THE “AIR WORM”: AN INTERFACE FOR REAL-TIME MANIPULATION OF EXPRESSIVE MUSIC PERFORMANCE

Simon Dixon, Werner Goebl
Austrian Research Institute for
Artificial Intelligence, Vienna

Gerhard Widmer
Department of Computational Perception
Johannes Kepler University, Linz, Austria

ABSTRACT

Expressive performance of traditional Western music is a complex phenomenon which is mastered by few, and yet appreciated by many. In this paper we explore various ways of interacting with expressive performances using methods that are accessible to non-expert music-lovers. A digital theremin is used as an input device, and users can control the two most important expressive parameters, tempo and loudness, during playback of an audio or MIDI file. Several modes of operation are possible: the “Air Worm” builds on previous work in performance visualisation, where the tempo is displayed on the horizontal axis and loudness on the vertical axis in a two-dimensional animation; the “Air Tapper” uses a conducting metaphor where the beat is given by the minimum vertical point in a quasi-periodic trajectory; and the “Mouse-Worm” allows users without a theremin to use a standard input device as controller.

1. INTRODUCTION

In Western art music, expert performers go beyond the instructions written in the musical score and shape the music by their use of parameters such as tempo, dynamics and articulation, in order to communicate both emotional and structural information. Despite significant research on this phenomenon (see [9] for a review), a formal model of expressive music performance remains an elusive goal. Only a small fraction of the observed expressive variations are explained by computational models [17], and computer generated performances are far from reaching the standard of human musical interpretation [11]. In other words, expressive performance is a complex domain requiring a great amount of training and experience to master it. At the same time, non-expert listeners (i.e. non-performers) can appreciate this art form and distinguish and categorise interpretations as good, bad, interesting, conservative, etc.

In this paper we explore the idea of transforming the passive listening experience into an active involvement with the music. In order to do this, we must address the problem of controlling this complex phenomenon without creating an interface that is difficult to use or that requires a great deal of learning. For example, musical instruments typically provide a rich interface with many parameters (and thus a high level of control), but this level of learning is beyond the scope of our target audience. Further, since expressive parameters are in general not well understood, or are heavily dependent on musical context, it is difficult or impossible to automate the setting of parameters. Our solution is to *start* with expert expressive performances, and allow users to edit or modify the performances in simple and transparent ways. Three main issues are addressed: the choice of suitable interfaces for non-expert users, the mapping of user actions to modifications of existing performances, and the implementation issues of modifying the performances.

As input device, we use a digital theremin, which allows users to control two parameters by the position of their hand(s) relative to two antennas. In the first system, called the Air Worm, we employ the metaphor of the Performance Worm [7, 13], where tempo and loudness are displayed in a 2-dimensional animation. The idea of this visualisation is then inverted so that the user’s movements in the tempo-loudness space control the expressive parameters, enabling the user to specify complex expressive trajectories with a wave of the hand. The second system uses a more standard conducting paradigm, where the tempo is defined by the inter-beat intervals between successive minima of quasi-periodic vertical hand movements. As an alternative to the use of the theremin, we also implemented versions of these two systems which use the computer mouse for input (the Mouse Worm and Mouse Tapper), so that computer owners can use the system without needing any special hardware.

The Air Worm works with both MIDI and audio input data. In choosing the format of performance data, there is a tradeoff between the quality of the output and the ease of modification of the data. If the data is provided in a symbolic format, such as MIDI, it is relatively easy to modify parameters such as tempo, loudness and articulation. On the other hand, loudness is the only parameter of audio data which can be easily modified; all others require complex processing. For example, a tempo change should scale the timing of events without changing the pitch, formants or timbre of the sounds. We use the synchronous overlap and add (SOLA) method to perform audio time scaling, which is a simple and computationally efficient method, but its output suffers from some audible artifacts of the scaling process.

In the remainder of the paper we provide a review of related work followed by an outline of the digital theremin.

We then describe the Air Worm and Air Tapper, and conclude with a discussion of the systems.

2. BACKGROUND

A historic example of expression control is the push-up piano player (e.g., the “Pianola”) that was built from the early 1900s for public and home entertainment [2, p. 255]. To add expression to the mechanical piano rolls, manufacturers gave the users the option to modify the tempo and dynamics during playback through levers on the machine. Artistically refined lay users could in this way create a performance that contained personalised expression.

Modern interfaces for musical instruments are numerous, but only a few systems are dedicated to controlling or manipulating existing expressive performances. A system that involves analogies to conducting is the Radio Baton control interface [14] which was combined with the Director Musices performance grammar implementation [8] into a successful artistic performance tool [15], as demonstrated in various public performances. The “conductor” controls the beat by a constant up and down movement of one baton, and alters the overall intensity with a second baton. The system uses scores annotated with some basic expression generated by the KTH performance rules.

A system using a similar beat input methodology was presented by Katayose and Okudaira [12], who used a MIDI theremin to track hand movements and control the tempo and intensity of preprocessed expressive MIDI performances. In a GUI the user can preset different parameters, e.g., the amount of the user’s influence on the expressive performance or the responsiveness of the gesture tracking system. A meticulous usability test showed that users appreciated the system.

A complete conducting system was proposed by Murphy [16]. It involves visual tracking of real conductors’ gestures in different basic meters (as monitored by a digital camera), a rough real-time beat estimation from audio files, and audio morphing according to the tracked tempo changes (done by existing software). However, no systematic evaluation was reported on how intuitively and responsively this system operates.

3. THE MIDI THEREMIN

The theremin is a musical instrument developed in the early 1900’s by the Russian physicist Leon Theremin. In its simplest form, it consists of two high frequency oscillators, one of fixed frequency, the other of variable frequency, which are combined to produce frequencies in the audible range by heterodyning (beat tones). The variable frequency oscillator is controlled by moving the hand towards or away from a vertical antenna, which changes the frequency of oscillation and thus of the beat tone. A horizontal antenna is used in a similar way to control the loudness of the signal. Unlike traditional musical instruments, the performer makes no physical contact with the instrument. This has the advantage of giving the performer a great deal of freedom, but the disadvantage of not providing the performer with any haptic feedback.

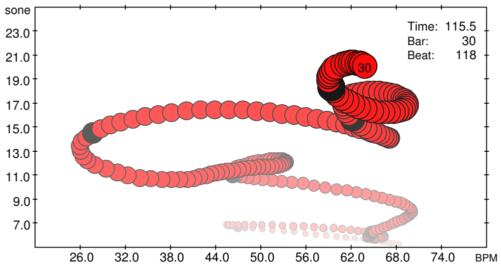

Figure 1. The Performance Worm is an animation of some of the expressive parameters of a musical performance. The position of the head of the worm gives the current tempo (horizontal axis) and loudness level (vertical axis). The figure shows the display at bar 30 of Rachmaninov’s Prelude op.23 no.6 played by Vladimir Ashkenazy.

The theremin provides continuous control over the parameters pitch and intensity, in stark contrast to the keyboard, which has a fixed set of discrete pitches and usually no control of intensity after the initial attack (except to end the tone). The MIDI protocol is strongly based on the keyboard idiom, with musical tones being represented by a note-on and note-off pair, where the intensity is determined by the *velocity* parameter of the note-on message. Additional MIDI messages, such as pitch bend and controller change messages complement the simplistic note descriptions to provide a richer control of sound.

Thus the MIDI theremin is something of a contradiction in terms, mixing discrete and continuous representations of sound. In fact the normal note on and off commands are not used at all; instead, a change in pitch (horizontal axis) is transmitted as a pitch bend message (14-bit resolution) and change in intensity (vertical axis) as a controller change message (7-bit resolution). We treat the theremin as an abstract controller of two independent parameters, rather than constraining it to control pitch and intensity. The system we use (MOO System MDT–02) transmits these two values at a rate of approximately 50 times per second, which is sufficient for low-latency tracking of gestures with a reasonable resolution.

4. THE AIR WORM

The Performance Worm provides a visualisation of the two most important parameters of expressive performance, tempo and dynamics, in a simple 2-dimensional animation, where the evolution in time of these parameters is visible as a trajectory which fades into the background (Figure 1). This intuitive representation can be inverted and used for control. That is, the user specifies the tempo-dynamics trajectory, either beforehand or in real time as the music is being played.

The Air Worm enables real-time control of a performance trajectory using the MIDI theremin. The position of the user's hand is tracked in order to steer the head of the Worm, and the tempo and dynamics of the performance are modified accordingly. The MIDI theremin provides approximately linear estimates of hand position across its range of operation, which are then scaled so they can be interpreted as tempo and dynamics values. The standard Worm display is shown on the computer screen in order to give the user visual feedback, which is particularly important since there is no tactile feedback in the system.

If the musical data is in MIDI format, it is a simple task to modify the tempo. This is done by means of a tempo factor F, which scales all time intervals used in playback. The 14-bit (-8192 to 8191) input value is mapped to an exponential curve via the formula:

$$F_{out} = k^{\frac{F_{in}}{8192}}$$

where k is the maximum tempo factor. For example, if $k = 2$, the tempo changes range between half and double the original tempo. The 7-bit (0 to 127) input value for volume is scaled linearly and used as the master volume setting:

$$V_{out} = \frac{V_{in}}{127}$$

For audio input data, advanced methods exist for time-scale modification without corresponding changes in pitch or timbre of the sounds (see [1] for a review). We use a time-domain method, synchronous overlap and add, which reduces amplitude and phase discontinuity at audio segment boundaries by cross-correlation of the overlapping portions of successive segments.

4.1. The Mouse-Worm

As an extension of the above-mentioned work, a mouse-driven interface was created as an alternative to using the theremin for input. The main advantage of this approach is that no special hardware is required to manipulate performances. It is also possible to have finer control over the parameters, since the user's hand is resting on the desk rather than being held as steadily as possible in the air.

5. THE AIR TAPPER

An orchestral conductor communicates high-level interpretive instructions to trained musicians via arm and hand movements, and this is thought to be a particularly natural method of expression because it places no constraints on the conductor. Timing information is communicated primarily via the trajectory of the baton, where beat times are given by turning points in the trajectory. The control of loudness is not so explicit, but it is generally correlated with the extent of the trajectory. Obviously this is a vast simplification of a complicated communication protocol, but it was necessary to define a simple protocol for the following implementation using the theremin.

The following steps are involved in using the theremin as a conducting device: encoding the timing information from the gestures, finding the corresponding beat times in the music, and synchronising the music reproduction to the gestures in real time. The beat is extracted by tracking the distance of the user's hand from the horizontal antenna and finding local minima. The time of each local minimum is taken to be the time of the beat, and the tempo is then calculated from the inter-beat interval. False beats are filtered out by deleting any minima where the maximum value since the last minimum is less than 10 units more than the minimum. Dynamics are controlled by the proximity of the hand to the vertical antenna, but since this does not remain constant during the conducting trajectory, the average distance from the vertical antenna is calculated for each beat and updated once per beat. An easier method of controlling dynamics is to conduct the beat at the far end of the horizontal antenna with one hand and use the other hand for dynamics.

In order to synchronise the music to the conducting, we need to know the times of the beats in the music files. The vast literature on beat tracking testifies to the difficulty of this task (see [10] for a recent review). We assume in general that the timing of beats is supplied as metadata, so that the notated beats can be aligned with the gestural data, although this feature has not yet been implemented. The current system supplies a tempo factor based on the rate of conducting, but does not synchronise the musical data to the conducting, which is a reasonable default when the beat times of the music are not known. In future work, we intend to integrate a beat tracking system (e.g., BeatRoot [5, 6]), so that metadata will not be required.

6. DISCUSSION AND CONCLUSION

We described the Air Worm, a new interface for manipulating musical expression by tracing out a trajectory in a two-dimensional tempo-loudness space; the Air Tapper, which provides an alternative way of specifying timing information; and modifications of these systems for use with a mouse instead of a digital theremin. Informal evaluation confirmed that these are intuitive and useful interfaces to expressive performance, even for the lay person.

One limitation of these control methods is that musical context is not taken into account. For best results, it would be necessary to consider timing and tempo separately [4, 3]. For example, a doubling of tempo should not necessarily double the rate of a trill or halve the length of a grace note, which is the current effect of the system. For typical expressive modifications, which are usually quite subtle, this does not turn out to be a great problem, but for more extreme modifications, it is a significant issue. For MIDI input, it is possible to detect such cases and modify the sequence accordingly, for example by adding or removing notes from a trill, but this would be virtually impossible to accomplish for polyphonic audio input. Likewise, the use of a master volume control for dynamics should be changed to modify the MIDI velocities of the notes, since louder notes have a different tone (usually more high frequency content) than quiet notes. (We assume that the synthesiser takes these factors into account; this is not true of all synthesisers.) Once again, this process is much more difficult for audio, since the effects of dynamics are more complex than can be achieved by simple operations such as filtering.

Another limitation is that control is given over only a small subset of all interpretative possibilities. However, this is also a great advantage of the system: the lower level aspects of interpretation (e.g. articulation, chord asynchronies, inter-voice dynamics) are provided by the original performance. We assume a hierarchical representation of interpretation, where interpretive strategies at each level of phrasing can be composed to give a complete interpretation [17, 18]. This is why real performances are used for input rather than flat mechanical renditions of the score: the low-level interpretive choices are then provided automatically. To give the user control over these choices as well would require a much more complex interface with a correspondingly steep learning curve, which would be at odds with our goal of making musical interpretation more accessible to non-experts.

7. ACKNOWLEDGEMENTS

This work was supported by the Vienna Science and Technology Fund as part of the project CI010 *Interfaces to Music*, and the Austrian Federal Ministry of Education, Science and Culture (BMBWK) as part of the START project Y99-INF. The Austrian Research Institute for Artificial Intelligence acknowledges the financial support of the BMBWK and the Austrian Federal Ministry of Transport, Innovation and Technology. We also thank Haruhiro Katayose for his generous gift of the digital theremin.

8. REFERENCES

[1] Bonada, J. (2002). Audio time-scale modification in the context of professional audio post-production. Predoctoral thesis, Universitat Pompeu Fabra, Audiovisual Institute.

[2] Bowers, Q. D. (1972). *Encyclopedia of Automatic Musical Instruments* (13th ed.). New York: Vestal Press Ltd.

[3] Desain, P. and H. Honing (1991a). Tempo curves considered harmful: A critical review of the representation of timing in computer music. In *Proceedings of the International Computer Music Conference*.

[4] Desain, P. and H. Honing (1991b). Towards a calculus for expressive timing in music. *Computers in Music Research 3*, 43–120.

[5] Dixon, S. (2001a). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research 30*(1), 39–58.

[6] Dixon, S. (2001b). An interactive beat tracking and visualisation system. In *Proceedings of the International Computer Music Conference*, pp. 215–218.

[7] Dixon, S., W. Goebl, and G. Widmer (2002). The Performance Worm: Real time visualisation of expression based on langner's tempo-loudness animation. In *Proceedings of the International Computer Music Conference*, pp. 361–364.

[8] Friberg, A., V. Colombo, L. Frydn, and J. Sundberg (2000). Generating musical performances with Director Musices. *Comp. Mus. J. 24*(3), 23–29.

[9] Gabrielsson, A. (2003). Music performance research at the millenium. *Psych. Mus. 31*(3), 221–272.

[10] Gouyon, F. and S. Dixon (2005). A review of automatic rhythm description systems. *Computer Music Journal 29*(1), 34–54.

[11] Hiraga, R., R. Bresin, K. Hirata, and H. Katayose (2004). Rencon 2004: Turing Test for musical expression. In *Proc. of the 2004 Conf. on New Interfaces for Musical Expression (NIME04)*, pp. 120–123. Hamamatsu, Japan.

[12] Katayose, H. and K. Okudaira (2004). Using an expressive performance template in a music conducting interface. In *Proc. of the 2004 Conf. on New Interfaces for Musical Expression (NIME04)*, pp. 124–129. Hamamatsu, Japan.

[13] Langner, J. and W. Goebl (2003). Visualizing expressive performance in tempo-loudness space. *Computer Music Journal 27*(4), 69–83.

[14] Mathews, M. V. (1991). The Radio Baton and the Conductor Program. *Comp. Mus. J. 15*(4), 37–46.

[15] Mathews, M. V., A. Friberg, G. Bennett, C. Sapp, and J. Sundberg (2003). A marriage of the Director Musices program and the conductor program. In R. Bresin (Ed.), *Proc. of the Stockholm Music Acoustics Conf. (SMAC'03), August 6–9, 2003*, Volume 1, pp. 13–15. Stockholm, Sweden: Dept. of Speech, Music, and Hearing, Royal Inst. of Technology.

[16] Murphy, D. (2004). Conducting audio files via computer vision. In A. Camurri and G. Volpe (Eds.), *Gesture-Based Communication in Human-Computer Interaction*, LNCS 2915, pp. 529–540. Berlin: Springer.

[17] Widmer, G. (2002). Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research 31*(1), 37–50.

[18] Widmer, G. and A. Tobudic (2003). Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research 32*(3), 259–268.