A MULTI-TIMELINES SCHEDULER AND A REHEARSAL FUNCTION FOR IMPROVING USERS' SENSATION OF ORCHESTRAL CONDUCTING WITH A CONDUCTING SYSTEM

Takashi Baba
Kwansei Gakuin University
takashi-baba@kwansei.ac.jp

Mitsuyo Hashida
Soai University
hashida@soai.ac.jp

Haruhiro Katayose
Kwansei Gakuin University
katayose@kwansei.ac.jp

ABSTRACT

The VirtualPhilharmony (VP) system conveys the sensation of conducting an orchestra to a user (a conductor). VPs performances are created through interaction between the conductor and orchestra, exactly like real performance. "Concertmaster function has been already implemented by incorporating the heuristics of conducting an orchestra. A precisely predictive scheduler and dynamical template have been designed based on analyses of actual recordings. We especially focused on two more problems to emulate a more real orchestral performance; one was that each note in the template was controlled by single-timeline scheduler; the other was that the interaction and communication between the conductor and the orchestra in repeated practices were not simulated. We implemented "Multi-timelines scheduler and "Rehearsal function to resolve these problems.

1. INTRODUCTION

Musical conducting, which sometimes involves directing an orchestra with as many as 100 members, has been a highly regarded profession since its professional status was established in the 19th century. A conductors manner of musical expression is quite different from that of an instrumentalist or a vocalist. Inspired by this unique manner of creative expression, composers and engineers in the field of computer science started to develop conducting interfaces in the late 1980s. The existing conducting systems have been designed as a means of musical expression through computer-generated music, but they had a problem that is a shortage of viewpoints of the musicality of the performers.

We have been developing a conducting system called "VirtualPhilharmony (VP)" that is intended to convey a realistic sensation of conducting an orchestra. Actual performances are created through interaction between the conductor (the user) and orchestra. Our intent is to simulate this interaction by incorporating into VP the heuristics of conducting an orchestra. We have already implemented "Concertmaster function, in which a precisely predictive scheduler and dynamical template have been designed based on analyses of actual recordings. There were two problems in Concertmaster function; one was that each note in the template was controlled by single-timeline scheduler; the other was that the interaction and communication between the conductor and the orchestra in repeated practices like rehearsals were not simulated. We implemented "Multi-timelines scheduler" and "Rehearsal function" to resolve these two problems. The aim of the former is to control expressiveness for each voice part or musical instrument in real-time. The aim of the latter is to revise the template in accordance with the user's musical intention or habit during the repeated rehearsals.

2. RELATED WORK

The mechanism of a conducting system performs four basic actions: 1) extract the beat points from the users arm and hand motions by using gesture sensors; 2) calculate the tempi from the beat points; 3) predict the tempo of the next beat by using the previous tempi; and 4) schedule the playing of notes on the basis of the predicted tempo.

One of the earliest related systems is the "Radio Baton" [1] developed by Mathews in 1987. The user controls a MIDI [1] by holding two batons over a sensor plate. Although this system was not intended for simulated conducting, the idea of tempo control based on user-given beats is similar to conducting.

Morita et al. simulated some directions to each musical instrument using user's left arm in conducting [2]. The position of the left arm was recognized by a camera, dynamics of the instruments pointed by the left arm were separately controlled.

"iFP" [3] by Okudaira et al. is a performance interface for making keyboard strokes or hand-shaking gestures. It extracts beat information and feeds it into an expressive performance template, where the expression data extracted from virtuoso piano performances are described. The template file describes deviations regarding dynamics, tempo transitions, and delicate controls within a beat of each note. By borrowing such expressiveness from real performances, iFP enables a user to add delicate nuances within a beat. It also provides parameters to balance the intensions of the user and the template. VP is based on iFP.

"Personal Orchestra (now "The Virtual Conductor")" [4] by Borchers et al. is an audio-based conducting system. It uses a phase vocoder to stretch the audio signal. A big

[1] Musical Instrument Digital Interface

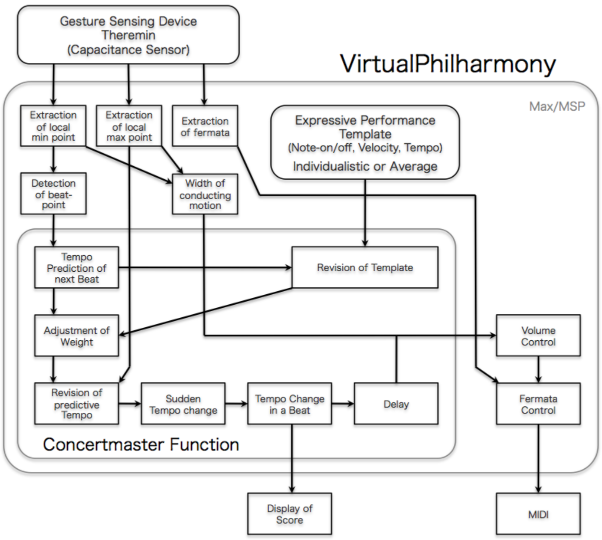

Figure 1. Overview of existing VP

advantage of this system is that the templates were constructed from recorded performances by the Vienna Philharmonic. The user can thus enjoy the sensation of conducting a real orchestra together with video images of the performance.

3. OUTLINE OF VP

3.1 Outline of existing VP

Figure 1 shows an overview of existing VP [5]. VP extracts the local minimum points and local maximum points by using a gesture sensor that detects the user's conducting motion. It detects the beat points and dynamics from the detected data. The user's tempo has an interval of two beat points. The Concertmaster function predicts the tempo of the next beat, revises the template, and schedules the playing of notes using the user's tempo and template tempo. VP controls the velocities of each scheduled note on the basis of the volumes detected from the sensor. Finally, VP broadcasts the musical performances from the speakers and displays the score. VP is implemented in the Max/MSP visual programming language (Cycling '74 [2]).

3.1.1 Gesture sensing device

An Etherwave theremin [3] (Moog) is used for detecting the user's conducting motions (Figure 2). A theremin is an electronic musical instrument that controls the frequency (pitch) and amplitude (volume) of an emitted sine wave on the basis of the changes in the capacitance between the user's arm and the theremins antenna. We used a theremin because it has high time resolution and high space resolution. The more precise the time resolution, the more precise the beat-point detection. The accelerometer sensor and CMOS sensor of the Wii Remote have time resolutions of 10 [ms]. The theremin's time resolution is 5 [ms].

[2] http://cycling74.com/
[3] http://web2.moridaira.com/Moog/theremin.htm/

Figure 2. Etherwave Theremin

3.1.2 Expressive performance template

The expressive performance template in VP is in MusicXML [4] format, which describes the score information, and in DeviationInstanceXML [5] format, which describes the deviation information. The deviation information used in VP is 1) the attack deviation for all notes, 2) the release deviation for all notes, 3) the volume deviation (velocity) for all notes, and 4) the tempo deviation for all beats. The template is a standard MIDI file (SMF) based on actual recordings. One to ten versions of the template are prepared for each musical piece.

3.1.3 Concertmaster function

The Concertmaster function predicts the tempo of the next beat on the basis of the tempi of beats detected from the user's conducting motion using a sensor and interactively schedules the playing of the notes in the template on the basis of the predicted tempo. The Concertmaster function performs the role played by a human concertmaster, who conveys the instructions of the conductor to the orchestra members and conveys the intentions of the members to the conductor. It does this by balancing the intentions of the user with the requirements of the template through the use of seven sub-functions: 1) predict tempo of next beat, 2) revise expressive performance template, 3) adjust interaction's degree, 4) revise predicted beat using local maximum point in user's conducting motion, 5) provide support for sudden tempo changes, 6) provide support for tempo changes in a beat and 7) adjust for delay between conducting motions and musical performance. We explain 1), 2), and 3) in the following sub-subsections.

Tempo prediction of the next beat

Schedulers in conventional conducting systems were controlled by beats. The tempo of the next beat was predicted from the tempi of the previous beats by using a moving average or linear prediction. Schedulers based on linear predictive coding (LPC) had a constant coefficient, a_k, and a constant coefficient degree, N. As a result, they could not make predictions like those made by human music players. The effect of the previous beats on the predicted beat in real performances is not constant due to variations in music style (e.g., waltz or march), beat time, and so on. A typical example is a Viennese waltz (triple time) in which the tempi of the second beat

[4] http://www.recordare.com/musicxml/
[5] http://www.crestmuse.jp/cmx/

are lengthened. This is a cue to the dancers to turn on the second beat. Such heuristics should be reflected in the parameters of the LPC, and N and a_k should be varied in accordance with the music style or beat time. We examined the effect of previous beats by analyzing actual recordings.

Revision of expressive performance template

The scheduler of iFP uses the following equation

$$P_{n+1} = \alpha A_{n+1} + \beta B_{n+1} + \gamma C_{n+1} \qquad (1)$$

where A_{n+1} is the moving average of the previous four beats, B_{n+1} is the difference between the current and previous tempo, C_{n+1} is the tempo indicated by the template, P_{n+1} is the predicted tempo of the next beat, and α, β and γ are the weighting coefficients for each variable ($0 \leq \alpha, \beta, \gamma \leq 1$). A_{n+1} and B_{n+1} are for the user side, and C_{n+1} is for the orchestra (pianist) side. In iFP, the balances among these weight parameters characterize the performance, i.e., the interaction between the user and template representing the virtual orchestra. However, the weight parameters are static, i.e., independent of the music style and tempo. A user may thus feel stress when his or her tempo differs from the template tempo. The weight parameters should therefore be adjusted to fit the music style of the user, and the template should be dynamically revised in accordance with the tempo. There are three sorts of revision: 1) revision of the tempo per beat, 2) revision of the beats in a Viennese waltz, and 3) revision of the rhythm pattern of the dotted notes. As for 3), although the ratio of playing time in a rhythm pattern of dotted notes is $3 : 1$, if a user plays such notes in accordance with the score, experience has shown that the tempo is faster; that is, the rhythm of the dotted notes approaches that of a triplet. The rhythm pattern of the dotted notes in the template is revised by using the predicted tempo.

Adjustment of interaction's degree

The scheduler in iFP is based on equation (1). In case of VP, the predicted tempo of the next beat P_{n+1} is decided by

$$\begin{aligned} P_{n+1} &= \alpha^{cond} A^{cond}_{n+1} + \beta B_{n+1} \quad \text{(user)} \\ &+ \alpha^{temp} A^{temp}_{n+1} + \gamma C'_{n+1} \quad \text{(template)} \end{aligned} \qquad (2)$$

where A^{cond}_{n+1} is the predictive tempo of the next beat based on the user's tempo, B_{n+1} is the latest tempo T_n, A^{temp}_{n+1} is the predictive tempo of the next beat based on the template tempo, C_{n+1} is the revised template tempo, and α^{cond}, α^{temp}, β and γ are the weighting coefficients for each variable. A^{cond}_{n+1} and B_{n+1} are the tempi belonging to the user side. When the user increases the weight of A^{cond}_{n+1}, the influence of the past tempo in the predictive scheduler increases; when the user increases the weight of B_{n+1}, the system closely follows the user. On the other hand, A^{temp}_{n+1} and C_{n+1} are the tempi belonging to the template side. When the user increases the weight of A^{temp}_{n+1}, the influence of the past tempo in the template increases; when the user increases the weight of C_{n+1}, the system ignores the user.

3.2 Evaluation of existing VP

We have conducted three evaluations of existing VP: 1) a quantitative evaluation of the Concertmaster function, 2) interviews with expert conductors, and 3) a general evaluation at conferences. We describe 2) and 3) in the following sub-subsections.

3.2.1 Interviews with expert conductors

Since evaluating VP quantitatively is impossible, we interviewed expert conductors, asking them for their impressions of VP after they had used it. The process was as follows: 1) the first author explained VP and the role of the Concertmaster function to the conductor, 2) the author demonstrated its use, 3) the conductor used VP, and 4) the conductor subjectively evaluated and tuned the parameters to what he/she felt to be the best values.

A dozen conductors participated, from Mr. Y. Yuasa, who is a professor at the University of Music and Performing Arts in Vienna, to Mr. M. Okochi, who is a conductor. A common criticism was the inability to change the tone color. One participant said, for example, "One of the significant roles of a conductor is to show differences of musical nuance by subtle movements of his/her hand. If VP can simulate them, it will be able to contribute to the actual practice of conducting." We are thus considering using more than one sensor to control the envelope of volume and tone color.

3.2.2 Exhibitions and showcases

We distributed a questionnaire to the thousands of people who used VP and received responses from 219 of them. The most frequent comments were about adding game elements. That is, they wanted to see computer-graphic characters synchronized with the music instead of a displayed score. While doing this could broaden the potential market, the most significant thing is to display the score for the conductor.

VP has been demonstrated at Interaction 2010 and EC 2011 in Japan and at NIME 2010 and ACE 2010. VP was awarded the Silver Award in the Creative Showcase at ACE (International Conference on Advances in Computer Entertainment Technology) 2010 and was awarded the Rendering Award at Rencon 2011. The first author entered an interactive section and took second place.

4. MULTI-TIMELINES SCHEDULER AND REHEARSAL FUNCTION

4.1 Toward improving user's sensation of conducting

4.1.1 Multi-timelines scheduler

Existing VP and the other conventional systems control all the musical instruments and orchestral parts on a single timeline. This approach is considered appropriate because one of the roles of a conductor is to unify the playing of the orchestra members on one timeline. However, this approach ignores the timelines the individual members have as a performance model. Although the members revise their timelines to fit the common timeline dictated by the conductor, their revised timelines do not completely match the common one. This gap yields an interaction between the conductor and the orchestra. A separate timeline must thus be designed for each orchestra member. In the case of homophony in Western classical music, the timeline for each member is commonized with a voice part (melody part, accompanying voice part, bass part, and so on). Therefore, a timeline should be prepared for each voice part. VP needs to control expressiveness for each voice part in real-time by the Multi-timelines scheduler.

"jPop-E (Polyphrase Ensemble)" [6] by Hashida et al. was the rule-based system that renders a natural expression of multiple music. It could independently control multiple voice parts. In order for a timing lag between some parts in the same area of the score to become too large, the synchronization points that synchronize all parts by force were provided every a break between phrases, and the tempo for each voice part was linearly revised so that all parts become the same timing in this point. This system could not control expressiveness for each voice part in real-time.

4.1.2 Rehearsal function

During the practices and rehearsals before a public performance, the conductor and orchestra coordinate with each other on precise musical expressions and directions by repeatedly practicing the performance and by oral communication. This interaction results in a better performance. With existing VP and the other conventional systems, a user would hesitate to play a public performance due to uncertain about the template (orchestra) and would thus likely give an awkward performance. To give a better performance, the user must be familiar with the template, and the template (computer) must be familiar with the user. The user achieves this by repeatedly playing a musical piece with VP, so that VP learns the users template little by little. That is, the computer learns the users musical intentions and habits and reflects them little by little. We simply simulate the practice session interactions.

"Synthetic Rehearsal" [7] by Vercoe et al. was a example that refers to the interaction during the repeated practices. Its aim was an ensemble between a live performer and a synthetic performer (computer model). The model learned the live performer's intention by repeating rehearsals, and transition of whole tempi in a template was revised in accordance with the live performer's tempi. However, this system could not control delicate expression within a beat, and macro-annotations such as *crescendo* and *ritardando*. In addition, it is controlled by single-timeline scheduler. So, we need to expand the concept of the Synthetic Rehearsal for interaction between a conductor and an orchestra.

4.2 Implementations

4.2.1 Multi-timelines scheduler

In VP, we introduce a timeline for each voice part as follows. 1) We manually divide the template into the voice parts. If there are two or more roles in one part, that part is subdivided into separate templates. For example, a cello sometimes plays melody with an oboe, and sometimes a bass plays melody with a contrabass. 2) VP simulates the interaction between the user and each template by using the Concertmaster function and generates a separate timeline for each template. 3) If the timelines are too far apart, they should be revised to the user's (conductor's) timeline little by little.

As for 1), information of the tempo deviation for all beats in XML file was used for the template tempo in existing VP. So, the tempo deviations plus the attack deviations for each voice part form each tempo curve.

As for 3), each voice part has a different threshold for the revision when the timeline are too far apart from another timeline. The threshold τ is

$$\tau = \frac{2(t^{v_A} - t^{v_B})}{P^{v_A}_{n+1} + P^{v_B}_{n+1}} \quad (3)$$

where t^{v_X} is the time of the next ((n+1)th) beat in X part, and $P^{v_X}_{n+1}$ is the predictive tempo of the next beat in X part.

$$P^{v_X}_{m+1}{}' = \begin{cases} if(\tau > 0.1)\ then & \\ \alpha P^{v_X}_{m+1} & (\alpha > 1) \\ if(\tau < -0.1)\ then & \\ \beta P^{v_X}_{m+1} & (\beta < 1) \end{cases} \quad (m \geq n) \quad (4)$$

where P is the predictive tempo of the next beat for each voice part, P' is revised one, and α and β are the weighting coefficients. If a part is too slow than the other part, the predictive tempi after the (n+1)th beat speed up, and vice versa. Because there are many cases where a melody part is independent of the other part, we heighten the threshold of the melody part than the others ($\tau > 0.3$ or $\tau < -0.3$). In addition, we manually decide the synchronization point that synchronizes all parts by force based on actual recordings. The points were provided every a break between phrases; all timelines are linearly revised towards the point. The user can freely reduce the default synchronization points and add them, by drawing the points in the score on a monitor with a pen tablet. Because the predictive tempi are revised for each beat using equation (3) and (4), VP can delicately control the scheduler than the Pop-E that only revised the tempi in the synchronization point.

4.2.2 Rehearsal function

There are the following four steps in VP's rehearsal; 1) drawing some signs in the score, 2) playing and recording a piece, 3) reflecting the user's intention or habit to the template, and 4) repeating from 1) until 3).

In step 1, the user can draw two kinds of signs in the score on a monitor with a pen tablet: the signs of divisions or omissions of the conducting beats, and four musical symbols, those are, *crescendo*, *diminuendo*, *accelerando*, and *ritardando*. The conductor sometimes shakes a baton

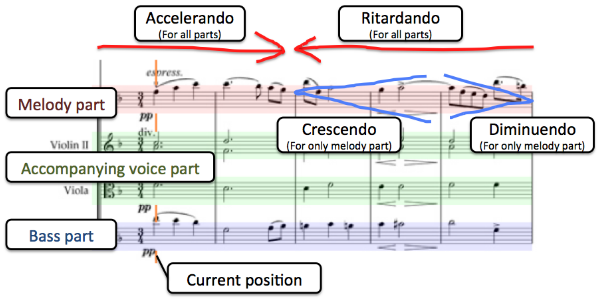

Figure 3. Drawing in a Score

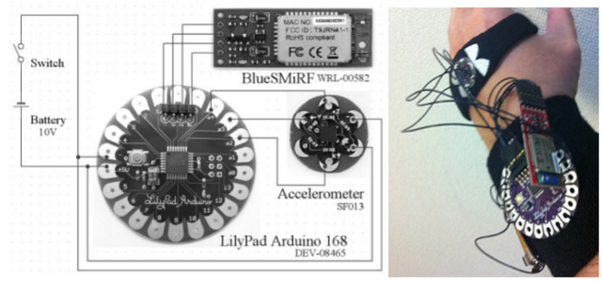

Figure 4. A circuit and a picture of the glove interface for left hands

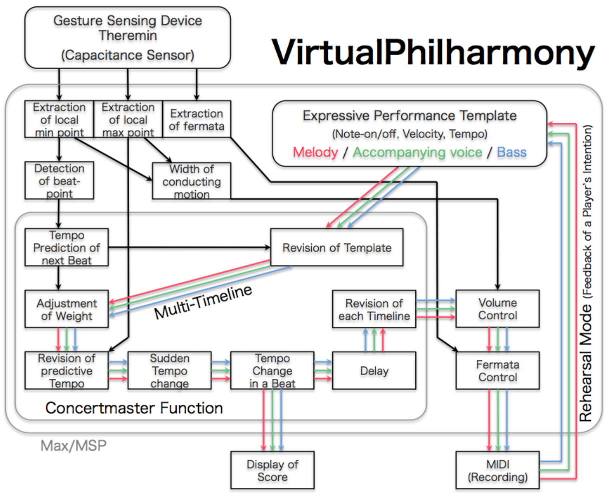

Figure 5. Overview of VP

with an up-beat after the preceding down-beat such as *ritardando*, and also does not shake it when the same tempo continues. Since default signs are drawn in the VP score in the beat points where the user should shake it, the user disappears them for the omission or add new signs for the up-beat in the score. The conductor also draws various musical symbols in the score, especially, the above four symbols are drawn very often. There are cases where these symbols are redrawn to emphasize the printed symbols in the score, and others where the conductor adds to draw them. In Rehearsal function, the user can draw the symbols as shown in Figure 3). A voice part, the starting point and the ending point are decided from the position (x, y) of the symbol that the user drew in the score. When there are some printed symbols under the symbol that the user drew, the changes of the templates between the relevant parts are emphasized. When the user added new symbols, revised values based on a beta curve are added to them. In case of *accelerando* and *ritardando*, the tempi data in the template after the symbol are all revised to the last tempo in the symbol.

In step 2, VP records an user's performance. If the user does not like this performance, he/she can cancel it.

In step 3, the weighted average of the deviations for each note in the recorded performance and the template is updated using the following equation as new deviations in the template;

$$i_{newtemp} = \alpha i_{rec} + \beta i_{temp} \qquad (\alpha + \beta = 1) \qquad (5)$$

where i_{rec} is a recorded note, i_{temp} is a note in the template, $i_{newtemp}$ is a note in the uploaded template, and α and β are the weighting coefficients. The weighting coefficients are heuristically changed on the basis of the deviation from the current template.

In step 4, the user repeats from step 1 until setp 3. He/she can erase the signs that were drawn by him/herself.

4.2.3 *Conducting using user's left arm*

Roles of a conductor's left arm in actual conducting are very various, for example, the case where the conductor synchronizes the motion of the his/her left hand with the right hand, he/she directs dynamics for each voice part, cues some voice parts, and so on. Since it is impossible to extract all these information using any gesture sensors, we need to limit use. In VP, we simulated volume control for a specific voice part.

Dynamics for each voice part is controlled by an original glove interface using an accelerometer sensor (Figure 4). The LilyPad Arduino is used for the accelerometer sensor and the microcontroller. The three-dimensional acceleration information with a sampling rate of 100 [Hz] is sent to the system through Bluetooth using a BlueSMiRF. The sensor is sewed on the side of the back of the left hand on the glove; there is a switch in the side of the palm of the left hand. The switch is on when the user clenches his/her fist, and it is off when he/she opens his/her hand. The sensor data is sent to the system when the switch is on. If he/she turns the palm of his/her hand upward, the velocity is increased, and vice versa.

The position of the left arm is detected by Kinect (infrared camera). When the switch is on, the Kinect shoots the range of 225 [cm] height by 300 [cm] width centered around the conductor (user) with 30 [fps]. VP detects the position of the left hand, and collates the captured position with the position of each voice part, which is manually set (e.g. A part of the first violin is on the conductor's right.).

The sampling rates of these sensors are rougher than the one of the theremin. Because they are used to detect rough position and motion of the users left arm, they do not need a latency of 5ms in the theremin.

Finally, we show overview of VP in figure 5.

5. DISCUSSION

5.1 Subjective evaluation by the developer

I (the first author) conducted subjective evaluations of the Multi-timelines scheduler and the Rehearsal function.

I played "Cavalleria Rusticana - Intermezzo" by P. Mascagni. Since this piece has a homophonic structure, it has one melody part. And it has a slow tempo. It is crucial in this piece to play the melody part slowly and carefully. In the first performance, I confirmed that the user could control the melody part independent of the other parts. Secondly, I changed from the default synchronization points to the original ones by erasing and adding some points; I sensed unreality when I put the point in the middle of a phrase. Thirdly, I used the Rehearsal function only once. At this time, I drew nothing in the score. Because I played the piece with faster tempi than the template's tempi, the revised template's tempi were a little faster. Fourthly, I drew various musical symbols in the score. In case of *crescendo*, a beta function curve makes the most natural expressiveness; in case of *ritardando*, a logarithmic curve makes the best performance. Finally, I repeated the rehearsals seven times, and drew a lot of symbols in the score. Eventually, I obtained a sensation of orchestral conducting with the template that sufficiently reflected our musical intention and habit.

We should quantitatively evaluate the multi-timelines scheduler and the rehearsal function, and ask expert conductors for their impressions about these.

5.2 Exhibitions and showcases

VP was demonstrated at EC 2011 and Seminar about visualization in Osaka, Japan. We obtained comments that the Rehearsal function was difficult to use because the time for its demonstration was too short. We have thus prepared a short video explaining the rehearsal function. Several musicians commented that the multiple timelines are very useful for musical education and entertainment.

5.3 Future works

One of the significant features of VP is the simulation of the interaction between the conductor and the orchestra. Through this interaction, VP provides a user with the sensation of conducting a real orchestra. There are two important aspects of our work: 1) we incorporated the heuristics of conducting an orchestra and developed performance models of an orchestra; and 2) we create a timeline for each voice part. Our future work will be based on these aspects.

1) VP's performance models are bottom-up models incorporating the heuristics of conducting an orchestra. However, they are not comprehensive. We need to construct top-down performance models by using mathematical models, e.g., HMM. Furthermore, we plan to construct non-parametric models for the more comprehensive models.

2) VP is a MIDI-based conducting system. Although this enables us to control each note very precisely, the sound is poor. In contrast, with an audio-based conducting system, the sound is very realistic, but there is little control of each note. Since we developed a timeline for each voice part in VP, we can apply this to an audio-based system, which will give us more control of the notes. We plan to divide a signal of a tune into voice parts by using sound source separation, to extract a timeline for each part by using beat tracking, and to control each separated signal on the basis of its timeline.

6. CONCLUSION

We described the VirtualPhilharmony (VP) conducting system, which conveys the sensation of conducting an orchestra. VP provides a user with a realistic sensation through its incorporated performance models of orchestra members and simulates the interaction between a conductor and an orchestra on the basis of these models. A main function was incorporated into VP on the basis of the heuristics of conducting an orchestra: Concertmaster function. To improve user's sensation of orchestral conducting, we implemented Multi-timelines scheduler and Rehearsal function. The former enabled the user to control each voice part independently; the latter enabled him/her to reflect his/her musical intention and habit.

We are improving VP so that it will be able to create even more genuine and realistic conducting performances. We believe that VP is useful for entertainment, musical education, and musical therapy as well as for actual practice of conducting.

7. REFERENCES

[1] "Review of Research in Computers and Music." ICMA Video Review, 1991.

[2] H. Morita, S. Hashimoto, and S. Ohteru, "A Computer Music System that follows a Human Conductor," *Computer - Special Issue: Computer-generated Music Archive*, vol. 24, no. 7, pp. 44–53, 1991.

[3] K. Okudaira and H. Katayose, "iFP A Music Interface Using an Expressive Performance Template," *Entertainment Computing 2004, Lecture Notes of Computer Science*, vol. 3166, pp. 529–540, 2004.

[4] J. Borchers, E. Lee, W. Samminger, and M. Muehlhaeuser, "Personal orchestra: a real-time audio/video system for interactive conducting," *Multimedia Systems*, vol. 9, pp. 458–465, 2004.

[5] H. M. Baba, T. and H. Katayose, "A Conducting System with heuristics of the conductor "VirtualPhilharmony"." Proc. NIME2010, 2010, pp. 263–270.

[6] N. N. Hashida, M. and H. Katayose, "jPop-E: an assistant system for performance rendering of ensemble music," in *Proceedings of the 7th international conference on New interfaces for musical expression*, 2007, pp. 313–316.

[7] B. Vercoe and M. Puckette, "Synthetic Rehearsal: Training the Synthetic." ICMC '85 Proceedings, 1985, pp. 275–278.