

# Human-Computer Music Performance: From Synchronized Accompaniment to Musical Partner

**Roger B. Dannenberg,  
Zeyu Jin**

Carnegie Mellon  
University  
rbd@cs.cmu.edu  
zeyuj  
@andrew.cmu.edu

**Nicolas E. Gold,  
Octav-Emilian Sandu,  
Praneeth N. Palliyaguru**

University College London  
{n.gold,praneeth.palliya  
guru.10,octav-emilian.  
sandu.10}@ucl.ac.uk

**Andrew Robertson,  
Adam Stark**

Queen Mary  
University of London  
{andrew.robertson,  
adam.stark}  
@eecs.qmul.ac.uk

**Rebecca  
Kleinberger**

Massachusetts  
Institute of  
Technology  
rebklein  
@mit.edu

## ABSTRACT

Live music performance with computers has motivated many research projects in science, engineering, and the arts. In spite of decades of work, it is surprising that there is not more technology for, and a better understanding of the computer as music performer. We review the development of techniques for live music performance and outline our efforts to establish a new direction, *Human-Computer Music Performance* (HCMP), as a framework for a variety of coordinated studies. Our work in this area spans performance analysis, synchronization techniques, and interactive performance systems. Our goal is to enable musicians to incorporate computers into performances easily and effectively through a better understanding of requirements, new techniques, and practical, performance-worthy implementations. We conclude with directions for future work.

## 1. INTRODUCTION

Live performances increasingly use computer technology to augment acoustic or amplified acoustic instruments. The use of electronics in performance predates computing by many years, and there are many different conceptual approaches. The most obvious and popular approach is the simple replacement of acoustic instruments with digital ones. Another popular approach is the use of interactive systems that mainly react to input from human performers. In these systems, humans effectively “trigger” sound events or processes.

Two key aspects of live performance with computers are autonomy and synchronization. *Autonomy* refers to the ability of the computer performer to operate without direct control by a human, and *synchronization* refers to the ability to adapt a performance to the timing of humans. For example, interactive systems that are triggered by live performers are autonomous because they require little or no direct human control, and their synchronization is limited to computed responses to live events.

As we consider other forms of music, particularly traditional musical forms with scores and multiple parts, syn-

chronization becomes essential. Performances with fixed recordings are used in many settings, but these are uncomfortable because they place the entire synchronization burden on humans. One of the promises of real-time sound synthesis was to eliminate fixed recordings, creating an opportunity to actively and adaptively synchronize computers to humans [1, 2].

An early system to address computer synchronization to live performers was the Sequential Drum [3]. The Sequential Drum assumes that a sequence of sound events to be played is mostly known in advance, but the timing and perhaps other parameters such as loudness are determined at performance time. A performer uses a drum-like interface where each drum stroke launches the next sound event in the sequence and perhaps also controls loudness and other parameters. A drawback of the Sequential Drum is its lack of autonomy – it requires a human’s full attention during a performance.

Conducting systems are related to the Sequential Drum and a common theme in computer music research [4]. If a conductor exists anyway and a computer can follow the conductor’s gestures, the computer could be considered an autonomous performer. Synchronization requires that the computer sense not only beats and tempo but start times and other cues as well. In practice, computers cannot follow “real” conducting intended for humans, but there is promise that conducting gestures can offer one mode of synchronization.

The difficulty of following conductors was one inspiration for Computer Accompaniment (CA) systems [5], which use score matching to synchronize computer accompanists to live performers. CA is autonomous and can synchronize to traditional scored music with high reliability. There are, however, some drawbacks. First, CA requires a score and for players to follow the score. Improvisation and rhythmic variation lead to timing problems, if not outright failures. Second, CA requires distinctive input to follow. When the followed instrument holds a long note or rests, there is no synchronization information. It is possible to follow multiple instruments [6], but this adds to the complexity. Finally, CA often has limited timing accuracy due to problems of accurate onset detection. CA generally works well for chamber music with expressive timing, but not well for different forms of popular music.

It is surprising that systems offering *autonomy* and *synchronization* for popular music performance have not

been pursued more actively. Our goal is to create computer performers that play music with humans. We are particularly interested in music with fairly steady beats and where synchronization must be achieved through beats and measures rather than score following. This is a realistic problem that is characteristic of nearly all popular music, including rock, folk, jazz, and contemporary church music. It should be noted that score following systems are not a solution to this problem because (1) they require consistent playing at the note level to match to scores and (2) they do not synchronize with the precision required for steady tempo. The problem is broad in that it touches on music performance practice, music representation issues, machine listening, machine composition, human-computer interaction, sound synthesis, and sound diffusion. We refer to this overall direction as *Human-Computer Music Performance*, or HCMP.

Our goal here is to introduce the problems of HCMP, survey progress that we have made working together and individually, and describe future challenges and work to be done.

## 2. EXAMPLES OF HCMP SYSTEMS

To date, we have constructed a number of HCMP systems. The first system was a large project to create a virtual string orchestra to play with a jazz big band [7]. The system used tapping for synchronization, a small keyboard for cues (each key mapped to a different cue), and PSOLA [8] for time stretching a 20-track audio file in real-time. For this performance, an extra percussionist tapped her foot and entered cues.

This system was reimplemented and integrated with effects processing software and used by the first author in an experimental jazz quartet. This system was designed to be operated by the author who also plays trumpet in the quartet. Cues are given by a capacitive sensor worn on the index finger, and the system uses MIDI files rather than audio.

The B-Keeper system [9] is designed to follow the timing variations of a live drummer. Dedicated microphones are placed on the kick and snare drum, which are used to create an accurate representation of relevant drum events (Figure 1).



**Figure 1.** The band Higamos Hogamos performing with B-Keeper.

## 3. CHALLENGES OF HCMP

Active research is being carried out on many fronts. This section describes just a few interesting problems presented by HCMP and some of the approaches to solving them.

### 3.1 Detecting the Beat

Beat tracking algorithms aim to output the times of the tactus, the regular pulse with which humans would naturally tap in time with the music. Most beat tracking algorithms first process the signal to create an onset detection function [10]. Methods such as autocorrelation, comb filtering and interval clustering can be used to detect periodicity in this signal. The algorithm must also determine the phase, typically using dynamic programming or probabilistic methods, with the premise that musical events which correspond to peaks in the detection function are most likely to occur on the beat.

Real-time beat tracking algorithms might be used to provide a tempo and phase estimate for the underlying beat which can be used to synchronise HCMP systems. Whilst offline beat tracking algorithms have access to the full audio file and can operate non-causally, beat trackers for live performance must operate causally in real time. Examples of real-time algorithms released as external objects for MaxMSP are: btrack~ [11], beatcomber~ [12] and IBT~ [13].

Beat trackers are relatively successful on rock and pop examples, although they can exhibit errors such as tapping on the offbeat and tapping at double or half the tempo (octave errors). Complex passages, such as those featuring syncopation, can be problematic. Where the tempo changes, there is an inherent trade-off between reliability and responsiveness [14]. However, for a successful HCMP system, performers require full confidence that the system will behave as they expect.

An alternative to sensing the beat in audio is sensing the beat from foot-tapping or other gestures. We have successfully used a foot pedal in a number of performances and studied the foot pedal as an interface for communicating beat timing to a computer. In our measurements, the standard deviation of foot tap times is about 40 ms [15]. This alone is not satisfactory for music with a steady tempo, but we use the “steady tempo” feature to our advantage by using regression over previous beat times to predict the tempo and next beat time.

One of the difficult problems of tempo estimation is that tempo is normally steady but changes rather rapidly at times. We can minimize *average* error by using long regression windows, e.g. performing linear regression on the 20 previous beats, but then the *worst case* error where the tempo changes will be musically unacceptable. On the other hand, optimizing for the worst case tends to highlight special cases, often where synchronization is not musically necessary. In practice, we compromise with a window size of 5 to 7 previous tap times to predict the next tap time. This choice is sufficiently responsive that good synchronization, even in rock music, can be achieved, but it does require careful tapping. Some practice and musical skills are necessary, and the system has

less-than-ideal autonomy, but this method can be reliable and effective.

We are also considering additional modes of acquisition (e.g. video) to augment audio analysis, drawing on instrumental technique (e.g. guitar strumming actions).

### 3.2 Score Representation

Synchronizing at the beat level is only the first step to musical synchronization. All performers need to be at the same musical position, e.g. “beat 2 of measure 5 of the chorus.” Before we can talk about synchronization at this level, we need a formal model of what synchronization means.

In traditional music theory, a score provides an unambiguous sequence of beats and measures. Scores also indicate what each player should play in a given beat. In popular music, scores are treated much more casually, and the mapping from score notation to performances is sometimes specified informally, e.g. “play a 4-bar introduction, play an extra chorus at the end.”

We could “solve” this problem by insisting on traditional scores, but the reality is that popular music performance often demands flexibility to adapt, even in the middle of a performance. It is not unusual for non-sectional changes also to be made, e.g. the band plays an extra measure by “intuition” before continuing with the next section. Systems that attempt to synchronize with a score need to identify the current score location within it. This requires identifying musical features in performance at the level at which the score is expressed. For example, a chord list requires chord identification, a lead sheet with melody may be able to use that in addition to chords, performances with lyrics may be able to follow the sung parts (using techniques such as that of Mauch et al. [16]).

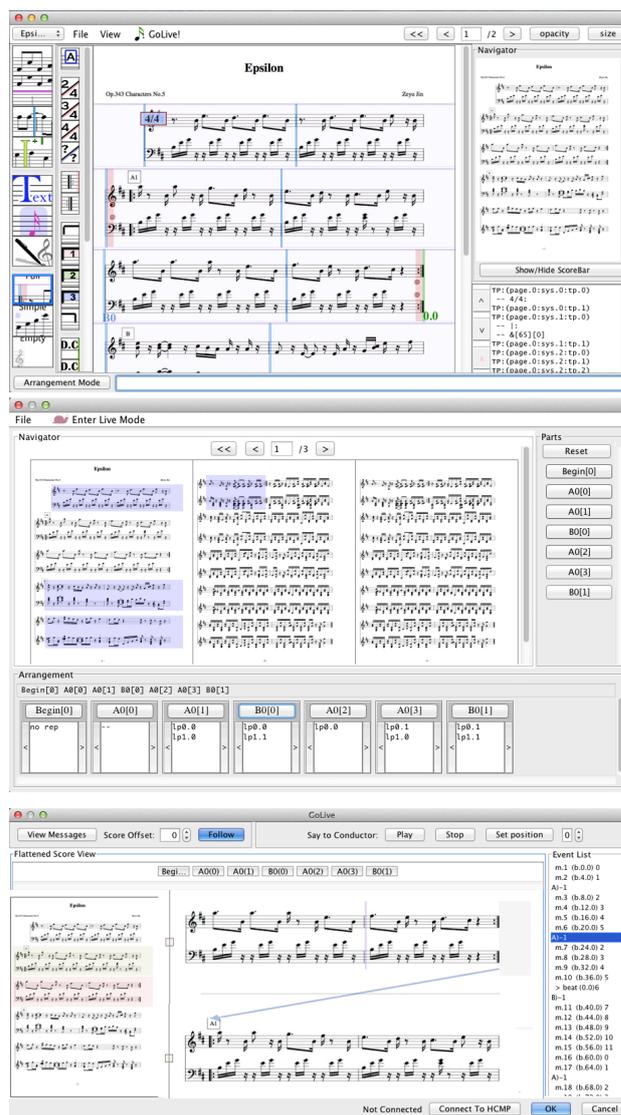
To address some of these problems, we have recently developed a music notation display system for HCMP. The system can import images or photos of music notation, thus leveraging existing printed music. Users can manually annotate the music images with control information to mark measures, time signatures, section names, repeats, codas, etc. (See top of Figure 2.) The system can then compute the normal performance order of the score, essentially “flattening” the repeats into a linear sequence. This flattened score provides a reference mapping from measure numbers back to score locations. This mapping can be shared across different media (audio players, MIDI players, visual displays) to coordinate them.

Another representation issue is that users may want to reorganize the score for a particular performance. We call this process “arrangement.” For example, an arrangement could be “play sections AABABA in that order, ignoring the structure implied by the score.” Figure 2 (middle) shows how an arrangement is constructed. The row of boxes represents an editable sequence of sections. Clicking on a box highlights the corresponding section in the score just above.

While this work solves many representation problems, more work is needed to communicate arrangements to computer players. Implementing jumps in audio or MIDI files is tricky (consider that sections may have pickup notes that precede the section and notes that sustain into

the next section). Ultimately, this illustrates the conceptual gap between human musicians who think of sections as high-level abstract objects to be realized in performance and computer players that model sections as immutable, concrete audio files or MIDI sequences. There are research opportunities here to raise the level of music abstraction offered by computers.

At performance time, the score is displayed in performance order using a “double-buffered” display allowing the performer to always look ahead in the score. (See Figure 2, bottom.) The performer can also use this display to give cues and indicate the current measure to the computer. This notation system is now complete, but work remains to integrate it with a performance system and to evaluate its use in live performances.



**Figure 2.** Digital music display system. Top is annotation system for indicating measures, sections, repeats, and other control structures. Middle is arrangement window showing original score and an editable sequence of sections constituting the arrangement. Bottom is live mode where score is displayed in performance order in real time with automatic page turning.

### 3.3 Synchronizing at Higher Levels

While our work on score representation offers a framework for coordinating performers, we still need to implement coordination. We propose the concept of cues as an approach to achieving synchronization in live performances. A cue is simply a signal that indicates a score location or directs a performer. Cues are typically given by humans to one another and take effect on the next section boundary. It is not uncommon to give cues many measures in advance because communication during a performance requires getting the attention of other performers and communication gestures may be unreliable.

In our systems, cues have several types [17]:

- *Position* cues indicate global position and indicate either when to start playing or that the computer and human(s) have diverged and need to resynchronize;
- *Intention* cues indicate a decision has been made about the future course of the performance; for example, this is the last repetition of a vamp;
- *Voicing* cues are not used for synchronization but indicate choices about how a player should render media. Volume changes or a request not to play can be indicated with voicing cues.

We have explored different means of giving cues. Our first system used a small MIDI keyboard where each key was labeled with a *position* cue that caused the computer to play a pre-recorded section of music. A later system used a wearable capacitive sensor attached to the index finger. By touching the sensor with the thumb, a cue can be given. This system detects cues reliably and does not intrude upon the human performer.

Currently, we are working on capturing gestures such as nodding your head in the direction of the computer performer. Detecting gestures from a continuous stream of sensor data is prone to false positives. We are evaluating the use of dynamic time warping and machine learning techniques to build a reliable system [18].

We are also exploring the potential of natural user interfaces to minimize disruption to performers. Ideally, a computer performer would not require explicit cuing but through understanding the performance norms of an ensemble and observing the gestures the human performers make, will be able to determine the intention and position cues for itself (i.e. full autonomy).

To that end, we have experimented with the Microsoft Kinect™ sensor as an interface for various applications including use as a conducting system to set initial tempo, as a way to determine intention cues through counting raised fingers on one hand (a practice used in contemporary church-music leading to direct the band to a numbered score section), to observe guitarists' actions, and as a way to automate page turning. The latter project detects head tilt gestures that control the direction of a page turn in a PDF file displayed on a computer or (soon) iPad™, controlled over a network.

In all cases, the major challenge is the robustness of detection in noisy, realistic scenarios. Distinguishing the neck of a guitar from a player's arm has proved difficult, even in "near-mode" where the sensor tracks only the upper half of the body. The sensor is also very sensitive to angle, making it potentially difficult to deploy in real-

istic scenarios. Music stands, piano lids, microphones and other normal musical equipment found in stage environments all work to confound the clear picture required for easy detection.

The page turning system (evaluated by two of the authors in a laboratory setting, one acting as pianist) works well with the sensor placed in front of or behind a pianist (although is very sensitive to off-axis placement – front is best). Unfortunately, this precludes the typical "forward" nod for page turns because neck movement in that plane is not currently detectable from those sensor positions. Other challenges include differentiating expressive movement from directive gesture.

We are also developing chord sequence recognition systems to identify a score section based on the chord sequence played thus far (similar to [19]). There are challenges in synchronizing the incoming chord sequence to the "model sequence" in the score, particularly in the presence of inaccurately played, missed, substituted or mis-identified chords, and difficulties arising from the need to define a chord with reference to a beat where the beat is itself defined with less than 100% accuracy. In addition, there are many examples of popular music where the chord sequence is so repetitive as to offer little information alone as to which of the sections is currently being played. In these cases, alternative cues such as the texture of the music may be helpful. Our distributed approach (described below) can also produce conflicting chord and beat identifications from different instruments that require resolution.

A broader level of synchronization (and context knowledge) may also be required. It is not unusual for ensembles to move seamlessly from one work (or part of a work) to another without forward planning (e.g. see Benford *et al.*'s study of Irish folk music sequencing [20]). Recognizing when this occurs and shifting to the new work is similar to recognizing sections in general, albeit with a larger range of potential sections to select from.

Finally, another important possibility for detecting cueing gestures is the digital score display system described in the previous section. An interesting aspect of this interface is that music notation can be bi-directional: The display can show the computer's location in the score to the reader using a cursor or highlighting graphical areas, and the reader can indicate his or her location to the computer by pointing to notation (e.g. measures) in the score.

### 3.4 Adjusting Tempo

What does the computer player play? One approach is to play pre-recorded audio, using time-stretching techniques to adjust the playback tempo. We constructed one HCMP system in which the computer played the role of a 20-piece string orchestra. Each string part was recorded on a separate track so that high-quality pitch-synchronous overlap-add (PSOLA) time stretching could be used [7]. Other techniques such as the phase vocoder can be used on polyphonic recordings [4].

Another approach is MIDI, since MIDI sequences are easily time-stretched. A challenge with MIDI is to simulate sounds of acoustic instruments. Sample-based syn-

thesis is good for isolated notes, but it has difficulty producing natural-sounding musical phrases. Progress has been made with large sample libraries and automated sample selection, but there is still much work to be done. Alternative techniques, including physical models and spectral models offer more flexible control, but expressive musical control is still an important problem.

Studies on latency in networked performance suggest that the just noticeable difference (JND), the latency setting at which the effect becomes noticeable to the performer, is between 20 and 30msec [21]. This also provides an estimate for the bounds within which the synchronization will be acceptable to human performers.

### 3.5 System Architecture

One lesson from building early systems is that robust interactive systems require careful design. The lack of a flexible program that supports new performances has hindered research, and we are working toward a more flexible, modular software architecture for HCMP.

HCMP systems decompose naturally into components:

- Input sensors for tapping and cueing,
- Beat and tempo estimation based on tapping or audio analysis,
- A virtual conductor that accepts position and tempo information and distributes it to players,
- Media players, including variable rate audio players based on time-stretching, and MIDI players.
- Score display (with automated page turning, position display)
- Development and configuration management system allowing users to combine media, define cues, make arrangements, and store everything so that it can be retrieved and used automatically in a performance.

We are developing components and plan to release a system based on “plug-ins” so that end-users can configure their own systems with just the features they need, and advanced users can extend the base system through scripting languages to provide custom features.

Our recent work [22] has shown that HCMP technology may be more likely to be adopted if it can be delivered quickly to users at low-cost and low-risk. We have therefore also been exploring the potential of mobile devices (such as smartphones) as a way to deliver HCMP systems. Each instrument in a band would be tracked by a smart device (e.g. resting on a music stand), undertaking its own audio processing and sending the results to a virtual conductor on another device for music generation.

This approach poses some interesting new problems. Since the audio processing for beat tracking and chord recognition is distributed among several devices, data fusion becomes paramount, especially in the absence of synchronous clocks. There are new opportunities also: textural detection may be easier (since the sound level can be more easily measured per instrument), beat tracking on an individual instrument may be better than on the ensemble as a whole (and could be based on knowledge of the individual instrument being tracked), and other device capabilities (e.g. video) may be usable. Additional equipment would not generally be needed by the users since we think it reasonable to assume that smart devices

would be widely available to ensembles through personal phones. Where new technology is required (e.g. for gesture tracking) we are seeking to use off-the-shelf consumer devices such as Kinect™ (as described above) to minimise deployment complexity and cost.

To expose the research issues, we have undertaken a feasibility study to evaluate interactive performance technologies on consumer devices and in realistic performance environments. The aims were to evaluate the difficulty of repackaging this technology for smartphone, to assess the musical performance issues raised by doing so (e.g. where should the smartphone be placed while performing?), and to understand challenges to the state of the art and shape the future development of such techniques. We repackaged existing state-of-the-art beat tracking [12] and chord-estimation [23] software into smartphone apps using libpd [24]. The app was deployed to several iOS devices (see Figure 3) linked by a wireless network.



Figure 3. iOS app for beat/chord detection.

A local church’s worship band area was used as a realistic physical evaluation environment, with a subset of the authors forming a band. Five genre-appropriate songs were used as test subjects and video, audio, and data from the systems were recorded. We also undertook “laboratory” evaluation using multi-track, multi-speaker recordings of the same songs to closely replicate live performance conditions, but allowing replication and experimental parameter control (see Figure 4). The gesture-control detection for band direction was also evaluated in this environment.



Figure 4. Simulation of live performance with multiple speakers and devices in a shared acoustic space.

Our analysis indicates that both platform and performance context provide significant challenges to state-of-the-art techniques. Problems include the distribution of tracking across multiple devices resulting in latency in reporting beats/chords, reconciling multiple independent timing streams, and the loss of the full mix at each tracker meaning the beat and chord tracking systems have less audio to work with. On the positive side, we also found evidence that beat tracking on an individual instrument track can outperform tracking on the whole mix.

## 4. HCMP EVALUATION

To measure progress, we need ways to evaluate HCMP systems. We have used a range of methods with varying levels of detail and rigour in the projects described in this paper. Evaluating HCMP research requires several approaches given the range of underpinning disciplines and potential outcomes. Interactive Music System (IMS) evaluation methods vary widely depending on the type of system and the particular interest of the researchers. Collins summarises three main evaluation forms for concert systems [25]: direct participant experience, indirect participant experience, and technical evaluation of algorithms. Hsu and Sosnick [26] address the first two aspects with a method based on the DECIDE framework. Stowell et al. [27] identify the difficulties in evaluating IMS, presenting and comparing qualitative and quantitative approaches including comparative listening tests, interaction analysis based on video, discourse analysis and the (somewhat controversial – see Ariza [28]) “musical Turing Test”. They offer useful guidance on the application of these techniques: Discourse analysis may be used to assess direct participant experience (i.e. the musicians themselves), and the musical Turing Test (in effect, survey methods) used to assess indirect participant experience (i.e. the non-musicians supported by, or listening to the music). Rowe states that programs for machine musicianship should exhibit behaviour that can be observed as correct or incorrect [29]. In the HCMP case passing the musical Turing Test will require at least satisfying (i.e. satisfactory and sufficient) output.

Stowell et al. acknowledge that most evaluation methods are focused on the experience of performers [27] (e.g. Hsu and Sosnick’s framework [26] does not address the third of Collins’ criteria), thus the evaluation of HCMP systems (particularly the sub-components) will need to be supplemented by objective criteria (e.g. measuring latency of interaction in comparison to experimentally-derived musical synchronisation criteria [30, 31], and measures of beat-tracking accuracy [32] and chord recognition [23]). Adoptability issues will also need to be addressed [22].

The ensemble nature of popular music means that other than low-level laboratory tests of system components, the main evaluation activity will need to involve groups of musicians (or simulation of this scenario). In addition to work with live bands, as shown above, multi-track recordings can be used to simulate the live environment by replaying performances through electrical or acoustical signal paths to multiple speakers and detection systems..

## 5. FUTURE WORK

Human musicians are often expected to improvise, or at least perform from lead sheets, which give music structure and chords but not the details of notes and rhythms. Since human musicians may not have the skills to construct drum beats, bass lines, and other parts, HCMP is an excellent domain to investigate automated music composition. Perhaps music analogies are an interesting way to specify goals, i.e. “I want a bass part that sounds like the one in ....” Similarly, parts must be performed expressively and musically. Perhaps there are synthesis-by-rule

approaches [33] or more general theories of expressive performance [34] for popular, beat-based music.

Ideally, an HCMP system would incorporate a learning mechanism that would allow it to extract useful information about the performance from rehearsals. This could make performances more reliable and more autonomous.

How can we evaluate general musicianship? Even synchronization is difficult to measure objectively: Once basic synchronization within 10 or 20 ms has been obtained, rhythmic “feel” that results from deliberate asynchrony [35] may be more important than precise synchronization. As we describe, a range of evaluation methods will likely be required, from measures of low-level performance (synchronization, chord identification) through to system-level evaluation methods involving performers considering their experiences alongside the systems, and audience-focused measures of reception. The standards required may vary depending on context: A computer that fills in for a human musician in a rehearsal or HCMP to facilitate practice at home may have modest requirements, while high-profile live performances in public may require virtuoso-level performance. The development of comprehensive and systematic top-to-bottom evaluation methodologies for HCMP is thus a key topic for future work.

## 6. CONCLUSIONS

HCMP has great potential to be widely used by many musicians. There are interesting scientific challenges as well as artistic ones. We are in the early stages of exploring possibilities and implementing systems that offer synchronization, interaction, and autonomy in live performance, with a focus on steady-tempo popular music, a problem which our research community has largely ignored. We believe that HCMP can become a practical, useful, and common way to make music, eventually used by millions. Ultimately, we hope that some of these users will leverage the unique properties of autonomous computer musicians to develop new musical genres.

### Acknowledgments

Parts of this work were undertaken while Kleinberger was at University College London. This work was partly supported by the Arts and Humanities Research Council [grant number AH/J012408/1]; the Engineering and Physical Sciences Research Council [grant number EP/G060525/2]; the Department of Computer Science, UCL; Microsoft Research; the Royal Academy of Engineering; and the National Science Foundation [grant number 0855958]. Depending on the proposed use, some data from the RCUK-funded portions of this work may be available by contacting Nicolas Gold. Please note that intellectual, copyright and performance rights issues may prevent the full disclosure of this data.

## 7. REFERENCES

- [1] R. B. Dannenberg, “An On-Line Algorithm for Real-Time Accompaniment,” in *Proc. Int. Computer Music Conf.* Paris, 1985, pp. 193-198.

- [2] B. Vercoe. "The Synthetic Performer in the Context of Live Performance," in *Proc. Int. Computer Music Conf.*, Paris, 1985.
- [3] M. V. Mathews and C. Abbott. "The Sequential Drum," *Computer Music Journal*, vol. 4, no. 4, pp. 45-59, 1980.
- [4] E. Lee, T. Karrer, and J. Borchers, "Toward a Framework for Interactive Systems to Conduct Digital Audio and Video Streams," *Computer Music Journal*, vol. 30, no. 1, pp. 21-36, 2006.
- [5] R. B. Dannenberg and C. Raphael, "Music Score Alignment and Computer Accompaniment," *Comm. of the ACM*, vol. 49, no. 8, pp. 38-43, 2006.
- [6] L. Grubb and R. B. Dannenberg, "Computer Performance in an Ensemble," in *Proc 3rd Int. Conf. for Music Perception and Cognition*, European Society for the Cognitive Sciences of Music, Liege, Belgium, 1994, pp. 57-60.
- [7] R. B. Dannenberg, "A Virtual Orchestra for Human-Computer Music Performance," in *Proc. Int. Computer Music Conf.*, pp. 185-188, 2011.
- [8] N. Schnell, G. Peeters, S. Lemouton, P. Manoury, X. Rodet, "Synthesizing a Choir in Real Time Using Pitch Synchronous Overlap Add (PSOLA)," in *Proc. Int. Computer Music Conf.*, Berlin, 2000.
- [9] A. Robertson and M. D. Plumbley, "B-Keeper: A beat tracker for real-time synchronization within performance," in *Proc. of New Interfaces for Musical Expression (NIME 2007)*, New York, 2007, pp 234-237.
- [10] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, M. B. Sandler, "A Tutorial on Onset Detection in Music Signals," in *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, 2005, pp. 1035-1047.
- [11] A. M. Stark, M. E. P. Davies and M. D. Plumbley, "Real-time beat synchronous analysis of musical signals," in *Proc. Digital Audio Effects Conf.*, 2009, pp. 299-304.
- [12] A. Robertson, A. Stark, and M. Plumbley, "Real-Time Visual Beat Tracking Using a Comb Filter Matrix," in *Proc. Int. Computer Music Conf.* Huddersfield, 2011.
- [13] J. Oliveira, F. Gouyon, L. G. Martins and L. P. Reis, "IBT: A Real-time Tempo and Beat Tracking System", in *Proc. Int. Symp. on Music Information Retrieval (ISMIR)*, 2010, pp. 291-296.
- [14] F. Gouyon, S. Dixon, "A Review of Automatic Rhythm Description Systems," *Computer Music Journal*, vol. 29, no. 1, pp.34-54, 2005.
- [15] R. B. Dannenberg, and L. Wasserman, "Estimating the Error Distribution of a Single Tap Sequence without Ground Truth" in *Proc. Int. Symp. on Music Information Retrieval (ISMIR 2009)*, 2009, pp. 297-302.
- [16] M. Mauch, H. Fujihara, and M. Goto "Lyrics-to-audio alignment and phrase-level segmentation using incomplete internet-style chord annotations," in *Proc. Sound and Music Computing Conf.*, Barcelona, 2010.
- [17] N. E. Gold and R. B. Dannenberg, "A Reference Architecture and Score Representation for Popular Music Human-Computer Music Performance Systems," in *Proc. New Interfaces For Musical Expression*, Oslo, 2011.
- [18] J. Tang *Extracting Commands From Gestures: Gesture Spotting and Recognition for Real-time Music Performance* (Master's Thesis), Carnegie Mellon Univ., 2013.
- [19] Z. Duan and B. Pardo, "Aligning Semi-Improvised Music Audio with Its Lead Sheet," *Proc. Int. Symp. on Music Information Retrieval*, Miami, 2011.
- [20] S. Benford, P. Tolmie, A.Y. Ahmed, A. Crabtree, T. Rodden, "Supporting Traditional Music-Making: Designing for Situated Discretion," in *Proc. ACM 2012 Conf. on Computer Supported Cooperative Work*, Seattle, 2012.
- [21] N.P. Lago, and F. Kon, "The Quest for Low Latency," in *Proc. Int. Computer Music Conf.*, 2004, pp. 33-36.
- [22] N. E. Gold, "A Framework to Evaluate the Adoption Potential of Interactive Performance Systems for Popular Music," in *Proc. Sound and Music Computing Conf.*, Copenhagen, 2012.
- [23] A. M. Stark and M. D. Plumbley, "Real-Time Chord Recognition for Live Performance," in *Proc. Int. Computer Music Conf.*, Montreal, 2009.
- [24] P. Brinkman, *Making Musical Apps*. O'Reilly, 2012.
- [25] N. Collins, *Introduction to Computer Music*. John Wiley & Sons, Inc., 2010.
- [26] W. Hsu and M. Sosnick, "Evaluating Interactive Music Systems: An HCI Approach," in *Proc. New Interfaces for Musical Expression*, Pittsburgh, PA, 2009.
- [27] D. Stowell, A. Robertson, N. Bryan-Kinns, M. D. Plumbley, "Evaluation of Live Human-Computer Music-Making: Quantitative and Qualitative Approaches," *International Journal of Human-Computer Studies*, vol. 67, no. 11, pp. 960-975, 2009.
- [28] C. Ariza, "The interrogator as critic: The Turing Test and the Evaluation of Generative Music Systems," *Computer Music Journal*, vol. 33, no. 2, pp. 48-70, 2009.
- [29] R. Rowe, *Machine Musicianship*. MIT Press. 2001.
- [30] M. Gurevich and C. Chafe. "Simulation of Networked Ensemble Performance with Varying Time Delays: Characterization of Ensemble Accuracy," in *Proc. Int. Computer Music Conference*, Miami, 2004.
- [31] R. A. Rasch. "Timing and Synchronization in Ensemble Performance," in *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition*. J.A. Sloboda, ed. Clarendon Press, pp. 70-90, 1988.
- [32] M. Davies, N. Degara, and M.D. Plumbley. 2009. Evaluation methods for musical audio beat tracking algorithms, *Technical Report C4DM-TR-09-06*, Centre for Digital Music, Queen Mary University of London.
- [33] J. Sundberg, A. Askenfelt and L. Frydén, "Musical Performance: A Synthesis-by-Rule Approach," *Computer Music Journal*, vol. 7, no. 1 (Spring), pp 37-43, 1983.
- [34] P. N. Juslin, A. Friberg, and R. Bresin, "Toward a Computational Model of Expression in Performance: The GERM model," *Musicae Scientiae*, Special issue, pp. 63-122, 2001-2002.
- [35] A. Friberg, and A. Sundström, "Swing Ratios and Ensemble Timing in Jazz Performance: Evidence for a Common Rhythmic Pattern," *Music Perception*, vol. 3, no. 19, pp. 333-349.