quad (3)$$

Now that the track note positions had been normalized. So at last to match the frequency and the amplitude of the notes with a representative haptic vibration. Matching the the lowest and highest pitch of the note with a fixed haptic frequency and amplitude between 60Hz 200Hz and 3.8 5.0 respectively. The maximum and minimum range of the frequency and amplitude depend directly on the used haptic interface, so this values may change depending on the haptic interface specifications *(see Equations 4 & 5)* .

$$fdist = (freq_{max} - freq_{min})$$
$$adist = (amp_{max} - amp_{min})$$
$$keyfreq = freq_{min} + (note_{step} * fdist) \quad (4)$$
$$keyamp = amp_{min} + (note_{step} * adist) \quad (5)$$

Once the vibration frequency and the start amplitude are known, the decay rate of the function is adjusted to the note's duration; using the *Equation 6*. So the vibration sensation will end exactly when the note ends to be played, no matter the note duration.

$$decay = \frac{\ln(force_{min}/keyamp)}{time\_lapse} \quad (6)$$

As a result, the vibration frequency and initial amplitude only depend on the note pitch and duration. In other hand, if we take the same notes from two different songs, they will not have the same haptic vibration. This is because the pitch of the notes between the songs is not the same. This happens because the relative distance between the notes pitch is considered to compute haptic frequency.

The result of the calculations are directly displayed into the haptic device. The haptic interface employed to display the feedback force is "SPIDAR", this human-computer interface was developed by Makoto Sato [5]. The "SPIDAR" haptic interface has a 1kHz update frequency, so is possible to have a 1ms sampling rate, which enables us to display a haptic vibration accurate with the mathematical model.

5. Further Work

By now this system is only capable of displaying the vibration of only one track of the song. This is because any further analysis of how to display several tracks at the same time haven't been made. But we don't discard that would be possible to display a resemble vibration from several tracks at the same time.

The developed system just can display the vibration of just one note at the time. But, currently we are working to find a good strategy to display the vibration of two notes playing simultaneously.

参考文献

[1] Bayer Materials, *ViviTouch - "Feel the game"*, http://vivitouch.com/.

[2] Kyota Yamada, Akira Komatsu. *Audio-Band Electromechanical Vibration Conventer*, US Patent, 43454067, (1982).

[3] Bregman, Albert S. *AuditoryScene Analysis : The Perceptual Organization of Sound*, MIT Press, (1994).

[4] Kelly Kevin, *The Music Animation Machine - "Music Worth Watching"*, http://www.musanim.com/.

[5] Sato Makoto, *A 3d spatial interface device using tensed strings*, Presence, (1994), pp. 81-86.

[6] Kuchenbecker J. Katherine, et al. *Improving contact realism through event-based haptic feedback*, IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 2, (2006), pp. 219-230.