Approaches to Overcoming Problems in Interactive Musical Performance Systems

Cory McKay
Faculty of Music, McGill University
555 Sherbrooke Street West
Montreal, Quebec, Canada H3A 1E3
cory.mckay@mail.mcgill.ca

Abstract

Interactive musical performance systems have a great deal of potential in terms of the performance and compositional possibilities that they make available. Unfortunately, they also suffer from a number of difficulties, primarily with regard to problems with dissemination, longevity and software and hardware reliability.

This paper discusses issues that must be considered if one wishes to overcome these problems, and also proposes a number of solutions. Topics dealt with include concerns of human performers, software and hardware design priorities, the value of standardizing performance parameter extraction systems, the utility of a client-server architecture and the potential advantages of systems that extract performance parameters from performers' audio signals rather than from specialized sensors. These solutions are discussed in terms of both their strengths and weaknesses.

1. Introduction

Interactive performance systems (IPS's) consist of computer software and hardware systems that react musically to the actions of one or more human performers. Such systems can serve as accompanists for human performers, as improvisational partners, or even as soloists accompanied by humans. Multiple IPS's can also perform with each other as well as humans.

A very basic IPS, for example, could simply monitor performer actions and initiate pre-recorded events as they are triggered by the performer. Another approach is to form interactive mappings that do not necessarily rely on specific triggers. Examples include adding harmonisations to the notes played by a performer or altering a performer's timbre using pre-set heuristics. Alternatively, systems can use score-following techniques to follow along as humans perform pre-defined scores, and dynamically time warp pre-set accompaniments in order to stay in step with humans.

There are also much more sophisticated systems that make use of advanced expert systems or artificial intelligence to model sophisticated musical knowledge. Such systems are able to dynamically react to performers in non-deterministic and musically interesting ways, including, for example, improvising with a performer.

IPS's offer many musical possibilities that are not available to traditional acoustic, electric or tape-based performances. By acting as an artificial accompanist or soloist, these systems take advantage of the extended sonic and technical capabilities of computers, while at the same time capitalizing on the essential expressivity and live dynamism of the human performers involved. IPS's can also initiate original and unexpected musical ideas that can inspire human musicians to explore musical avenues that might not have otherwise have occurred to them. These systems also have important educational potential, as they can be used by student performers to gain experience performing in "ensembles" when other human performers are not available.

Unfortunately, IPS's also present a number of difficulties, particularly with regard to dissemination, longevity and software and hardware robustness and reliability. Composers and designers must consider the limitations as well as the potential of both the technology and the human performers involved in order to overcome these problems. Failure to do so can result in overly ambitions works that are prohibitively difficult to perform, have a dangerously high likelihood of failing during concerts and have limited potential for performance without the direct involvement of composers and designers.

2. Problems and solutions

Any survey of concert programs will reveal that IPS's are used in only a small portion of contemporary art music performances. Given the potential of these systems, it is important to understand why this is so. The limited proliferation of these systems has been noted elsewhere, most notably in the work of Bruce Pennycook (1997). This section reviews some of the problems of observed by Pennycook and expands upon them. Some fundamental solutions that could be used to improve matters are also presented.

To begin with, human performers are often asked to perform actions that are unnatural and excessively difficult. IPS's that demand too much of performers will not draw performer interest, something which will limit the likelihood of performers choosing to use a particular system, and thereby limiting its proliferation.

Human performers should not, in general, be expected to perform tasks that are alien to their training. They should not be asked to make gestures that are significantly different from the types of gestures that they are used to making or that will impinge on their ability to perform.

Performers should also be given easily and quickly comprehensible feedback from the performance system, whether visual, auditory or haptic. A clear explanation of how the system can or will react to their different types of actions should be given to them as well.

Furthermore, performers should be given sufficient time to learn how to use the system, including significant rehearsal time with the fully operational system. Failure to meet these needs of performers will likely influence the quality of their performance, and could well make performers reluctant to work with a system, particularly in the long term.

Composers who wish their interactive works to be performed in contexts that extend beyond situations involving their direct involvement must consider a further set of problems. IPS's must be robust enough that they do not require constant supervision during performance by someone intimately familiar with their implementations. They must be error tolerant to the extent that they will not be misled by minor performer mistakes and will not fail entirely if major mistakes occur. They should, in most cases, require minimal control by a technician, and what control is needed should be simple, intuitive and easy to learn. An easy-to-use interface and clear documentation are essential to a system if one wishes it to be disseminated and used.

Software designers can help mitigate some of these problems by implementing standardized software back-end systems that extract a wide range of potentially useful parameters from performers and output them in a standardized form that can be mapped in arbitrary ways by composers. The existence of such a system would free composers from needing to worry about extracting parameters, and would allow them to concentrate on designing mappings for their compositions.

More importantly, the mappings could easily be distributed to concert organizers who are familiar with the standardized parameter extraction system and have it pre-installed at concert venues. This kind of standardization could lead to an ease of transmission that would make it much easier for concert organizers to present concerts without the direct intervention of the composer which, in turn, could lead to wider dissemination and performance of composers' works.

It is suggested here that systems that automatically convert audio as it is produced by performers into simple symbolic transcriptions (e.g. MIDI) in real-time hold a great deal of potential in this respect. Such systems would allow performers to simply play their instruments in ways that they are accustomed to, with no need for unnatural gestures or additional intrusive and unnatural sensors that could be distracting and impinge on their freedom of movement. Miniaturization and wireless technology could be taken advantage of so that microphones, the only necessary sensors, could essentially be made invisible to performers by placing them in the barrel of a clarinet, for example, or on the inside of the soundboard of an acoustic guitar.

This approach would also allow composers to take advantage of the rich sound offered by traditional acoustic instruments directly if they wish. There would no longer be any need to use the potentially awkward artificial electronic MIDI instruments that have sometimes been used in order to acquire accurate symbolic data, and there would therefore be no need to rely on the limitations of synthesis imposed by the use of such instruments unless, of course, one wishes to.

In addition, the audio transcription approach not only provides the full range of symbolic control information that was previously only available from MIDI instruments or instruments equipped with often awkward sensors, but also allows one to make use of additional signal processing-based features that can be extracted from the acoustic audio signal. Parameters related to timbre, for example, could be very useful.

There are sometimes important creative advantages to using hyper-instruments (i.e. original instruments usually involving built-in sensors and electronics) that cannot be ignored, of course. Real-time transcription parameter extractors can certainly be used with such instruments, however, simply by micing the audio output of the instruments, whether acoustic or from a speaker. While this does require redundant processing of audio when parameter streams are likely available directly from the instrument, the advantages discussed elsewhere in this section can very well be seen as more than compensating for this extra processing load. Of course, the hyper-instruments themselves would need to be robust, portable, etc. in order to avoid compromising the advantages of the type of system proposed here, but any good hyper-instrument should already have these qualities as a matter of course.

A further current problem with the dissemination of interactive performance systems is that installation incompatibilities can cause significant problems when software is moved from one platform to another. This is a particular problem when attempts are made to use a system that was developed on old hardware or using an old operating system equipped with an obsolete software system. The experiences of Wuan-Chin Li (2004) when trying to use the Max patches from a fifteen-year old piece by Jean-Claude Risset are a good illustration of this problem. Software portability is therefore essential with respect to a composition's longevity.

In an effort to deal with this problem, it is suggested here that all IPS software be developed in a portable language such as Java, with a strict discipline enforced against system-specific calls. Java makes use of the Java Virtual Machine (JVM) to run Java bytecode rather than requiring system specific compilations of software. Furthermore, Java relies on the JVM to communicate with sound acquisition hardware, which means that Java code can be installed and used easily and effectively on any computer equipped with audio sampling hardware and the JVM.

An additional problem that has limited the dissemination and longevity of IPS's is the difficulties in acquiring, transporting and replacing specialized hardware. The basic physical simplicity and portability of equipment is therefore another important advantage to real-time transcription-based IPS's, as all that is needed are a computer, microphones and a performer's instrument.

3. Problems with real-time transcription

Automatic audio transcription is a relatively new technology, and is certainly far from perfectly refined. Although monophonic transcriptions can be performed fairly reliably, polyphonic transcriptions are often error prone.

This means that monophonic instruments such as flutes, for example, pose essentially no problem for IPS's based on real-time transcription. Ensembles of monophonic instruments are not problematic either, as a separate localized directional microphone can be used for each instrument in order to segregate the signals from different instruments. This effectively reduces a polyphonic transcription problem to several much easier monophonic transcription problems.

Unfortunately, this is of little help when dealing with polyphonic instruments such as pianos or guitars, so the polyphonic problem must still be dealt with. This is a problem that does not currently have any perfect solution, unfortunately. However, polyphonic transcription systems are constantly improving, and an IPS does not necessarily always need to know every note that a human is playing. In some cases, just enough information to get a general idea of what is being played is sufficient.

Delays between the playing of a note and its detection by a transcription system, referred to as "latency," present an additional potential problem. Capture, processing and analysis of audio input can be computationally intensive, particularly when combined with potential further delays due to mapping and sound synthesis.

There are a number of basic technical difficulties involved here. Sufficiently long audio frames are needed to derive fine enough spectral peaks from an FFT (Fast Fourier Transform) analysis. Transients during note attacks can be difficult to map to pitch, so it may sometimes be necessary to wait until the sustain portion of a note envelope before pitch can be identified. In most systems, lower pitches also take longer to track than higher pitches because of their increased period. Although processing speed can be improved by using a low data rate, this does not solve all of these fundamental latency problems, and it could also compromise the effectiveness of the transcription because of the degraded sound quality.

Fortunately, there is reason to believe that consistent latencies of up to 20 to 30 ms are acceptable to musicians performing with each other (Maki-Patola and Hamalainen 2004; Lago and Kon 2004). Furthermore, IPS scenarios involving real-time transcription are less sensitive to latency than sensor-based MIDI instruments, as performers are provided with acoustic feedback from their instruments immediately, independently of system delays.

IPS processing delays perceived by performers, in moderation, can be considered to be comparable to delays present in multi-human performances, when there is a certain amount of delay that passes before a response to a performer's action is manifested by another performer. Human performers certainly have reaction times that are not negligible, and musicians such as orchestral performers who play in large concert halls must deal with significant delays due to acoustics as well. The point of all of this is that competent performers are naturally able to deal with reasonable amounts of latency without difficulty. Of course, the key word here is "reasonable," and one must still be careful not to use overly computationally expensive signal processing techniques, as latencies past a certain point will certainly cause problems for both performers and listeners.

Performers are more sensitive to variable delays, or "jitter," than to consistent latencies. There is evidence that variations as small as 6 ms are detectable (Friberg and Sundberg 1995). A solution to this problem is to stabilize delays by dynamically adding a small artificial delay when the natural delay is low. This regularizes latencies, but increases the overall average latency.

Even if one has access to very fast hardware, there are further latency issues that must be considered. For example, the duration of a note cannot be known until the note is over. Delay is also unavoidable in situations where triggers are based on phrases, as one must wait until the end of a phrase before knowing for certain that it is over. Of course, these types of delays are just as much of an issue with standard sensor-based MIDI instruments, and are by no means unique to audio transcription systems.

One potential solution to these problems is to use a beat tracking sub-system that anticipates where the next beat will be based on recent input. Eck (2002) has proposed a particularly effective beat tracking system, and there are many others as well. This would help to improve matters with more sophisticated IPS's that behave more like human performers.

4. Performance parameters to extract

It is important to be clear about the specific types of information that can be usefully extracted from audio signals and used as the inputs for mappings, particularly since standardization is considered to be a priority here. It is desirable, on the one hand, to extract as many parameters as possible in order to provide composers with a wide palette. On the other hand, however, the extraction of too many parameters can aggravate latencies during performance and can compromise the simplicity and ease of use of a system.

It is argued here that the best compromise is to extract only the most obvious and clearly useful parameters, which should be more than sufficient for the majority of performers, while at the same time making it possible to implement plug-in modules to extract further parameters as needed. The implementation of this extensibility must be treated with care, and should include clear and highly rigorous requirements in the API, as loose extensibility could compromise the portability of the system.

The primary output from the transcription performance parameter acquisition back-end system should likely be a basic stream of MIDI Note Ons with associated MIDI pitches and velocities. This in itself can be used to provide a wide variety of useful features. However, it could also be useful to provide several additional data streams for those needing further information.

One such stream could consist of a pulse for every note onset detected, in case this information is needed before pitch can become available and the corresponding Note On can be produced. This could be particularly useful for polyphonic instruments, as polyphonic onset detection is more reliable currently than polyphonic transcription.

An additional stream could be made up of continuous measurements of the overall amplitude of the input signal and, if possible, of each individual note proportionally. A further stream could consist of continuous fundamental frequency values. This is essential if the system is to deal effectively with microtonal playing of any kind. This stream could be ignored by mapping systems that are only concerned with the quantized pitches included in the MIDI Note Ons.

Another possibility would be to derive timbre-based features from the audio signal. The relative amounts of energy in the low, middle and high spectral regions is just one example of the type of simple feature that could be of use with a wide variety of instruments. Harmonic density is another example. The ability to take advantage of spectral information is an important advantage of transcription-based systems over standard sensor-based instruments, which in many cases ignore this very rich source of expressive information.

It may also be appropriate to allow some very basic exceptions to the exclusive reliance on audio data. This is because a system relying only on audio inputs prevents performers from providing input to the system in any way that is not audible to an audience. Humans performing with each other often interact using non-audible visual cues, for example, and it may be desirable in many pieces to implement some comparable means of communication with an interactive system. This could be particularly useful in cases where a cue is missed and manual state switching is needed. Basic foot switches or visual cues detected automatically using a camera and an image processing sub-system are two potentially useful approaches. In keeping with the design concerns discussed in Section 2, of course, performers should not be required to perform unnatural actions, and any additional equipment must be simple, easily available and, ideally, standardized.

5. Pitch tracking technical details

Although a detailed discussion of signal-processing techniques is beyond the scope of this paper, it is important to briefly review some background on pitch trackers, given the emphasis placed on parameters extracted from audio signals.

With regards to monophonic pitch tracking, Roads (1996, 507–20) provides a good overview of the many techniques known to be effective. These include the use of zero-crossings, autocorrelation, adaptive filtering, FFT-based techniques, tracking phase vocoder analysis, Cepstrum analysis and pitch detection based on models of the ear. Although there is far too much research on monophonic pitch detection to cite comprehensively, the recent work of Brossier, Bello and Plumbley (2004a) does appear particularly promising as an approach that offers a combination of a speed, flexibility and accuracy.

As mentioned earlier, polyphonic pitch tracking is a much more difficult and, correspondingly, interesting subject. A great deal of research has relied on the analysis of frequency-domain data and on the use of relatively sophisticated pattern recognition techniques to separate out different notes that could be occurring simultaneously. Determining which partials belong to which notes, or even when note onsets are occurring, can be a difficult task. Telling the difference between harmonics belonging to a single note and a simultaneous note played an octave apart is a particularly troubling problem. Note doubling in different voices can be even more difficult to detect.

Much of the research on polyphonic pitch detection has involved music with multiple different instruments, and techniques such as range filtering and spectrum templates have often been used in order to separate out instruments. For example, Hainsworth and Macleod (2001) extracted bass lines from polyphonic signals by filtering out high frequencies. Unfortunately, these approaches are of little use for transcription of simultaneous notes originating from a single instrument in an arbitrary register, which is the problem that we are concerned with here. However, the use of a dictionary of note templates belonging to just one instrument could still be of use if one wishes to store a template for each pitch that a pattern recognition system can try to match.

The use of blackboard systems has been proposed by both Martin (1996a) and Kashino et al. (1995), and they have been used with some success. Monti and Sandler (2002) have expanded on this approach as well. Martin's system, which was intended for transcribing four-voice piano, made use of STFTs (short-time Fourier Transforms) to generate associated sets of onset times, frequencies and amplitudes that were input to the blackboard system. Martin later suggested modifying the signal processing front-end in order to solve problems with misidentification of octaves (Martin 1996b). He suggested first using a bank of filters to produce log-lag correlograms, and then determining pitch by measuring the periodic energy in each filter channel as a function of lag. The correlograms could then be input as the basic unit to the blackboard system.

Kashino's approach involved the use of a Bayesian probability network for coordinating the blackboard system. Kashino also used knowledge sources with information about stream segregation taken from research in human auditory scene analysis, and gave his knowledge sources more high-level musical knowledge than Martin. Also, unlike Martin's system, Kashino used knowledge sources programmed with the frequency components of different instruments played with different parameters in order to deal with more than just one instrument. He later suggested replacing the Bayesian network with a Markov Random Field hypothesis network (Kashino 1996). One problematic aspect of Kashino's approach, from the perspective of a generalized system, is that the knowledge sources were programmed with style-specific theoretical information.

Bello and Sandler (2000) have implemented a system based on Martin's design that used a sequential scheduler. Aside from refining the knowledge sources and adding high-level musical knowledge, they also implemented a chord recognizer knowledge source as a feed-forward neural network. The network was trained using spectrographs of different chords of a piano and it produced candidate chords. The network could output more than one hypothesis at each iteration, allowing the system to perform a parallel exploration of the solution space. Bello, Monti and Sandler (2000) used a system that received the averaged STFT of a signal and identified the peaks in the spectrum. These peaks were stored as tracks that the system followed over time. Both of these research directions are further described in Bello's PhD dissertation (2003).

Research has also been performed on using models of the human auditory system. The work of Brown and Cooke (1994) and of Godsmark and Brown (1999) is of interest in this respect. Martin (1996b) and Marolt and Privosnik (2001) have both used gammatone filterbanks to decompose audio signals into a number of frequency bands in such a way that the frequency and width of each band closely resembled equivalent bands on the basilar membrane. The output of each band can be processed by a model of the inner hair cells of the cochlea.

Further analysis can then be performed using short-time auto-correlation. One variation includes using a bank of filters to produce log-lag correlograms, and then determining pitch by measuring the periodic energy in each filter channel as a function of lag (Martin 1996b). Martin claims that this approach makes the bottom-up detection of octaves possible. Instead of using autocorrelation or some kind of peak-picking algorithm next, Marolt and Privosnik (2001) employed a network of adaptive oscillators. These oscillators adapted their phase and frequency in response to an input. Partial tracks could then be formed by observing the output of each oscillator.

An approach that is gaining increasing popularity is the use spectral modeling synthesis (SMS) analysis (Serra 1997; Cano 1998; Roebel et al. 2004). This algorithm models an input signal as a number of sinusoids plus a residual noise component. The sinusoids can provide a good input to a pattern recognition system. Some initial progress has been made with this technique (McKay and Hatch 2003), and it is worthy of further investigation.

Dixon (2000) used a tracking phase vocoder in order to alleviate some of the uncertainty in low notes. Rather than using the centre frequency of each bin, a more accurate estimate was attained by using phase information obtained from adjacent FFT windows. The rate of phase change in the bins surrounding a spectral peak was used to find the actual frequency present.

Some of the most impressive recognition rates have been achieved by Goto (2000). This system extracted both bass and melody lines from complex audio signals. However, band-pass filters were used to segregate these two streams of information. This approach would be unable to deal with notes in arbitrary registers, unfortunately. The sophisticated tracking agents used are quite promising for general use, however.

Very good results have also been achieved by Klapuri (2003), who has made use of a promising iterative process to estimate fundamental frequencies. Klapuri's thesis (1998) and dissertation (2004) include further information on this, as well as information on useful onset detection techniques for polyphonic music.

Perhaps some of the most promising research in the context of the particular goals of this project is the work of Adallah and Plumbley (2004), who designed a system for transcribing polyphonic piano music. However, this system did use a complex set of techniques that could introduce long latencies. Raphael (2002) has presented an alternative approach to transcribing piano music using hidden Markov models. One of the few systems that consider speed as a serious priority in polyphonic transcription is that proposed by Lepain (1999).

The blackboard approach used in some of the systems discussed above is of particular interest, as a number of specialized classifiers can be combined. For example, knowledge sources specializing in detection of note onsets, pitches and note endings could all work together to vote on the final transcription output. A sequential approach could be used as well, where decisions such as silence / a note playing, one note present / multiple notes present, new onset / old note sustaining, what pitch(es) are present, and whether a note is ending could be made. The decisions of an early classifier would influence which specialized classifiers are used at a later stage. Of course, the cost of the accuracy improvements introduces by using multiple expert subsystems is increased latency.

Much of the above research also includes discussion of the simpler problem of detecting note onsets. In addition to this, a good general real-time approach is presented by Brossier et al. (2004b).

6. Client/server architecture

An essential issue in IPS's is how the extracted performance parameters are mapped to the music performed by the system. This is left purposefully vague here, as it is essential that composers and designers have creative freedom that excessive standardization could potentially stifle. Composers should therefore be able to implement their own mapping system, and use it to communicate with the parameter extraction system discussed in the preceding sections.

There are a wide variety of sophisticated features that mapping systems could extract from the data streams discussed in Section 4 and use to form mappings. Robert Rowe (2001) has provided a good overview of a number of these techniques, including a wide range of information that can be extracted just from a simple MIDI stream. Chadabe (1997) has written a good reference on IPS's that have been used by a variety of composers and performers in the past.

Unfortunately, any mapping system will inevitably encounter unexpected events, such as performer errors. The system must therefore be robust enough to deal with and recover from such difficulties. Those musical events that cause the most important actions to be performed by the computer should be given particular attention in this respect. Composers should therefore consider mappings in terms of their robustness to performer errors as well as from the obvious aesthetic perspective.

Furthermore, it is essential that individual composers implement their mapping systems to be portable and easy to use. In order to help with this, a client/server architecture is suggested. The standardized parameter extraction system could be implemented as a server, and individual mapping systems could be implemented as clients that use standard network protocols to read control data from the server.

Aside from the advantage of making it easy to port mapping systems to the parameter extractor, this also has the advantage of potentially dividing the processing workload over multiple computers, so that one would handle the parameter extraction tasks, and another would handle mapping and synthesis. This could decrease lag problems.

The client/server interface also provides an intuitive way to implement multi-agent IPS's. Multiple servers could be used to handle multiple performers, and multiple clients could be used to simulate multiple IPS's performing with humans and with each other.

8. Conclusion

This paper has discussed important issues related to promoting the reliability, robustness, dissemination and longevity of interactive pieces. The following priorities and solutions have been proposed:

- Performers should not be expected to perform unreasonable tasks that are foreign to their training.
- Limitations in performer bandwidth must not be exceeded.

- A clear explanation must be made to performers of specifically how the system will respond to their actions.

- Performers should be given sufficient time to learn how to use the system, including significant rehearsal time with the fully operational system.

- Intuitive feedback should be provided to performers, ideally including auditory, visual and, potentially, haptic feedback.

- Software should be well tested and robust to unexpected eventualities such as performer mistakes.

- Systems should require only minimal supervision during performances.

- Basic overrides should be available, although they should only be necessary on very rare occasions.

- Software should have an easy-to-use and well-documented interface.

- Software should be portable between different computer platforms and easy to install.

- Implementations should use well-known languages that are likely to have a long life-span

- Only inexpensive, commonly available and easy to transport equipment should be used whenever possible.

- Any proprietary or unique hardware or software that will be difficult to acquire and set up should be avoided possible unless absolutely necessary.

In particular, it was suggested that the implementation of a standardized system dedicated to extracting parameters from performers should be constructed and distributed. This system could be implemented as a standardized server that could communicate with individualized mapping clients implemented based on the particular needs of individual composers and performers.

It was also suggested that there are a number of benefits to limiting the inputs of the parameter extraction system to only the audio signal produced by performers, with the possible addition of a camera-based gesture recognition system and simple devices like footswitches. This would have strong advantages in aiding the dissemination, reliability and longevity of pieces relative to IPS's that use complex sensors as well.

Unfortunately, current technological limitations relating to polyphonic transcription limit the application of this latter suggestion to monophonic instruments. It is hoped, however that continuing technological advances will soon make this goal realizable.

9. Bibliography

Abdallah, S., and M. Plumbley. 2004. Polyphonic music transcription by non-negative sparse coding of power spectra. *Proceedings of the International Conference on Music Information Retrieval*. 318–25.

Bello, J. 2003. Towards the automated analysis of simple polyphonic music. *PhD Dissertation.* Queen Mary, University of London, Centre for Digital Music.

Bello, J., G. Monti, and M. Sandler. 2000. Techniques for automatic music transcription. *Proceedings of the International Symposium on Music Information Retrieval.*

Bello, J., and M. Sandler. 2000. Blackboard system and top-down processing for the transcription of simple polyphonic music. *Proceedings of the COST G-6 Conference on Digital Audio Effects.*

Brossier, P., J. Bello, and M. Plumbley. 2004a. Fast labeling of notes in music signals. *Proceedings of the International Conference on Music Information Retrieval.* 331–6.

Brossier, P., J. Bello, and M. Plumbley. 2004b. Real-time temporal segmentation of note objects in music signals. *Proceedings of the International Computer Music Conference.* 458–61.

Brown, G., and M. Cooke. 1994. Perceptual grouping of musical sounds: A computational model. *Journal of New Music Research* 23: 107–32.

Cano, P. 1998. Fundamental frequency estimation in the SMS analysis. *Proceedings of the COST G6 Conference on Digital Audio Effects.* 99–102.

Chadabe, J. 1997. *Electric sound: The past, present and future of electronic music.* Upper Saddle River, NJ: Prentice Hall.

Dixon, S. 2000. Extraction of Musical Performance Parameters from Audio Data. *Proceed-*

ings of the First IEEE Pacific-Rim Conference on Multimedia.

Eck, D. 2002. Finding downbeats with a relaxation oscillator. *Psychological Research* 66(1): 18–25.

Friberg, A., and J. Sunberg, 1995. Time discrimination in a monotonic, isochronous sequence. *Journal of the Acoustical Society of America* 98(5): 2254–531.

Godsmark, D. and G. Brown. 1999. A blackboard architecture for computational auditory scene analysis. *Speech Communication* 27: 351–66.

Goto, M. 2000. A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.*

Hainsworth, S., and M. Macleod. 2001. Automatic bass line transcription from polyphonic music. *Proceedings of the International Computer Music Conference.*

Kashino, K., and N. Hagita. 1996. A music scene analysis system with the MRF-based information integration scheme. *Proceedings of the International Conference on Pattern Recognition*. 725–29.

Kashino, K., K. Nakadia, T Kinoshita, and H. Tanaka. 1995. Application of Bayesian probability network to music scene analysis. Proceedings of the International Joint Conference on AI, CASA workshop. 52–9.

Klapuri, A. 2004. Signal processing methods for the automatic transcription of music. *Doctoral Dissertation.* Tampere, Finland: Tampere University of Technology.

Klapuri, A. 2003. Multiple fundamental frequency estimation by harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing* 11(6): 804–16.

Klapuri, A. 1998. Automatic transcription of music. *Master's Thesis*. Tampere, Finland: Tampere University of Technology.

Lago, N., and F. Kon. 2004. The quest for low latency. *Proceedings of the International Computer Music Conference*. 33–6.

Lepain, P. 1999. Polyphonic pitch extraction from music signals. *Journal of New Music Research* 28(4): 296–309.

Li, W. C. 2004. A performer's musicological research in performing interactive computer music. *Proceedings of the International Computer Music Conference*. 98–104.

Maki-Patola, T., and P. Hamalainen. 2004. Latency tolerance for gesture controlled continuous sound instrument without tactile feedback. *Proceedings of the International Computer Music Conference*. 409–16.

Marolt, M, and M. Privosnik. 2001. SONIC: A system for transcription of piano music. *Proceedings of the Workshop on Current Research Directions in Computer Music.*

Martin, K. 1996a. A blackboard system for automatic transcription of simple polyphonic music. *M.I.T. Media Lab Perceptual Computing Technical Report* #385, July 1996.

Martin, K. 1996b. Automatic transcription of simple polyphonic music: Robust front end processing. M*.I.T. Media Lab Perceptual Computing Technical Report* #399, November 1996.

McKay, C., and W. Hatch. 2003. Transcriber: A system for automatically transcribing musical duets. *Technical Report*. McGill University, Canada.

Monti, G., and M. Sandler. 2002. Automatic polyphonic piano note extraction using fuzzy logic in a blackboard system. *Proceedings of the International Conference on Digital Audio Effects.*

Pennycook, B. 1997. Live electroacoustic music: Old problems, new solutions. *Journal of New Music Research* 26(1): 70–95.

Raphael, C. 2002. Automatic transcription of piano music. *Proceedings of the International Symposium on Music Information Retrieval*. 15–9.

Roads, C. 1996. *The computer music tutorial*. Cambridge, MA: MIT Press.

Roebel, A., M. Zivanovic, and X. Rodet. 2004. Signal decomposition by means of classification of spectral peaks. *Proceedings of the International Computer Music Conference*. 446–9.

Rowe, R. 2001. *Machine musicianship*. Cambridge, MA: MIT Press.

Serra, X. 1997. Musical sound modeling with sinusoids plus noise. In *Musical signal processing*. C. Roads, S. Pope, A. Picialli, and G. De Poli eds. Lisse, The Netherlands: Swets & Zeitlinger.