USING COMPUTER ACCOMPANIMENT TO ASSIST NETWORKED MUSIC PERFORMANCE

CHRISOULA ALEXANDRAKI[1] AND ROLF BADER[2]

[1] *Dept. of Music Technology and Acoustics Engineering, Technological Educational Institute of Crete, Greece*
chrisoula@staff.teicrete.gr
[2] *Institute of Musicology, University of Hamburg, Hamburg, Germany*
R_Bader@t-online.de

This paper proposes a novel scheme for audio communication in synchronous musical performances carried out over computer networks. The proposed scheme uses techniques inspired from computer accompaniment systems, in which a software agent follows the performance of a human musician in real-time, by synchronizing a pre-recorded musical accompaniment to the live performance. In networked settings, we attempt to represent each remote performer participating in the networked session by a local software agent, which adapts a pre-recorded solo performance of each musician to the live music being performed at remote locations.

INTRODUCTION

Within the last decades, the ever increasing availability of computational resources and the vast digitization of musical material have led to a sound-based rather than conventional score-based analysis of musical works. There are many reasons why this discipline shift was essential, including the fact that in most popular and folk music genres there is no score at all describing musicians' performance. Most importantly however, this shift was imposed by the fact that musical scores do not effectively reveal central aspects of musical performance. Focusing on musical performance, audio-based analysis of musical material has on one side enabled the computational modelling of musical interpretation [1] and on the other side permitted the development of software agents that are capable of listening, performing and composing music at a level which is comparable to human musical skills. The development of such agents is the main focus of a research track known as computer accompaniment [2] or more recently 'Human Computer Music Performance' (HCMP) [3].

Meanwhile, the advent of broadband and highly reliable network infrastructures has enabled distributed, network-based synchronous musical collaborations. Networked Music Performance (NMP) is becoming increasingly popular both among researchers and music scholars as well as among interested individuals. Although current network infrastructures allow transatlantic musical collaborations [4], NMP still remains a challenge. This is evident by the experimental nature of such performances, as well as by the fact that NMP technology is not widely offered to musicians.

This paper proposes an innovative perspective for the establishment of real-time NMP communications which exploits achievements from the area of HCMP. Specifically, this work investigates the idea of representing each performer of a dispersed NMP system by a local computer-based musician. For each musician participating in an NMP session, a local agent 'listens' to the local performance, 'notifies' remote collaborators and 'performs' the music reproduced at remote ends, therefore eliminating the need for audio stream exchange. Listening involves detecting the occurrence of a new note in real-time (i.e. at the onset). Notifying involves informing remote peers about the arrival of a new note using low bandwidth information. Finally, performing involves receiving notifications about the remote occurrence of new notes and rendering the performance of the corresponding musicians using pre-recorded solo tracks. These tracks are adapted in terms of tempo and loudness, so as to better reflect the expressive aspects of the remote live performance. Assuming that the algorithms implementing the functionalities of listening and performing can become sufficiently robust, this type of communication scheme can provide superior sound quality compared to alternative low-latency and low-bitrate transmission of music, such as using MIDI or facilitating compression codecs.

The rest of this paper is structured as follows: the next section presents a brief overview of research achievements relevant to the present work. Following, the methodology of the proposed approach is described in terms of the algorithmic implementation of the required functionalities. The section that follows presents preliminary evaluation results for the offline and real-time performance of the respective algorithms.

Finally, the paper is concluded by a brief discussion of achievements, shortcomings and future challenges.

1 RELATED WORK

The perspective of analysing a live performance and using the results of this analysis to inform remote peers in networked music collaborations has not been adequately investigated or even reflected in the relevant research literature up to now. The following subsections provide a brief overview of research trends in the domains of NMP and HCMP. The last subsection presents some research initiatives aiming at combining achievements from both domains.

1.1 Networked Music Performance

Physical proximity of musicians and co-location in physical space are typical pre-requisites for collaborative music performance. Nevertheless, the idea of music performers collaborating across geographical distance was remarkably intriguing since the early days of computer music research. The relevant literature appoints the first experimental attempts for interconnected musical collaboration to the years of John Cage. Specifically, the 1951 piece “Imaginary Landscape No. 4 for twelve radios” is regarded as the earliest attempt for remote music collaborations [5]. Telepresence across geographical distance initially appeared in the late 1990s [6] either as control data transmission, noticeably using protocols such as the Remote Music Control Protocol (RMCP) [7] and later the OpenSound Control [8] or as one way audio transmission from an orchestra to a remote audience [9]. True bidirectional remote audio interactions became possible with advent of broadband academic network infrastructures in 2001, the Internet2 in the US and later the European GEANT. In music, these networks enabled the development of frameworks that allowed remotely located musicians to collaborate as if they were co-located. As presented by the Wikipedia, current known systems of this kind are the Jacktrip application [10], currently distributed with an open source license, the DIP [11] and the DIAMOUSES project [12]. These systems currently form the main bulk of academic research in NMP.

At present reliable NMP is restricted within academic community boundaries having access to high-speed networks. As a result, NMP research is not offered to its intended target users (i.e. music performers) and thus has not yet revealed its full potential. The main technological barriers to implementing realistic NMP systems concern the fact that these systems are highly sensitive in terms of latency and synchronization, because of the requirement for ‘real-time’ communication, as well as highly demanding in terms of bandwidth availability and error alleviation, because of the acoustic properties of music signals. In this respect, a substantial body of research efforts are currently being invested in developing audio codecs intended to eliminate network bandwidth demand, without significantly affecting audio quality or communication latencies [13].

1.2 Computer Accompaniment

In the mid 80s, the concept of the ‘synthetic performer’ appears through the inspiring works of Vercoe [14], and Dannenberg [15]. The motivation in these works is grounded on a computer system which will be able to replace any member of a music ensemble through its ability to listen, perform and learn musical structures in a way which is comparable to the one employed by humans. The concept of the synthetic performer was later extended to ‘machine musicianship’ [16] so as to encompass musical skills that are complementary to performance.

In the years that followed, most research efforts concentrated in audio-to-score alignment of monophonic and polyphonic music, without however abandoning the ultimate ambition to develop real-time computer-based performers. In 2001, Raphael presents his Music-Plus-One system [17] for the first time. Music-Plus-One is currently available as a free software application that provides an orchestral accompaniment of a soloist using a big repertoire of recordings, which can be purchased online. It uses phase vocoder techniques to synchronize the orchestral recordings to the live solo, which is analyzed using HMM score following. In this work, the research focus is concentrated on predicting the future evolution of the live performance before it actually occurs. This type of prediction is necessary for allowing smooth synchronization between the soloist and the accompaniment. Without prediction, part of the note must be perceived before it is actually detectable by the employed algorithms, therefore leading to poor synchronization. Early approaches to guiding prediction used heuristic rules [18]. Raphael used Bayesian Belief Networks to predict the flow of live performance [19].

More recently, Dannenberg [3] classifies computer accompaniment systems under the more general term ‘Human Computer Music Performance’, referring to all forms of live music performance involving humans and computers. Consequently, computer accompaniment systems are integrated to a more general class of systems that use multiple input and output modalities (audio, visual, gesture) to support music performance. To this end, a new tendency has recently made its appearance as ‘co-player music robots’. For example in the work of Otsuka et al. [20], particle-filter score following of a human flutist is used to guide the Thereminist, a humanoid robot playing the Theremin [21].

Although research in computer accompaniment has a history of more than two decades, and it continuously progresses to new approaches and computational techniques, Human Computer Music Performance still remains a vision rather than a practice [3]. Hence, the progress made is not sufficient to address all types of complexities encountered in music performance and there are still many challenges to be met.

1.3 Computer Accompaniment over the Internet

Possibly the most similar research initiative to the approach presented in this paper is a system called 'TablaNet' [22]. TablaNet is a real-time online musical collaboration system for the tabla, a pair of North Indian hand drums. These two drums produce twelve pitched and unpitched sounds called *bols*. The system recognises *bols* using supervised training and k-means clustering on a set of features extracted from drum strokes. The recognised *bols* are subsequently sent as symbols over the network. A computer at the receiving end identifies the musical structure from the incoming sequence of symbols by mapping them dynamically to known musical constructs. To cope with transmission delays, the receiver predicts the next events by analyzing previous patterns before receiving the original events. This prediction is done using Dynamic Bayesian Networks. Finally, an audio output estimate is synthesized by triggering the playback of pre-recorded samples.

An alternative perspective has been presented for a networked piano duo [23]. In this approach MIDI generated from two MIDI pianos is matched to a score. Matching is achieved using the dynamic programming algorithm of Bloch and Dannenberg [24]. During matching, three types of deviations of the performance to the score are detected: tempo deviations (based on the detected inter-onset intervals), deviations in dynamics (based on the note velocity of MIDI messages) and articulations (based on note duration). Subsequently, these deviations are transmitted across the network and they are used to control a MIDI sequencer reproducing the score of the remote performer. Although this is an inspiring work in studying expressive aspects of music performance, it is not made clear why transmitting score deviations is more advantageous than sending the live MIDI stream of each pianist.

No further works have been found to specifically address real-time audio analysis and network transmission, neither for re-synthesis nor for informing performance context, to geographically dispersed music collaborators. Consequently, the perspective demonstrated in the current work provides a potential for advancing a new path of investigations, possibly revealing highly novel and previously undermined research challenges.

2 METHODOLOGY

A prototype application demonstrating the feasibility of the proposed approach has been implemented in C++. This application assumes that the signals exchanged through the network are mono-timbral, thus considering a single instrument located at each network location, as well as monophonic, i.e. no chords or polyphony is currently being treated.

The implementation of the proposed scheme comprises three functionalities, which are offline audio segmentation, score following and real-time audio rendering. Offline audio segmentation detects note boundaries on a pre-recorded solo performance of each musician and results in separate audio files, each containing the waveform of a different note. These files are used to render the live performance of each remote musician. Note boundaries are additionally used to train an HMM which is used by the score following functionality. Score following (a.k.a. real-time audio-to-score alignment) constitutes the listening-component. Specifically, during live performance, note onsets are detected by aligning the performance of each musician to the corresponding music score. Finally, real-time audio rendering is the core functionality of the perform-component, which concatenates note segments to re-synthesize the remote live performance. The following subsections describe the corresponding algorithms in more detail.

The methodology presented here is a follow-up of our previous work reported in [25]. The present paper extends that work by introducing certain algorithm improvements, a re-synthesis method to render the performance of remote musicians and some supplementary evaluation results.

For the moment, the algorithms presented in the following subsections operate on mono audio signals, sampled at 44.1 kHz with a sample resolution of 16 bit. These signals are partitioned in blocks of 512 samples (i.e. 11.6 ms), which was found to be a good compromise given the time constraints of the target application and the frequency resolution required for discriminating between different pitches. For the acquisition of spectral features, FFT uses zero padding to provide a better estimate of the dominant spectral peaks.

2.1 Offline Audio Segmentation

Given a solo recording of a monophonic instrument and the corresponding musical score (in the form of a MIDI file), this functionality aims at segmenting the recording at the time instants of note onsets. For this purpose an onset detection algorithm has been devised, which is particularly suited to the application at hand. The algorithm attempts to identify as many onsets in the recording as there are notes in the score.

Two acoustic features are facilitated for onset detection: a pitch value determined using a wavelet transform and a feature which is similar to Spectral Flux. The estimation of wavelet pitch is based on the algorithm of Maddox and Larson [26], which was found to give good pitch estimates for small block sizes. This feature is used to identify non-percussive onsets associated with subtle pitch changes. Subtle onsets are normally introduced by certain types of articulation such as legato playing. Onsets of this type are identified when a pitch value that is sustained for a number of blocks changes to a new pitch value which is also sustained for a certain number of blocks. It was experimentally found that requiring a constant pitch for 100ms before and after the change, successfully accounts for consecutive legato notes. This value is two times the psychoacoustic threshold of 50ms, within which perceptual discrimination of successive sounds becomes difficult [27]. Although the algorithm for pitch detection used here is causal, using it for onset detection necessitates processing audio blocks which follow the onset and is therefore inappropriate for online onset detection.

Subsequently to the identification of subtle onsets, the spectral flux feature is computed to account for salient onsets. This feature is used in the following form:

$$SF'(n)=\frac{\sum_{k=0}^{K-1}H(|X(n,k)|-|X(n-1,k)|)}{\sum_{k=0}^{K-1}|X(n,k)|} \tag{1}$$

In formula (1), |*X(n, k)*| is the spectral magnitude of the k^{th} bin of the n^{th} audio block and *K* is the total number of bins up to the Nyquist frequency. The function *H* is the half way rectifier function:

$$H(x)=\frac{x+|x|}{2} \tag{2}$$

The *SF′* feature is similar to spectral flux which has been previously used for onset detection by several researchers (e.g. [28]). However here, the spectral flux is divided by the sum of spectral magnitudes across all frequency bins. This serves to eliminate spurious detections due to increased signal energy.

SF′ is used as an Onset Detection Function (ODF) to identify salient, more percussive onsets, as the M top maxima of the ODF, assuming M is the number of percussive notes to be found (i.e. the number of notes on the score minus the number of legato onsets identified using pitch estimates). Knowing the number of onsets that need to be found increases the robustness of peak-picking on the ODF. Moreover, for each maximum value of the ODF, if it appears very close (within a minimum allowed Inter-Onset-Interval) to a previously identified onset it is discarded from the list of potential candidates. Additionally, if a maximum *SF′* value is followed by silence (defined by a maximum threshold of the Log Energy feature), it is also discarded. Every time a potential candidate is discarded, the ODF is searched again for the next maximum value, until the number of onsets that need to be detected has been attained.

2.2 Score Following

Score following uses Hidden Markov Models (HMM) to identify note onsets in real-time, on the live audio stream. The HMM uses the topology depicted in Fig. 1. Each note *n* is represented using three states, namely *Attack*, *Sustain* and *Rest*. The transition probabilities show that from each state it is possible to either depart to the next state or remain at the same state. The only exception is for *Sustain* states for which the *Rest* that follows may be skipped so as to account for legato playing.

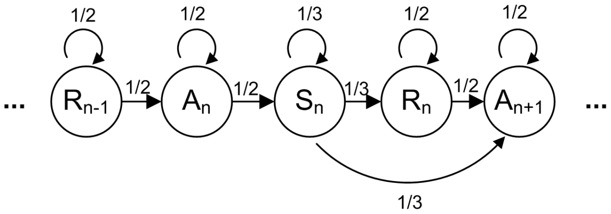

Figure 1: The HMM topology

Observations are generated by computing the following features per audio block:

- Log Energy and its first order difference
- Spectral Activity as defined in [29]
- Spectral Flux as defined in formula (1)
- Peak Structure Match [29] and its first order difference, for each pitch found in the score. This is in fact the ratio of the energy contained in the harmonic structure of a specific pitch frequency, compared to the entire energy of the audio block.

Observation probabilities are computed using an L-multivariate Gaussian, where L is the number of features. L depends on the number of distinct pitches appearing in the score of each solo part. Consequently, assuming N HMM states, the model comprises an NxN transition matrix, an NxL mean observation matrix and an LxL covariance matrix depicting inter-feature correlations.

Probabilities are trained, prior to live performance, using the Baum-Welch algorithm applied on the solo recording. In respect with training, it is well known that HMMs having a left-right topology, as in this case cannot be trained using a single observation sequence [30]. This is because for each state, the transitions departing from that state will follow a single path therefore yielding very low probability to alternative, however possible, paths. Hence, in order to have sufficient data to make reliable estimates one needs to use multiple observation sequences. Because of this, and due to the fact that in the reference scenario, only a single performance (i.e. the solo recording) is available as a training sequence, the current implementation for score following does not train transition probabilities. The Baum-Welch algorithm is only used to train observation probabilities. However in a more elaborate scenario, recordings obtained during offline rehearsals can be incorporated in training the model, therefore providing a better estimate for all types of probabilities.

A further issue related to HMM training concerns the fact that, although the Baum-Welch is an unsupervised training algorithm, correct initialization of model parameters (i.e. probabilities) prior to training is crucial to the performance of the model after training and even more so when dealing with continuous system observations [30]. Signal features correspond to continuous observations as opposed to discrete observations symbols derived from a finite set of possible values. In the past, different strategies have been employed to address this problem. Specifically, for the task of audio-to-score alignment, Cont [29] used the Yin algorithm [31] for blind pitch detection to discriminate among different pitch classes informing score states. An alternative approach to initialisation could be to synthesize an audio waveform from the score, using a software program or an API such as Timidity++[1] and initialize the model (i.e. compute HMM probabilities) according to the synthesized waveform, which accurately follows the MIDI file (see for example [32] on a similar application of Timidity++). In the approach presented in this paper, the note boundaries identified during the offline segmentation are used to provide an alignment of the recording to the score, hence initialising HMM probabilities prior to Baum-Welch training.

Finally, the trained HMM is used to detect the occurrence of a note onset during live performance. HMM decoding uses the Viterbi algorithm which is generally used to compute the optimal alignment path between two sequences. . In this case, the sequences are the HMM states defined by the score (Fig. 1) and the sequence of feature vectors describing the solo recording. The Viterbi algorithm is recursive and conventionally non-causal. In offline settings and with a well trained model, the algorithm yields very accurate alignments. For the system presented here, the Viterbi algorithm has been modified to operate causally, in other words to compute the HMM state of each audio block using knowledge only of previous blocks. Therefore skipping the termination and backtracking step of the original algorithm (see [30]). A further optimization employed here which does not affect the performance of the causal algorithm, concerns the fact that for each audio block only the observation probabilities for neighboring HMM states (±2 states) of the state identified in the audio previous block are computed. This is permitted due to the HMM topology used here, which gives zero probability for skipping more than one state. This optimization significantly reduces the complexity of the algorithm, as the estimation of observation probabilities based on multivariate Gaussians involves the computation of a large number of exponents (or logarithms, depending on implementation).

2.3 Real-time Audio rendering

The music played at each network location is rendered remotely by concatenating the note segments of the solo recording of the corresponding musician. When re-synthesizing an audio stream from audio units, as in concatenative synthesis approaches, different types of unit transformations, such as pitch, amplitude and duration, may be required prior to concatenation [33]. In the scenario considered here, namely synthesizing the performance of a piece having a predefined score and a pre-existing recording only amplitude and duration transformations are necessary. Seen from the perspective of expressive performance, a performer may alter the interpretation of a music piece in terms of loudness (hence requiring amplitude transformations), tempo (hence requiring different spacing between note onsets) and articulation. Transformations in articulation are more difficult to address as, at the simplest case, they would require detecting the time instant of note releases, a task that is even more error prone than that of onset detection.

In the present methodology, segment concatenation needs to take place as soon as the onset is remotely detected and before the end of the note. Hence, a mechanism predicting the expected loudness and duration of each upcoming note needs to be incorporated. This prediction may be based on the loudness and duration of the previous notes and thus these properties need to be monitored during performance. Overall, the audio rendering process involves three phases, which are performance monitoring and future event estimation, segment transformations and finally segment concatenation.

As already mentioned, performance monitoring and the extrapolation of the future evolution of a music piece is a central issue in computer accompaniment systems. In the present methodology, the actual quantities being monitored are the Root Mean Square (RMS) amplitude, and the Inter-Onset-Interval (IOI) of each note. The computation of these quantities is performed when the onset of the next note appears and therefore the previous note is assumed to have terminated. As soon as RMS and IOI are computed for the note just passed, they are communicated to remote network collaborators as low bandwidth information associated with the occurrence of the onset of the current note. At the receiving

[1] http://timidity.sourceforge.net/

location, these values for RMS and IOI are compared to the RMS and IOI of the corresponding note segment (of the pre-existing solo recording), yielding two ratios depicting RMS and IOI deviations of the live performance to the solo recording. These ratios correspond to the required gain and time scaling factor for the previous note.

Subsequently, the expected gain and time-scaling factor of the current note is estimated as the average of these values for the past four notes. The number of notes over which the average values are estimated can change or remain constant over the duration of the piece. For instance, these averages may be estimated using all previous notes or they may be based on the preceding four or five notes to account for the fact that deviations in tempo and dynamics can be constant within music phrases, but varying over the entire duration of the music piece. Another possibility would be to compute a weighted mean, such as a recursive average for which more recent notes have a greater influence to the prediction. For the moment, computing mean values over the past four notes appeared to give satisfactory estimates. Clearly, these techniques provide very rough estimates and are not literally predictive, as no probabilities are involved in the computation of future estimates. A more sophisticated mechanism for making predictions in expressive performance needs to be incorporated. This issue is addressed in current and ongoing research efforts.

Transforming the amplitude is achieved by multiplying the entire segment by the estimated gain factor. As for duration transformations, again knowledge from the score is exploited so as to make the process more efficient. Specifically, given the original duration of the note segment and the estimated factor for time-scaling, a new duration is computed. The score is used to provide the pitch of the current note segment and time scaling is performed pitch-synchronously. As the signals to be transformed are monophonic, the part of the note following the initial transient (of the order of 50ms to 100ms) is assumed to be periodic. Both when time stretching as well as when time shrinking the first part of the segment is left unprocessed, as it is assumed to carry the initial transient of the note. Initial transients should remain unprocessed to time/pitch scaling operations due to two reasons. Firstly, because they are generally non-periodic and secondly because the initial transients are related to the sound production mechanism of acoustic instruments and are thus important in terms of timbre perception. Moreover, as they always span a small region of the signal, time scaling initial transients would result in an unnatural acoustic effect. Excluding transients in time-scaling transformations is an established technique, addressed for example in [34].

According to the determined time scaling factor and the pitch period, a number of periods that need to be inserted or removed from the note segment is estimated. Subsequently, that number of periods is inserted or removed from the part of the note following the initial transient. Insertions and removals are distributed uniformly within the duration of the original note segment, so as to more effectively retain the shape of the amplitude envelop. For example, if 5 periods need to be inserted/removed for a total of 50 periods contained in the original note segment, then those are first period following the initial transient (the number of samples is determined by the pitch of the note) as well as the periods (i.e. the same number of samples) appearing after 10 periods from the previous insertion or removal. This approach may be considered analogous to PSOLA techniques, but without the overlap-add step [35].

The result of this technique for time scaling is shown on Fig. 2. The top waveform shows the original segment, the middle waveform shows the same segment stretched by a factor of 2.23 while the bottom waveform is shrinked by a factor of 0.73. The vertical dotted lines show the end of the initial transient. Up to that point the three waveforms are identical.

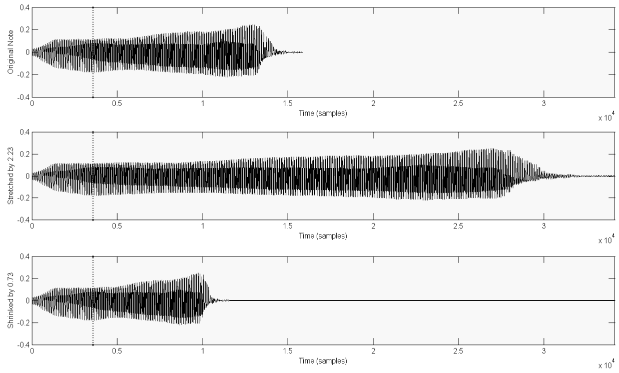

Figure 2: Pitch synchronous time domain transformations.

Subsequently, the transformed segment is concatenated to the transformed segment of the previous note. During concatenation a short amplitude cross-fade is applied on a single audio block (i.e. of 512 samples) so as to eliminate signal discontinuities that would result in perceivable click-distortions.

3 EXPERIMENTAL VALIDATION

Evaluation experiments are currently is progress. This section presents some results on a small dataset of twelve recordings of monophonic instruments accompanied by the corresponding MIDI files (i.e. the scores). These recordings have been manually annotated to provide the ground-truth onsets. Although this is far from being a formal evaluation, these experiments demonstrate a number of strong and weak points of the algorithms presented here.

Table 1 shows the results of this ‘mini-evaluation’. The evaluation experiments follow the standard MIREX[2] evaluation measures for the tasks of onset detection and real-time audio to score alignment. Due to limited space, only the most informative measures are reported in this paper.

idx	File	# notes	F_1	F_2	F_3	Abs. Avg Offset (ms)	Avg. Latency (ms)
1	Flute1	24	1	1	0.96	12.6	2.3
2	Flute2	26	1	0.92	0.88	11.8	3.0
3	Tenor Sax	9	1	0.88	0.88	10.1	2.6
4	Bassoon	36	0.94	0.57	0.96	5.6	2.4
5	Trumpet1	24	1	1	1	20.8	3.0
6	Trumpet2	24	1	0.92	1	12.1	2.5
7	Horn	42	0.86	0.76	0.77	17.9	2.5
8	Trombone	23	0.96	0.68	0.91	7.9	2.3
9	Violin	36	0.94	0.73	0.67	7.9	2.0
10	Viola	32	0.95	0.65	0.66	12.4	2.4
11	Guitar	25	0.88	0.8	0.7	16.0	2.8
12	Kick Drum	25	1	0.87	0.83	4.7	2.0
TOTAL/AVG		**326**	**0.96**	**0.82**	**0.85**	**11.7**	**2.5**

Table 1: Evaluation results for the tasks of offline onset detection and HMM score following.

The columns F1, F2, F3 refer to F-measures in the three cases: (1) offline onset detection, (2) real-time HMM alignment without training and (3) real-time HMM alignment after Baum-Welch training. Correct detections use a tolerance of 50ms around the ground truth onset. Due to the lack of multiple performances of the same piece of music by the same instrument, the same waveform was used in all three cases. The column entitled “# notes” contains the total number of notes of the audio and score file, while the measures ‘Abs. Avg. Offset’ and ‘Avg. Latency’ both refer to HMM alignment after Baum-Welch training (case 3). ‘Abs. Avg. Offset’ is the average of the absolute offset between the detected and the ground truth onset, while ‘Avg. Latency’ refers to the average of the time elapsed between the arrival of an audio block and the rendering of the synthesized block in the current prototype. It is important to pinpoint that this is not the latency of the score follower. Instead it is intended for comparison with the so called Ensemble Performance Threshold (EPT), which defines a psychoacoustic limit in music communication latencies during performance. The EPT is estimated to be of the order of 20-40ms one way latency [36] and in NMP settings it defines the total tolerable communication latency, including buffering, processing and transmission delays. For these experiments, capturing and playback occurs on the same machine (a Lenovo ThinkPad with an Intel Core Duo 2GHz processor, 2GB RAM PC with a CentOS Linux distribution), while the application uses the Jack Audio Connection Kit[3] for receiving audio input and sending output to the sound card in real-time.

F-measures are additionally depicted in Fig. 3. It can be seen that the performance of the offline onset detection algorithm determines and is always superior to the performance of the other two real-time detection methods. This provides evidence for the fact that precise model initialization is crucial when dealing with continuous observations, as previously discussed. Moreover, the diagram shows that Baum-Welch training improves the performance of the alignment in most cases, without however exceeding the performance of the onset detection algorithm.

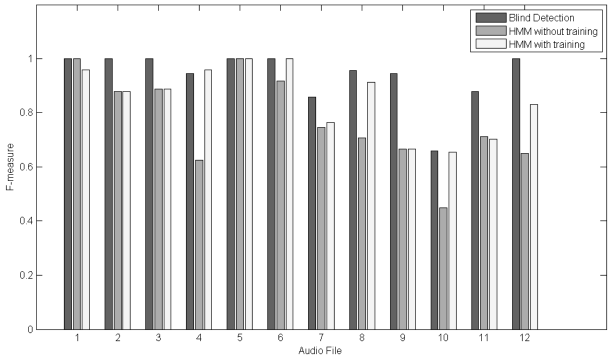

Figure 3: Comparison of the three methodologies for onset detection.

In respect with latencies, it can be seen that the process of audio capturing, real-time HMM decoding, audio segment transformation and concatenation does not introduce significant latencies. The average processing latency is of the order of 2.5 ms per note event, and does not significantly contribute to the end-to-end communication latencies. However, the average offset values for correct onset detections are significantly larger, yielding an average value of 11.7ms, and may considerably affect the quality of communication during performance. The offset in the arrival of note attacks becomes significant in cases where the music performed has a fast tempo, requires rhythmic synchronization or involves percussive instruments, in which cases the EPT should not exceed the value of 30 ms [36].

This issue as well as the real-time audio rendering technique presented above need to be further investigated by conducting a formal user evaluation involving dislocated music performers and psychoacoustic experiments.

4 CONCLUSIONS AND FUTURE WORK

This paper presented a novel scheme for real-time music communication over computer networks. The experimental validation shows that it is feasible to

[2] http://www.music-ir.org/mirex

[3] http://jackaudio.org/

employ this scheme for efficient low-latency and high-quality communication of music, thereby eliminating the need for audio stream exchange.

Clearly, the current implementation is far from offering a working product. The employed algorithms need several optimizations and improvements not only in terms of algorithmic performance but more importantly in terms of enabling more realistic performance scenarios. Specifically, one of the main deficiencies of the current prototype is the fact that music performers are assumed to precisely interpret the score without any errors. Clearly this is rather an ideal situation that rarely ever occurs. Our algorithms need to take into account performance errors as well as the fact that, in cases where collaboration occurs for the purposes of a music rehearsal or an improvisation session, musicians will occasionally stop before the end of a music piece or repeatedly perform certain parts of the music score. We expect to address this issue by improving the HMM training algorithm to automatically learn from several audio streams, so as to be able to detect performance errors as well as arbitrary music pieces. In fact, we envision a system which will be able to progressively learn and recognize the individualities of different instruments and different performers, though continuous use.

A further improvement concerns the possibility of accommodating polyphonic and possibly multi-timbral music, therefore enabling remote music concatenation for arbitrary instruments and music pieces. We are currently investigating the possibility of incorporating chords in our model.

Finally, the integration of the proposed methodology to a functional NMP software platform will allow for conducting user experiments with human performers that will further inform future enhancements.

ACKNOWLEDGEMENTS

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES, Project: MusiNet.

REFERENCES

[1] M. Delgado, W. Fajardo and M. Molina-Solana, “A state of the art on computational music performance,” *Expert Systems with Applications* vol. 38, no. 1, pp. 155-160 (2011).

[2] R. B. Dannenberg and C. Raphael, “Music score alignment and computer accompaniment,” *Communications of the ACM* vol. 49, no. 8, pp. 38 (2006).

[3] R.B. Dannenberg, “Human Computer Music Performance”. In *Multimodal Music Processing*, edited by Müller M., Goto M., Schedl M., 121-133, Wadern: Dagstuhl - Leibniz Center for computer science GmbH (2012).

[4] A. Carôt, A. and C. Werner, “Network Music Performance—Problems, Approaches and Perspectives.” *Proceedings of the Music in the Global Village Conference.* Available on-line: http://globalvillagemusic.net/2007/wp-content/uploads/carot_paper.pdf (2007).

[5] A. Carôt, P. Rebelo and A. Renaud, “Networked Music Performance: State of the Art.” *Proceedings of the AES 30th International Conference*, pp. 16-22 (2007).

[6] A. Kapur, G. Wang and P. Cook, “Interactive Network Performance: a dream worth dreaming?,” *Organised Sound* vol. 10, no. 3, pp. 209-219 (2005).

[7] M. Goto, R. Neyama Y. Muraoka, “RMCP: Remote Music Control Protocol – design and Interactive Network Performance applications,” *Proc. of the 1997 Int. Computer Music Conf.*, pp. 446–449 (1997).

[8] M. Wright and A. Freed, “Open Sound Control: a new protocol for communicating with sound synthesizers,” *Proc. of the 1997 Int. Computer Music Conf.*, pp. 101–104 (1997).

[9] A. Xu, et al, “Real time streaming of multi-channel audio data through the Internet,” *Journal of the Audio Engineering Society* vol. 48 no.7/8, pp. 627-641 (2000).

[10] J.P. Cáceres and C. Chafe, “JackTrip: Under the hood of an engine for network audio.” *Journal of New Music Research* vol. 39 no.3, pp. 183-187 (2010).

[11] R. Zimmermann, et al, “Distributed Musical Performances: Architecture and Stream Management,” *ACM Transactions on Multimedia Computing Communications and Applications* vol. 4, no. 2, article. 14 (2008)

[12] C. Alexandraki and D. Akoumianakis, “Exploring New Perspectives in Network Music Performance: The DIAMOUSES Framework,” *Computer Music Journal* vol. 34, no. 2, pp. 66-83.

[13] U. Kraemer, et al., “Network Music Performance with Ultra-Low-Delay Audio Coding under Unreliable Network Conditions.” *Proceedings of the 123rd Audio Engineering Society Convention*, pp. 338–348 (2007).

[14] B.L. Vercoe, “The Synthetic Performer in the Context of Live Performance,” in *Proceedings, International Computer Music Conference, Paris,* pp. 199-200 (1984).

[15] R.B. Dannenberg, “An On-Line Algorithm for Real-Time Accompaniment,” *Proceedings of the 1984 International Computer Music Conference*, pp. 193-198 (1984).

[16] R. Rowe, *Machine Musicianship,* Cambridge, MA: The MIT Press (2001).

[17] C. Raphael, “Music Plus One: A System for Expressive and Flexible Musical Accompaniment,” In *Proceedings of the International Computer Music Conference*, pp. 159-162 (2001).

[18] R.B. Dannenberg, “Real-Time Scheduling and Computer Accompaniment.” In Mathews, M.and Pierce, J. eds. *Current Research in Computer Music*, MIT Press, Cambridge, pp. 225-261 (1989).

[19] C. Raphael, “A Bayesian Network for Real-Time Musical Accompaniment.” In *Proceedings of Advanced in Neural Information Processing Systems*, pp. 1433-1440 (2001).

[20] T. Otsuka, et al, “Real-Time Audio-to-Score Alignment Using Particle Filter for Coplayer Music Robots,” *EURASIP Journal on Advances in Signal Processing* (2011).

[21] T. Mizumoto, et al, "Thereminist robot: Development of a robot theremin player with feedforward and feedback arm control based on a theremin's pitch model," in *IEEE Intl. Conf. on Intelligent Robots and Systems*, pp. 2297-2302 (2009).

[22] M. Sarkar and B. Vercoe, “Recognition and prediction in a network music performance system for Indian percussion” *Proceedings of the 7th international conference on New interfaces for musical expression NIME 07,* pp. 317-320 (2007).

[23] A. Hadjakos, E. Aitenbichler and M. Mühlhäuser, "Parameter Controlled Remote Performance (PCRP): Playing Together Despite High Delay". In *Proceedings of the International Computer Music Conference,* pp. 259-264 (2008).

[24] J. P. Bloch. and R.B. Dannenberg, “Real-Time Computer Accompaniment of Keyboard Performances.” In *Proceedings of the 1985 International Computer Music Conference,* pp. 279-289 (1985).

[25] C. Alexandraki C. and R. Bader. 2013. "Real-time concatenative synthesis for networked musical interactions," *Proceedings of Meetings on Acoustics*, 19, art. no. 035040, 9 p.

[26] R.K. Maddox and E. Larson, “Real-time time-domain pitch tracking using wavelets,” http://courses.physics.illinois.edu/phys406/NSF_REU_Reports/2005_reu/Real-Time_Time-Domain_Pitch_Tracking_Using_Wavelets.pdf (2005).

[27] Bregman A. “Auditory Scene Analysis: The Perceptual Organization of Sound”, MIT Press, (1990).

[28] S. Dixon, “Onset detection revisited.” *In Proc of the Int Conf on Digital Audio Effects DAFx06*, pp. 133–137 (2006).

[29] A. Cont, “Improvement of Observation Modeling for Score Following.” IRCAM. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.4970&rep=rep1&type=pdf (2004)

[30] L.R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” *Proceedings of the IEEE* vol. 77 no. 2, pp. 257–285 (1989).

[31] A. de Cheveigné and H. Kawahara, “YIN, A Fundamental Frequency Estimator for Speech and Music,” *Journal of the Acoustical Society of America*, vol. 111, pp. 1917–1930 (2002).

[32] N. Hu, R.B. Dannenberg and G Tzanetakis, “Polyphonic audio matching and alignment for music retrieval.” In 2003 *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 185-188 (2003).

[33] E. Maestre, et al, “Expressive Concatenative Synthesis by Reusing Samples from Real Performance Recordings,” *Computer Music Journal* vol. 33, no. 4, pp. 23-42 (2009).

[34] A. von dem Knesebeck, P. Ziraksaz, and U. Zölzer, “High quality time-domain pitch shifting using PSOLA and transient preservation,” in *Proc. 129th Audio Eng. Soc. Convention*, paper 8202 (2010).

[35] S. Roucos and A. Wilgus, “High quality time-scale modification for speech,” In Proceedings *of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp: 493–496 (1985).

[36] N. Schuett, “The effects of latency on ensemble performance.”Available at: https://ccrma. stanford.edu/groups/soundwire/publications/papers/schuett_honorThesis2002.pdf