

CONDUCTING A VIRTUAL ENSEMBLE WITH A KINECT DEVICE

A. Rosa-Pujazón, I. Barbancho, L. J. Tardón, A.M.Barbancho

Dpt. Ingeniería de Comunicaciones, E.T.S.I. Telecomunicacion

Universidad de Málaga, Campus de Teatinos s/n, 29071 Málaga, Spain

alejandr@uma.es, ibp@ic.uma.es, lorenzo@ic.uma.es, abp@ic.uma.es

ABSTRACT

This paper presents a gesture-based interaction technique for the implementation of an orchestra conductor and a virtual ensemble, using a 3D camera-based sensor to capture user's gestures. In particular, a human-computer interface has been developed to recognize conducting gestures using a Microsoft Kinect device. The system allows the conductor to control both the tempo in the piece played as well as the dynamics of each instrument set independently. In order to modify the tempo in the playback, a time-frequency processing-based algorithm is used. Finally, an experiment was conducted to assess user's opinion of the system as well as experimentally confirm if the features in the system were effectively improving user experience or not.

1. INTRODUCTION

Computers have become an extremely common tool in our everyday-life, to a degree that we are constantly interacting with them. Yet, standard human-computer interfaces show their shortcomings whenever trying to emulate interaction metaphors that do not naturally map easily to a keyboard-mouse setting, such as, for example, musical instrument simulation. However, the evolution of sensing and motion-tracking technologies has allowed for the development of new and innovative human-computer interfaces that improve user experience towards a more 'natural' interaction paradigm, thus bringing a new and vast array of computer-generated applications that fit much more accurately their real-life counterparts.

With regards to interactive music applications, these advanced human-computer interfaces have been used for a wide array of fields: new instruments creation/simulation [1], body motion to sound mapping [2] [3] [4] [5], gaming [6] [7], modification of visual patterns by using sung or speech voice [8], tangible and haptic instrument simulation [9] [10], drum-hitting simulation [11] [12] [13], etc.

One example of musical performance that is inherently linked to human body motion is that of the orchestra conductor, yet surprisingly there are only a handful of studies that address conducting simulation through the use of advanced human-computer interfaces. Conducting is re-

quired to coordinate and synchronize the performance of an ensemble. Therefore, the conductor must indicate musical parameters such as dynamics or tempo, using his hand and baton gestures to such purpose. Previous research has focused on capturing the conductor's hand or baton motion through the use of infrared sensors [14] [15] [16] [17], inertial sensors [18] or the Wiimote [19] [20], changing the tempo of the pieces performed accordingly. Some studies have also added some form of dynamics control [15] or heuristics [21] to provide a more satisfying user experience.

In this paper, we aim to present a new interaction paradigm for conducting gesture capturing, so that the user can effectively conduct a virtual orchestra, indicating the tempo and beat times of the piece performed, the overall dynamics and the specific volume levels for a concrete set of instruments in the ensemble. In order to achieve this, a Kinect sensing device is used, thus providing an inexpensive and off-the-shelf alternative for a non-intrusive experience for the user, as well as the possibility of tracking both user hands simultaneously. Additionally, we have conducted an experimental study to assess the usefulness of the application developed, as well as to find potential ways to further improve user experience.

The technical details of the system implemented will be discussed in the next section. Then, the following section will cover the details of the experiment performed, whose results will be presented and discussed in the subsequent sections. The conclusions drawn from the study will be depicted in the last section.

2. SYSTEM DESCRIPTION

As previously indicated, we opted for a Microsoft Kinect for XBOX device in our human-computer interface design. Kinect camera offers both a RGB-image and an depth image of the scene capture. By combining the depth map data with the OpenNI library and the NITE plugin, it is possible to extract 3D information to create an skeletal joint model to follow user movements. Concretely, the system is able to track the position in 3D space of up to 15 nodes or joints, corresponding to the head, neck, torso, hands, feet, etc. of the user. To provide some form of visual feedback, the application rendered a basic virtual environment coded in C/C++ using the OpenGL graphics library [22] and the OGRE graphics engine [23]. The environment was textured and modelled to resemble a concert hall (see figure 1). PortAudio library is used for sound management, and the tracks for each of the sets of instruments in the virtual



Figure 1. Virtual environment for the application

ensemble are stored and read from separate WAV files. In order to adequately implement an ensemble conductor simulator, there are two main problems that need to be addressed: how to translate conducting gestures to changes in the performance, and how to smoothly change the tempo of the piece being played.

2.1 Changing the tempo of the piece: time-stretching

In order to change the tempo of the song being played, it is necessary to modify the playback speed according to the rate indicated by the conductor. Nevertheless, due to the duality of time and frequency domains, simply changing the playback speed also has an undesired effect in the form of changes in pitch of the music played. Thus, a slower playback time will result in a decrease of the pitch, while a faster one will make the pitch go higher. In order to smoothly play a musical piece at different playback speed without such pitch changing artifacts, it is necessary to resort to a time-stretching algorithm.

Time-stretching algorithms can be typically implemented in time-domain, using the so-called Synchronous Overlap-and-Add algorithm or SOLA [24]. This algorithm consists in dividing the data signal into successive segments, and then adding these segments together with a certain overlap, as can be seen in the figure 2. The overlapping is performed so that the last part of each segments "fades-out", while the the first part of the next segment "fades-in", taking into account the cross-correlation of both overlapping sections to maximize the smoothness in the transition.

SOLA time-domain algorithm provides a computationally fast time-stretching alternative. However, the main problem with the SOLA algorithm is that it works at its best with structurally simple signals (such as speech [25]), but not so well with polyphonic data, and the time-stretching range is more limited. In order to achieve better quality output, it is necessary to resort to more complex techniques.

In particular, we have implemented a time-stretching algorithm based on the phase vocoder [24]. The phase vocoder is a time-frequency processing technique that uses short-time Fourier analysis and synthesis to transform a given

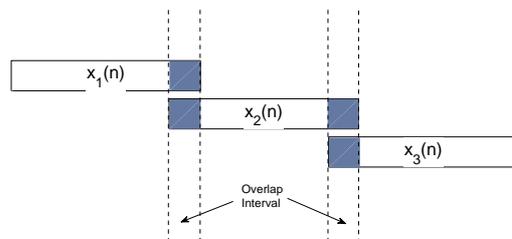


Figure 2. Time-stretching in time-domain

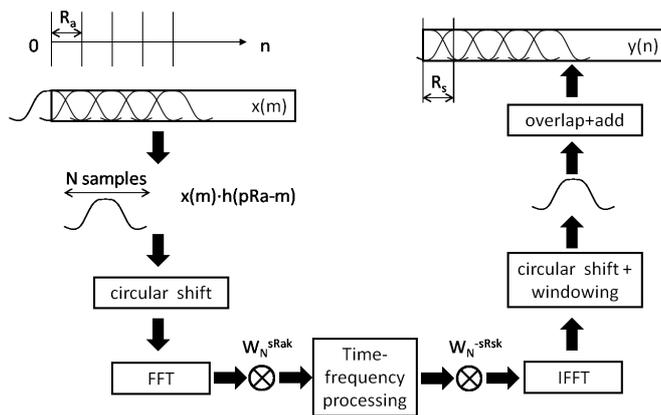


Figure 3. FFT/IFFT block implementation for phase vocoder

data signal, according to the equation of the short-time Fourier transform (STFT) with a window $h(n)$,

$$X(n, k) = \sum_{m=-\infty}^{\infty} x(m)h(n - m)W_N^{mk}$$

$$k = 0, 1, \dots, N - 1, \quad W_N = e^{-j2\pi/N}$$

Working with this equation, it is possible to implement the phase vocoder using two different models [24]: the filter bank summation model and the block-by-block analysis/synthesis model. We have implemented our time-stretching algorithm following the latter. This model is based around the use of the fast fourier transform and its inverse (FFT/IFFT), dividing the input signal in overlapping segments (using a hop-size R_a). The FFT is performed then on each segment, as well as additional transformations to ensure phase coherency. After that, the data is processed as desired, and the output signal is synthesized using the inverse procedure, combining the successive processed segments by an overlap and add method with hop-size R_s . This process is portrayed in figure 3.

In order to implement time-stretching with a phase-vocoder, the hop-sizes for analysis and synthesis (R_a and R_s) are selected accordingly to the time-stretching factor desired (R_s/R_a), and the phase value for each frequency bin is adjusted accordingly [24]. In our application, the timestretching value ranged from 0.5 to 2.0.

2.2 Gesture recognition and interpretation

In order to conduct a given performance, an orchestra conductor gives a series of indications to the musicians by waving his hands, signalling the beat times, indicating the entry points, and controlling the dynamics of the whole ensemble. The role of the conductor is critical, as he is responsible for providing "expressiveness" to the performance. However, for a naïve user, giving the precise indications that a real conductor would give to the ensemble can easily become a daunting and nearly impossible task. For that reason, for the purpose of this application, we have focused on developing a gesture recognition system that can be used by expert and lay users alike. In particular, since the full-body skeleton-tracking functions of OpenNI/NITE provide the 3D coordinates for both hands of the user, we have assigned different functions to the right and left hand gestures respectively. Concretely, right-hand gesturing controls the tempo of the performance as well as the time positions of the beats, while left-hand gesturing controls the dynamics in the performance as well as the volume of each of the instruments taking part in the ensemble.

2.2.1 Conducting tempo

Beat times are indicated by moving the right hand in an horizontal waving motion; this gesture motion was selected as it was found in previous tests with users without musical background showed that they tended to wave their hands horizontally when asked to make conducting gestures, and, in general terms, users reported to be more comfortable with a horizontal motion rather than a vertical one when signalling high tempo. The system marks the time instant when the hand starts and stops a move (be it from left to right, or from right to left), and the time difference is then taken to calculate the new indicated conducting tempo in beats per minute. This time difference is then compared to the original tempo of the piece being played, thus calculating the time-stretching factor (T_s) that must be applied.

$$T_s = \frac{\text{beatsPerMinuteOriginal}}{\text{beatsPerMinuteIndicated}} \quad (1)$$

However, the system might give false positives if the measures are noisy. To ensure that detected motion does correspond to intended gestures indicated by the conductor, the length of the overall gesture is calculated, as per the following equations:

$$u(t) = \begin{cases} 1 & \text{if } \left| \frac{dp(t)}{dt} \right| \geq V \\ 0 & \text{otherwise} \end{cases}$$

$$d(t) = \sum_{n=0}^{N_{start}} \left| \overrightarrow{p(t - nT_f)} - \overrightarrow{p(t - (n - 1)T_f)} \right| u(t)$$

where $\overrightarrow{p(t)}$ is the 3D vector position of the right hand at instant t , T_f represents the time between frames (roughly 30 milliseconds), N_{start} is the last known sample for which

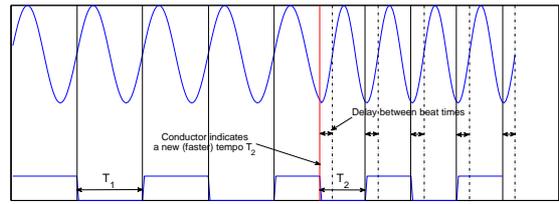


Figure 4. Delay effect when beat times are not properly synchronized

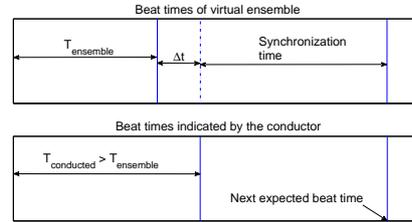


Figure 5. Beat time synchronization when the conductor indicates a slower tempo

$u(t)$ changed to a value of 1 and V is a minimum velocity value (set at approximately 0.2 m/s). Thus, given a waving gesture, if that gesture is performed horizontally and the accumulated distance moved $d(t)$ exceeds a certain minimum value L , it is assumed that the user has performed a conducted gesture. L was set to a reasonable value for such waving gestures, but long enough so that no arbitrary noise could trigger a false positive (approximately 400 mm).

However, the tempo conducted alone does not offer enough information for the system to adequately follow the conductor's indication, as just updating the system tempo without taking into consideration the actual position of the conducted beat times would result in a phase different between the beat times indicated by the conductor, and the actual beat times of the piece played. Such a situation creates the feeling that the orchestra is too slow and cannot follow the conductor gestures appropriately. This problem is better portrayed in figure 4, where the conductor's indicated tempo changes the period of a simple signal (a sinusoidal wave). If the conducted tempo only changes the playback tempo without taking the beat times into consideration (represented in the figure by the instants where the sinusoidal wave has a phase value of 0 radians), this introduces delay in the response of the system (the beat times of the piece played come at later time than the beat times indicated by the conductor).

To address this problem, it is necessary to synchronize the beat times of the piece played so that they match the next expected beat times that the conductor will most likely indicate. This assumption only makes sense if the tempo between beats is expected not to change too abruptly, but this is a reasonable assumption for the performance of an orchestra in real life. In particular, if the user conducts the virtual ensemble towards a slower tempo, the ensemble must actually play at an even slower tempo than the one indicated in order to synchronize its beat times with the ones of the conductor, and vice versa. This situation is portrayed in figure 6.

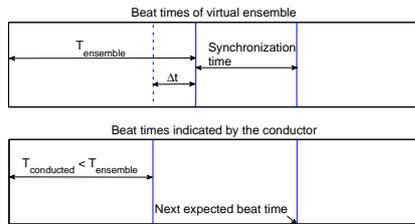


Figure 6. Beat time synchronization when the conductor indicates a faster tempo

In this figure, the user indicates a slower tempo with his gestures. However, the system does not realize this until a time of Δt seconds has passed. In order to match the conductor’s next expected beat time, it is necessary to decrease even further the tempo of the piece played for the time period denoted as ”synchronization time”. The opposite situation is portrayed in figure 5, where the conductor indicates the system to increase the tempo.

Therefore, whenever the user conducts the orchestra to a change in tempo, the system takes into account this time difference Δt to properly synchronize its next expected beat time with the user indications, modifying the playing tempo accordingly. Thus, the timestretching factor T_s is updated according to the expected beat times by following these equations.

$$T_s = \begin{cases} \frac{T_{conducted}}{T_{conducted} - \Delta t} T_s & \text{if } T_{conducted} > T_{ensemble} \\ \frac{T_{conducted}}{T_{conducted} + \Delta t} T_s & \text{if } T_{conducted} < T_{ensemble} \end{cases}$$

Initially, the system followed these equations to instantly change the tempo in the piece played to match the beat times of the conductor. Nevertheless, we found that the changes occurred too abruptly, making the response of the virtual ensemble feel rather unnatural. In fact, given a real orchestra, the musicians would not probably change the tempo in their performance instantly with the motion of the conductor, but would rather do it over a period of time. Thus, in order to offer a more natural answer, a second iteration of the system was implemented. This second version still calculates the expected beat times for adequate synchronization, but instead of automatically updating the tempo to the new value indicated by the conductor, the system dynamically updates the tempo of the piece played until both system and conductor beat times are sufficiently synchronized. Concretely, the tempo is slowed or accelerated by adding a factor of ± 0.025 to the timestretching factor at a rate of 4 times per second (thus, the timestretching value is updated in intervals of 250 ms).

2.2.2 Control of dynamics

In this system, the conductor would use his left hand to indicate the system how to control the dynamics of the performance. In particular, by raising his left hand, the conductor indicates the system to raise the volume of the performance, while lowering the hand brings the volume



Figure 7. Instrument selection for dynamics control

down. The application also allows the user to select a specific instrument of the ensemble and modify the volume levels for that instrument exclusively. In particular, the user only has to point toward the instrument he wants to select, and raise or lower his left hand accordingly to whether he wishes to raise or lower the level of volume. The system determines which instrument the user is pointing at by calculating the pointing vector of his left arm, taking the left shoulder and left hand positions as references. The application indicates which instrument set is currently selected by placing a red arrow over the image that represents that instrument set (see figure 7).

3. EXPERIMENTAL SETUP: METHODS AND MATERIALS

In this section, the different details of the experiment conducted will be presented, so that the same experiment can be easily reproduced by fellow researchers if needed.

3.1 Participants

A total of 24 participants took part in the experiment conducted, 3 female and 21 male, with ages ranging from 23 to 34 years (average 29,71 years, variance 10,30). There were 1 undergraduate, 15 graduates and 8 postgraduates. From the 24 participants, 2 had a strong formation in music, and 1 of these along with 2 more participants were actual professional musicians. Of the remaining 20 participants, 4 had played previously a musical instrument regularly. The rest of the participants (16) were nave musical users, with no previous formation or experience in music practice or music theory knowledge.

3.2 Materials

For the experiment, we used the previously discussed system. Thus, users were presented with a virtual reality application in which they had the possibility of modifying the tempo and/or the dynamics of the song played. For the purpose of the experiment, an excerpt of Peer Gynt’s ”In the hall of the mountain king” was played constantly in a loop

while the system was tracking user's movements, and the playback was modified according to the motion detected. Two sets of instruments were considered for the tests with their corresponding WAV files: violin and trombone.

3.3 Procedure

The experiment was performed in a research lab in the School of Telecommunications of Málaga. Each participant performed the trials scheduled assisted by a researcher, who explained him/her the details of the tests as well as observed the participants behaviour during the experiment. Participants were instructed to use their right hand waving motion to conduct the tempo in the ensemble, and their left hand to indicate changes in dynamics (in the same way as described in the previous section). At the end of their performance, the researcher asked the participants to fill in a questionnaire concerning their opinion on the experience; additionally, the researcher also had a casual interview with the participants regarding their overall experience and their perception of the strengths and weak points of the system.

From a previous pilot study with a smaller sample of participants (4 in total), we had found that users did not notice the effects of the tempo synchronization algorithm in their experience, i.e., they seemed to be satisfied with just being able to change the tempo in the piece by "waving" their right hand, but did not pay attention to whether the beat times of the piece were synchronised or not with the hand motion's starting and ending points. Also, users had described the dynamic control interaction implemented to be sort of cumbersome and detrimental to the experience.

In order to further assess these issues, we defined two experimental factors in our study: a *tempo* factor and a *dynamics* factor. The *tempo* factor controls whether the synchronization algorithm previously described was present or not, while the *dynamics* factor controls whether the user can modify the dynamics in the piece being played, or just the tempo of the piece.

The combination of the two factors yields a total of $2 \times 2 = 4$ experimental conditions. A repeated measures approach was followed [26], so that there were 4 experimental sessions for each participant, each session corresponding to one of the aforementioned experimental conditions. To avoid order effects, the order in which the participants performed their sessions was fully counter-balanced. Each session was scheduled to last no less than 2 minutes and no more than 7 minutes. Each participant was told to spend as much time as they deemed necessary "playing" with the application at each experimental session, and were only instructed to stop or continue if the aforementioned time constraints were not met.

3.4 Data retrieval on user experience

The data was collected from the questions listed in the questionnaire which participants filled in at the end of the experiment. In extent, each participant was asked to evaluate the following aspects of their experience with a score from 0 (least satisfactory) to 10 (most satisfactory):

- Overall satisfaction with the application (*Satisfaction*)
- Level of control over the parameters of the piece played (*OverallControl*)
- Level of control over the tempo of the piece played (*TempoControl*)
- Synchronization between motion and the changes in the piece played (*Synchronization*)
- How intuitive was the interaction (*Intuitiveness*)
- Ease of use of the application (*EaseOfUse*)
- Level of realism perceived (*Realism*)

For each of this items, a dependent variable was created (with the name indicated in brackets). In addition to the aforementioned items, users were also encouraged to state personal comments and impressions regarding their experience with the application.

4. RESULTS

In order to analyze the variables, a repeated measures two-factors ANOVA 2×2 was performed on the factors *tempo* and *dynamics* previously defined. The principal effects analysis for the *tempo* factor had a significant effect on the variables *Satisfaction* ($F_{1,23} = 25.09, p < 0.000$), *OverallControl* ($F_{1,23} = 18.81, p < 0.000$), *TempoControl* ($F_{1,23} = 21.49, p < 0.000$), *Synchronization* ($F_{1,23} = 15.02, p < 0.001$) and *Realism* ($F_{1,23} = 6.27, p < 0.020$). In the case of the *dynamics* factor, there was a significant effect on the variables *Satisfaction* ($F_{1,23} = 9.75, p < 0.005$), *OverallControl* ($F_{1,23} = 9.37, p < 0.006$), *Synchronization* ($F_{1,23} = 15.02, p < 0.005$), *Intuitiveness* ($F_{1,23} = 5.28, p < 0.031$) and *EaseOfUse* ($F_{1,23} = 13.80, p < 0.001$). The estimated marginal means for the variables *Satisfaction*, *OverallControl*, *TempoControl* and *Synchronization* are presented in figure 8.

The quantitative effects that each of these factors had on the average values for each of the variables observed are summarized in table 1. Concretely, in the case of the *tempo* factor, every variable where it had a significant effect increases its value when the tempo synchronization algorithm is present. The same situation is found for the *dynamics* factor, with the exception of *Intuitiveness* and *EaseOfUse* variables, which offer lower values when the user is allowed to control the dynamics of the piece.

No significant second order interactions were found between the two experimental factors considered ($F_{1,23} \leq 3.01$ for all the variables observed). Overall, user experience according to the variables observed was quite positive, with *Intuitiveness* being the variable that scored the highest values and *EaseOfUse* being the one that presented the highest variance. Figure 9 illustrates the average score for each variable and their standard deviations.

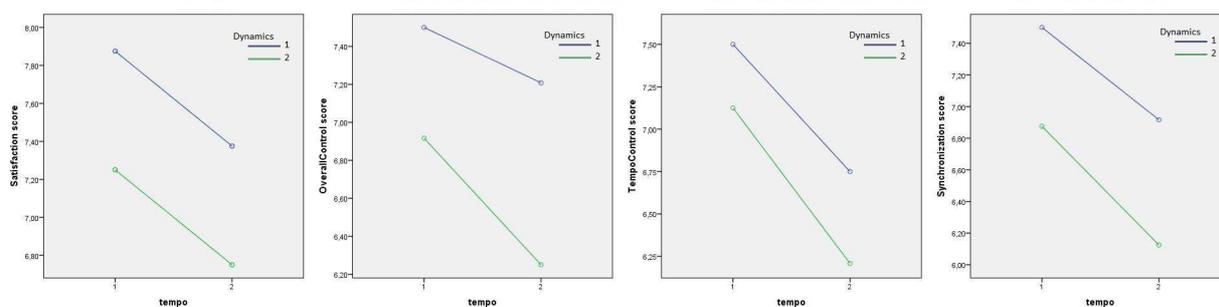


Figure 8. Estimated Marginal Means for the dependent variables Satisfaction, OverallControl, TempoControl and Synchronization. The *tempo* factor takes values 1 (tempo synchronization present) or 2 (tempo synchronization not present). The *dynamics* factor takes values 1 (dynamics control present) or 2 (dynamics control not present)

	Cond. 1	Cond. 2	Cond. 3	Cond. 4
Satisfaction	7.875	7.25	7.375	6.75
OverallControl	7.5	6.917	7.208	6.25
TempoControl	7.5	7.125	6.75	6.208
Synchronization	7.5	6.875	6.917	6.125
Intuitiveness	8.292	8.417	8.125	8.458
EaseOfUse	7.208	7.792	7	7.7917
Realism	7.25	7.083	7	6.833

Table 1. Average scores for each variable observed at each of the 4 experimental conditions: condition 1 (both tempo synchronization and dynamics control present), condition 2 (only tempo synchronization present), condition 3 (only dynamics control present) and condition 4 (none present)

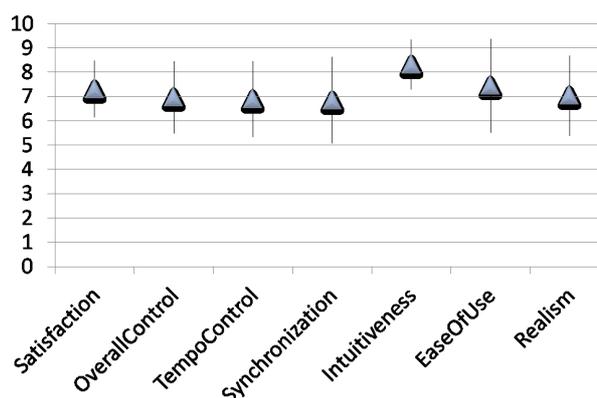


Figure 9. Average values for the variables considered, along with \pm their standard deviations σ

5. DISCUSSION

The results yielded from the experiment conducted showed a quite positive response from the participants that took part in the experiment. As expected, an adequate synchronization between the beat times in the piece and the starting and final positions of user's "waving" motions was critical to the overall experience of the user. Interestingly though, from the interviews had with the participants, the vast majority of them did not consciously notice a significant difference between the two tempo conducting modes considered. Nevertheless, the results extracted from the analysis of the variables observed did show that user perception of satisfaction, control and realism among others was indeed significantly higher if the beat times of the piece were adequately synchronised with the beat times of the hand. In the case of the 4 participants that had a strongest musical background, they did acknowledge to have noticed this difference between the two conducting modes, yet no particular differences were found in the statistical analysis performed in this regard.

The presence of dynamics control showed also a positive income in user experience regarding satisfaction, control and synchronization. However, the added complexity of the interface made the system less intuitive and, especially, more difficult to use. In fact, from observation of user behaviour during the experiments, some of the participants

found it difficult to control both tempo and dynamics at the same time, as their left hand might hamper their right hand motion when trying to select the violin. This is a flaw that comes mainly from the camera-based nature of the system, as it may be possible that one hand obscured the line of sight of the 3D sensor to the other one. In the particular case of the right hand, the system was highly sensitive to this kind of occlusion.

Previous work has focused mainly on capturing the conductor's gestures to modify the tempo of the piece played by applying the corresponding timestretching algorithm. A few studies, however, have also implemented the possibility of controlling the dynamics of the piece played, as is the case of Borchers et al. [15], offering a much more complete experience to the users. Our work also aims to offer this more complete experience, by adding the possibility of controlling the volume of the different instruments in the ensemble. However, as found in the tests performed, additional steps must be taken to ensure that users can actively used both hands without interfering the commands given by each other because of occlusion.

Most of the previous research has favored the use of infrared or inertial batons [15] [16] [17] [18], or, more recently, the use of Nintendo's Wiimote [19] [20]. However, this kind of devices is usually very expensive [17] or have ergonomic and usability issues (in the case of the Wii Re-

mote, its shape is not that adequate for baton emulation, and its additional weight when compared with an infrared baton [21] might rise some issues in long sessions). By using a Kinect device as the basis of our system, we provide a non-intrusive interaction paradigm that minimizes the effects of such issues (both major and minor). Interestingly though, we would like to point out the fact that, in our user study, we also found that a small sample of the participants got "tired" after the experimental sessions and even reported arm pain because of the conducting gesture (2 cases). However, this can be explained in the unusual length of the experimental sessions.

Finally, one aspect that some participants criticized in the application was that they perceived some lag between their motion and the response given by the system. This is caused because of a delay introduced by the sensing device, and it is an issue where the system should be improved in its next iteration.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the work performed towards the implementation of an advanced human-computer interface for conductor simulation, using an off-the-shelf device that allows for optimal usability without involving a high purchase cost. We have implemented a time-stretching algorithm for tempo modification and developed a gesture recognition system for dynamics and tempo indication by the user. The application developed has been tested by musicians and naïve users, with positive impressions on the experience perceived by both types of users. Also, it has been experimentally confirmed that the addition of a better synchronization algorithm and dynamics control does indeed improve user experience, even if the users were not consciously aware of it. From the results yielded in the experiment, we conclude that the application developed provides a satisfactory exploratory experience in music interaction, which can be enjoyed alike by nave and expert users.

In future works we hope to improve further on the system designed. In particular, we intend on improving the gesture recognition module to expand the range of gestures identified from the already supported waving gesture to more complex gestures similar to the ones performed by orchestra conductors according to the time signature of the piece played. Also, as indicated in previous works [27], other features from the conductor's gesturing and body expression can have a significant effect on musician action. We also hope to expand the tests of the application by performing a larger user study to gather additional data in order to identify other key aspects where the application can be improved towards a better user experience. Last but not least, as satisfactory as the user response has been, we have found that issues like hand occlusion and lag perception need to be addressed and improved upon for future implementations of the system.

Acknowledgments

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2010-21089-C03-02 and Project No. IPT-2011-0885-430000 and by the Junta de Andalucía under Project No. P11-TIC-7154. The work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

7. REFERENCES

- [1] S. Jordà, "The reactable: tangible and tabletop music performance," in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. ACM, 2010, pp. 2989–2994.
- [2] A. Antle, M. Droumeva, and G. Corness, "Playing with the sound maker: do embodied metaphors help children learn?" in *Proceedings of the 7th international conference on Interaction design and children*. ACM, 2008, pp. 178–185.
- [3] E. Khoo, T. Merritt, V. Fei, W. Liu, H. Rahaman, J. Prasad, and T. Marsh, "Body music: physical exploration of music theory," in *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, 2008, pp. 35–42.
- [4] G. Castellano, R. Bresin, A. Camurri, and G. Volpe, "Expressive control of music and visual media by full-body movement," in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 390–391.
- [5] M. Halpern, J. Tholander, M. Evjen, S. Davis, A. Ehrlich, K. Schustak, E. Baumer, and G. Gay, "Moboogie: creative expression through whole body musical interaction," in *Proceedings of the 2011 annual conference on Human factors in computing systems*. ACM, 2011, pp. 557–560.
- [6] L. Gower and J. McDowall, "Interactive music video games and children's musical development," *British Journal of Music Education*, vol. 29, no. 01, pp. 91–105, 2012.
- [7] C. Wang and A. Lai, "Development of a mobile rhythm learning system based on digital game-based learning companion," *Edutainment Technologies. Educational Games and Virtual Reality/Augmented Reality Applications*, pp. 92–100, 2011.
- [8] G. Levin and Z. Lieberman, "In-situ speech visualization in real-time interactive installation and performance," in *Non-Photorealistic Animation and Rendering: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, vol. 7, no. 09, 2004, pp. 7–14.
- [9] S. Bakker, E. van den Hoven, and A. Antle, "Moso tangibles: evaluating embodied learning," in *Proceedings*

of the fifth international conference on Tangible, embedded, and embodied interaction. ACM, 2011, pp. 85–92.

- [10] S. Holland, A. Bouwer, M. Dalgelish, and T. Hurtig, “Feeling the beat where it counts: fostering multi-limb rhythm skills with the haptic drum kit,” in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*. ACM, 2010, pp. 21–28.
- [11] S. Trail, M. Dean, T. Tavares, G. Odowichuk, P. Driessen, W. Schloss, and G. Tzanetakis, “Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect,” 2012.
- [12] K. Ng, “Music via motion: transdomain mapping of motion and sound for interactive performances,” *Proceedings of the IEEE*, vol. 92, no. 4, pp. 645–655, 2004.
- [13] A. Hofer, A. Hadjakos, and M. Mhlhuser, “Gyroscope-Based Conducting Gesture Recognition,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2009, pp. 175–176. [Online]. Available: http://www.nime.org/proceedings/2009/nime2009_175.pdf
- [14] H. Morita, S. Hashimoto, and S. Ohteru, “A computer music system that follows a human conductor,” *Computer*, vol. 24, no. 7, pp. 44–53, 1991.
- [15] J. Borchers, E. Lee, W. Samminger, and M. Mühlhäuser, “Personal orchestra: A real-time audio/video system for interactive conducting,” *Multimedia Systems*, vol. 9, no. 5, pp. 458–465, 2004.
- [16] L. Peng and D. Gerhard, “A wii-based gestural interface for computer-based conducting systems,” in *Proceedings of the 2009 Conference on New Interfaces For Musical Expression*, 2009.
- [17] E. Lee, T. Nakra, and J. Borchers, “You’re the conductor: a realistic interactive conducting system for children,” in *Proceedings of the 2004 conference on New interfaces for musical expression*. National University of Singapore, 2004, pp. 68–73.
- [18] P. Bakanas, J. Armitage, J. Balmer, P. Halpin, K. Hudspeth, and K. Ng, “mconduct: Gesture transmission and reconstruction for distributed performance,” in *ECLAP 2012 Conference on Information Technologies for Performing Arts, Media Access and Entertainment*. Firenze University Press, 2012, p. 107.
- [19] D. Bradshaw and K. Ng, “Analyzing a conductors gestures with the wiimote,” in *Proceedings of EVA London 2008: the International Conference of Electronic Visualisation and the Arts*, 2008.
- [20] T. Nakra, Y. Ivanov, P. Smaragdis, and C. Ault, “The ubs virtual maestro: An interactive conducting system,” *NIME2009*, pp. 250–255, 2009.
- [21] T. Baba, M. Hashida, and H. Katayose, “virtualphilharmony: A conducting system with heuristics of conducting an orchestra,” in *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, 2010, pp. 263–270.
- [22] D. Shreiner, *OpenGL reference manual: The official reference document to OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [23] G. Junker, *Pro OGRE 3D programming*. Apress, 2006.
- [24] U. Zölzer, X. Amatriain, and J. Wiley, *DAFX: digital audio effects*. Wiley Online Library, 2002, vol. 1.
- [25] D. Malah, “Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 121–133, 1979.
- [26] D. C. Howell, *Statistical methods for psychology*. Wadsworth Publishing Company, 2012.
- [27] K. Parton and G. Edwards, “Features of conductor gesture: Towards a framework for analysis within interaction,” in *The Second International Conference on Music Communication Science, 3-4 December 2009, Sydney, Australia*, 2009.