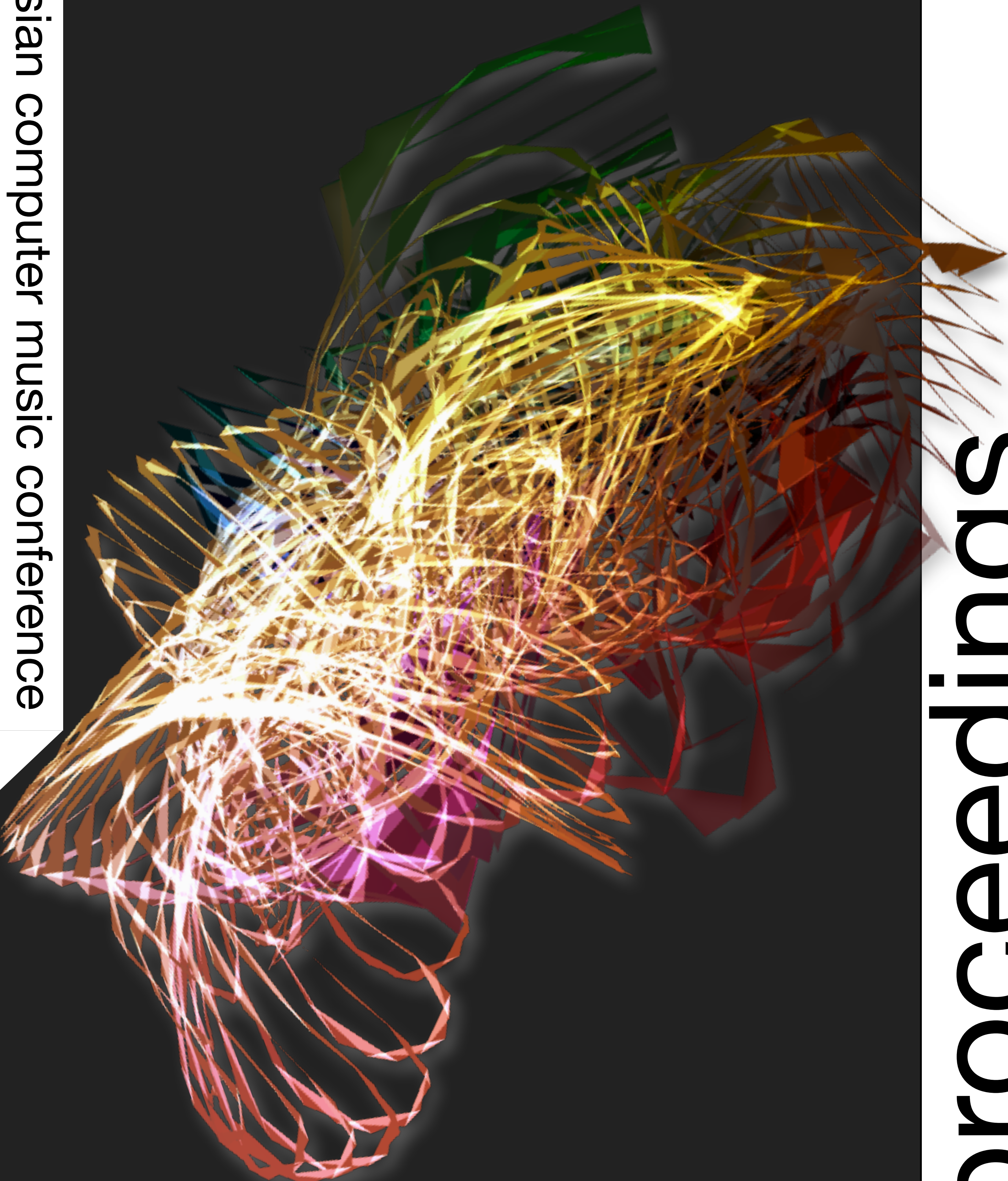
improvise

australasian computer music conference



proceedings

Brisbane July 2-4 2009

IMPROVISE

Proceedings of the Australasian Computer Music Conference 2009

Queensland University of Technology
Brisbane, Australia

2-4 July 2009

Published by the Australasian
Computer Music Association

P.O. Box 8024
Monash University LPO
VIC 3800
Australia

July 2009

ISSN 1448-7780

The paper refereeing process is conducted according to the specifications of the Australian Government for the collection of higher Education research data, and fully refereed papers therefore meet Australian Government requirements for fully-refereed research papers.

Contents

Refereed Papers

Artist Talks/Posters

DISTRIBUTED PERFORMANCE IN LIVE CODING

Ben Swift, Henry Gardner
School of Computer Science
ANU College of Engineering and Computer Science
{ben.swift, henry.gardner}@anu.edu.au

Alistair Riddell
Photography and Media Arts
ANU College of Arts and Social Sciences
alistair.riddell@anu.edu.au

ABSTRACT

Distributed composition is an extension to current live coding practice, in which audience members are given control of certain aspects of the performance. The scope of this control is dictated by the live coder as the performance progresses. In this way, the role of performer shifts from the live coder alone, to be distributed amongst participating audience members. The live coder acts as a compositional overseer. This offers new possibilities for creativity and engagement in a live coding performance.

1. INTRODUCTION

Live coding is primarily an improvisational performance practice. During a performance a live coder must create musical structures (such as generative processes) and manipulate their associated parameters over the course of the performance [16]. The common live coding practice of code projection (projecting a view of the performer's code onto the screen during the performance) means that audience members can see the processes underlying the performance as well as hear the music which is produced [10]. While not every audience member at a given live coding gig has the necessary background to understand the code on screen, those who do are free to examine and appraise the performer's decisions on algorithmic choice and parameter selection.

The notion of performance in live coding incorporates elements of both composition and improvisation. This paper describes a performance paradigm in which audience members are, at the discretion of the performer, given direct control over some of the parameters in the performer's code. Audience members can then use a control device to change these parameters over time, while the live coder continues to work on other parts of the code. The role of the performer is therefore distributed amongst the live coder and the audience participants. We believe that this concept of *distributed performance* is a promising extension to current live coding practice, and suggests a new performance modality for exploration as the practice of live coding matures.

At present, we have a working distributed performance system implemented using *Impromptu* [17] as a live coding environment and the MRMR OSC Controller [12] application on the Apple iPhone (or iPod Touch) as a control device. Section 2 of this paper discusses some of the opportunities and conditions associated with distributed performance in live coding, and Section 3 describes the details of our implementation as well as the results of our first concert.

2. WHY RELINQUISH CREATIVE CONTROL?

2.1. Interaction Model

The experimental computer music group the *Hub* were pioneers in the use of technology to facilitate new interactions between performers. Tim Perkins, one of the founding members of the *Hub*, reflects in their CD liner notes

> [our] emphasis has been on connections between musicians, the excitement of using computers to define a new social context for music making, as well as exploring the possibilities of systems too complex for direct control.[8]

Similarly, incorporating audience interaction into the practice of live coding offers new possibilities for distributed control of complex musical systems. While live coders sometimes perform in groups, ranging in size from pairs [16, 11] up to whole orchestras [15, 13], these are static, egalitarian set-ups, with a pre-set number of performers having a more or less equal role in the performance. In contrast, distributed performance in live coding allows audience members to control specific aspects of the musical performance, with the live coder changing the roles and relationships of the audience participants on-the-fly. The live coder acts as an overseer, dynamically distributing creative control in response to the changing demands of the performance.

Distributed performance is not a musical democracy or a chaotic free-for-all where each participant acts independently of all others. The live coder is still in control of the overall shape of the piece, but he chooses particular parameters to 'distribute'; those which he knows to be suitable for audience control (see Figure 1). The performer also chooses the type and range of the control given to the participant, depending on the control affordances of the interaction device in use. The live coder determines the role of the audience as the code is written. This improvisational dimension fits well with the idea of live coding being an improvisational practice, the live coder creating and responding to musical ideas like a modern concerto artist [3].

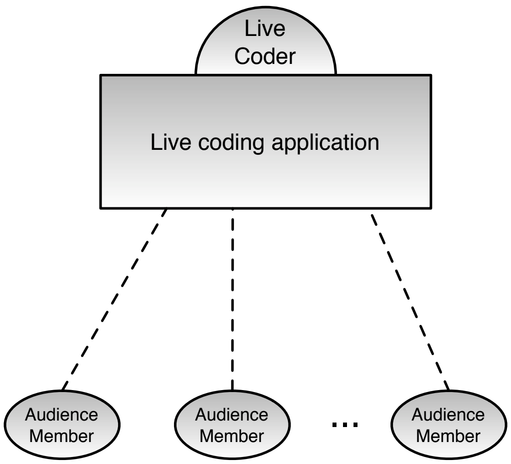


Figure 1. A distributed performance set-up

It is important to point out that the idea of outsourcing some control to an audience member is not designed to address a perceived shortcoming in current live coding practice. Clearly, a work of art is not necessarily poorer for the fact that it is the vision of a singular skilled artist rather than a collaborative effort involving the viewing public. However, sharing some control with the audience in a live coding performance gives rise to some interesting possibilities

1. increased audience engagement
2. the introduction of a social dynamic between different audience members, each in charge of a different part of the code
3. the introduction of musical material beyond the control of the live coder, to which he/she must react
4. a new modality of live coding performance

2.2. Engaging the Audience

Mechanisms for audience participation in a performance can lead to an enhanced performance experience [2, 18, 14]. During a performance, live coders often set up generative processes which create ongoing streams of musical material [4]. These processes may be governed by a small number of parameters which control the musical material they generate. Allowing an audience member to control the trajectory of these parameters over the course of a performance gives them a share in the creative process.

Furthermore, the projection of the code for the audience is designed to facilitate audience engagement [3]. The code is an abstraction, a symbolic representation of the performance. In live coding this symbolic representation is made visible to the audience, in contrast to other forms of improvisational performance where the decisions and creative processes of the performer are inscrutable. The live coder can take advantage of this 'transparency of creative process' by inviting the audience to modify this symbolic representation (that is, the code), resulting in a deeper engagement with the performance.

2.3. Introducing a Social Dynamic

With more than one audience performer, the relationship between the various participants can be explored [9]. As the live coder delegates control of certain parameters out to different audience participants, each audience member can observe the gestures and control manipulations of their peers in the audience. Each participant is responsible for a certain aspect of the whole performance, and the interplay between the participants is a key factor governing the overall aesthetic of the piece. Furthermore, any audience participants who have not chosen to participate directly (that is, who do not have a control device) can observe the behaviour of the performers distributed throughout the audience. The divide between performer and audience present in many conventional music performance practices is removed, and this adds further incentive for all audience members to engage with the performance.

There is also scope for more complex interaction between participants in the audience. The mapping of a participant's control to a parameter need not be one-to-one, control may be given to two (or more) audience participants to control collaboratively. The co-location of the audience members can be utilised, such as by giving two participants a pop-up instruction to find each other and synchronise control gestures. The activity of individual participants and the relationships between them can be displayed visually, alongside the projected code. These relationships, such as groupings amongst participants, can be controlled dynamically by the live coder, or even to a participant to which the ability has been delegated.

2.4. Responsive Coding

In any improvisational practice, creative decisions are made continuously as the performance unfolds, often in response to material unanticipated by the performer [7]. In jazz, for instance, this 'new material' can be introduced by other members of the ensemble, while in live coding, novel material may be the result of stochastic processes. Indeed, responding to the music as it unfolds is one of the challenges of live coding, and can also be a catalyst for inspiration.

Incorporating audience control is an opportunity for the live coder to respond to the choices of the other participants in the performance. This adds a responsive element akin to jamming in a free jazz ensemble, as the music produced by the participants may provide inspiration for the live coder to create and modify the code in new ways. The asymmetry in the degree of control possessed by the live coder (a great deal of control) and the participant (limited control, as dictated by the live coder) means that the situation is not quite the same as an instrumental jam, where each musician is limited only by their instrument and ability. However, if participants could enter a 'code entry' mode and send actual snippets of code to the live coder's code buffer, then this interaction between coder and audience participants begins to resemble that of a more traditional improvisational ensemble.

2.5. Conditions of Distributed Performance

Certain conditions must be considered when extending a live coding environment to accommodate distributed performance. Coding a performance in real-time is already a cognitively demanding activity, and introducing audience control into the performance can add to the cognitive load of the live coder. For this reason, it is important to customise the live coding software environment to make the distribution of control as simple and effective as possible. A selection of different interaction modes, based on the affordances of the interaction device and the requirements of the performance, can be designed ahead of time. These pre-set modes can then be pushed to a (possibly random) participant with a simple function call in the code. The distributed performance infrastructure should be pre-prepared so that the live coder can spend their time writing musical (rather than boilerplate) code. Some ideas regarding this 'interaction infrastructure' in the live coding environment are discussed in Section 3.1.3.

Feedback is crucial in distributed performance to ensure that the participants are aware of exactly what they are controlling, and the scope of that control [5]. If the participants cannot determine which aspect of the music they are influencing, they will quickly lose interest in the interaction. Feedback can either be auditory (incorporated into the musical performance) or visual (alongside/overlaid on the projected code).

For visual feedback to be useful, each participant must register a name (or other unique identifier) which can be used to tag any visual feedback applicable to them. This can be done in a simple configuration step upon joining the performance with their interaction device. The system can then give meaningful visual feedback, such as a graphical overlay of the participant's name attached to the particular parameter they are controlling in the projected code.

In live coding, there are certain parameters in the code which have immediate and noticeable effects (this is described by John Croft as *aesthetic liveness* [6]), such as timbral parameters (filter cutoff and resonance, etc.). If these parameters are outsourced, they require minimal visual feedback, as the effect of any participant manipulation is sonically obvious. There are also some parameters whose effects are subtler (such as reverb) and less immediate (changing the parameters of a generative process). Outsourcing these parameters requires more informative visual feedback.

3. DISTRIBUTED PERFORMANCE: OUR APPROACH

3.1. Technical Specifications

We are in the process of implementing a distributed performance system for use in our own live coding practice. Our distributed performance set-up revolves around a single live coder performing on a laptop (using *Impromptu*), and a flexible number of audience participants interacting wirelessly with the live coder using an Apple iPhone (or iPod Touch [1]) running the MRMR OSC Controller application. The basic architecture of the system is shown in Figure 2.

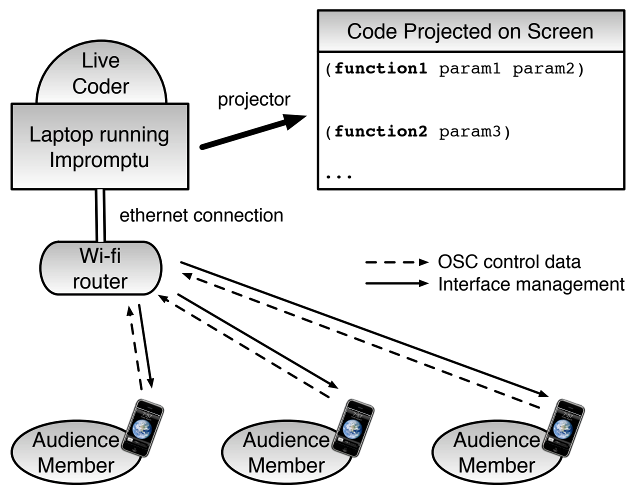


Figure 2. Our live coding set-up

3.1.1. The iPhone as an interaction device

Each iPhone connects to the network over 802.11g wi-fi. The live coding laptop is connected to the same local network using an ethernet connection. As a control device, the main affordances of the iPhone are its 3.5" multi-touch screen and built-in 3-axis accelerometer. The accelerometer, combined with the device's small size, means that the iPhone can be used as a gestural controller.

The iPhone was chosen for two main reasons:

1. While far from ubiquitous, the iPhone is a popular phone, which in our experience has reasonably high ownership rates amongst the type of crowd which would attend a live coding gig. By using a 'BYO controller'

set-up, we eliminate the need for custom interaction hardware, allowing us to have several participants without the expense of purchasing and maintaining a fleet of our own interaction devices.

2. The iPhone provides a standardised development environment for our interaction software. We are currently using an open source software controller called the MRMR OSC Controller [12] to send control data to the live coding environment. In the future, we would like to have interaction clients available for other mobile OSes (such as Symbian, WebOS or Android), but currently our efforts are focused on the artistic practice of distributed performance in live coding, and the iPhone and MRMR has proved valuable in getting the system up and running in a short period of time.

The iPhone and iPod Touch possess an internal speaker, and the iPhone (but not the Touch) also has a built-in microphone. These audio interfaces also offer interesting possibilities for interaction and feedback, although they have not been incorporated into our system at this time.

3.1.2. The MRMR OSC Controller

The MRMR OSC Controller is available as a free download from the online App Store. Upon start-up, it automatically detects and connects to the live coder's laptop using Bonjour Zeroconfig networking. The MRMR software allows the live coder to present a unique interface (called a *patch*) to each iPhone participating in the performance. Each patch is a combination of widgets: buttons, sliders, multi-touch zones, text input fields and accelerometer data (see Figure 3). The participant can use these widgets to send control data back to the live coder using the Open Sound Control (OSC) communication protocol.

One key feature of the MRMR application is that a patch can be modified (or changed completely) at any time by the live coder. Each widget can be represented as a formatted string, specifying they widget's type, size, position and label. A patch is simply a collection of such strings, which is interpreted by the MRMR App to present the appropriate UI to the participant. Strings representing new widgets can be pushed to any device at any time, and the UI will be updated instantaneously.

This allows the control options presented to each audience participant to be changed during the performance. For instance, if a particular piece has three distinct movements, then the interfaces can be updated to reflect the different needs of each movement. Also, different participants may be given different interfaces, allowing the potential for users to be given different roles in the performance. In this regard, the MRMR OSC controller is well suited to the ever-changing needs of the live coder in distributed performance.

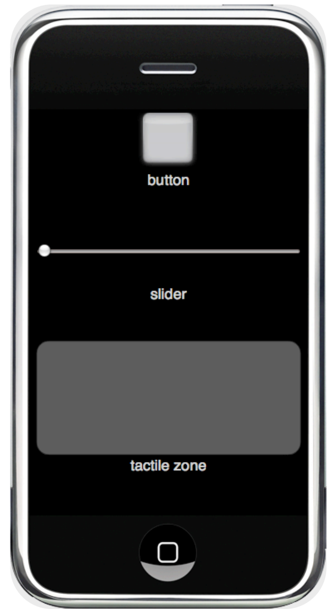


Figure 3. An example MRMR patch

3.1.3. Coding with participation in mind: distributed performance in Impromptu

As we continue to develop our system, we also hope to learn how to best customise a live coding environment to allow for distributed performance. Currently, we are using a client-server interaction model, in which the connections between the iPhones and the live coding laptop are managed automatically. Impromptu, which is a Scheme-based live coding environment, manages a list of the names and IP addresses of all connected devices, and this list is available to the live coder at any time.

As the code is written during a performance, the live coder can (with a function call) push a MRMR patch or read control data from a given device by device name *or* IP. Given the immediacy required in live coding, it is desirable to allow control data from a given iPhone to be accessed as succinctly as possible, and we are currently investigating several possible Scheme forms for doing this. In particular, we are currently using the iPhone control data to vary certain parameters in the live coded Scheme functions.

3.2. *inMates*: Lessons From Our First Concert

The *inMates* concert in April 2009 was the inaugural performance of our distributed performance system. The system is still under active development, and the *inMates* performance served as a means to evaluate the technical and artistic progress of the system, as well as suggest further refinements to our distributed performance approach.

The *inMates* performance took the form of a virtual drum circle. Each distributed performer was assigned control of a certain type of drum, either a djembe, taiko or conga. The participants did not have the ability to hit the drum directly, but could control the beating of their drum along five dimensions. Each participant determined the period, (and phase offset) of their drum pattern, quantised to 32nd-note increments. The participants could also control the volume of their drum, and the 'type' of drum hit (for example, a heel or a slap hit on the conga). Each of these parameters could be controlled by a slider on the iPhone. They could also choose to stop playing, dropping in and out of the circle as they desired, by toggling a button on the iPhone's screen. All up, each performer had five parameters under their control.

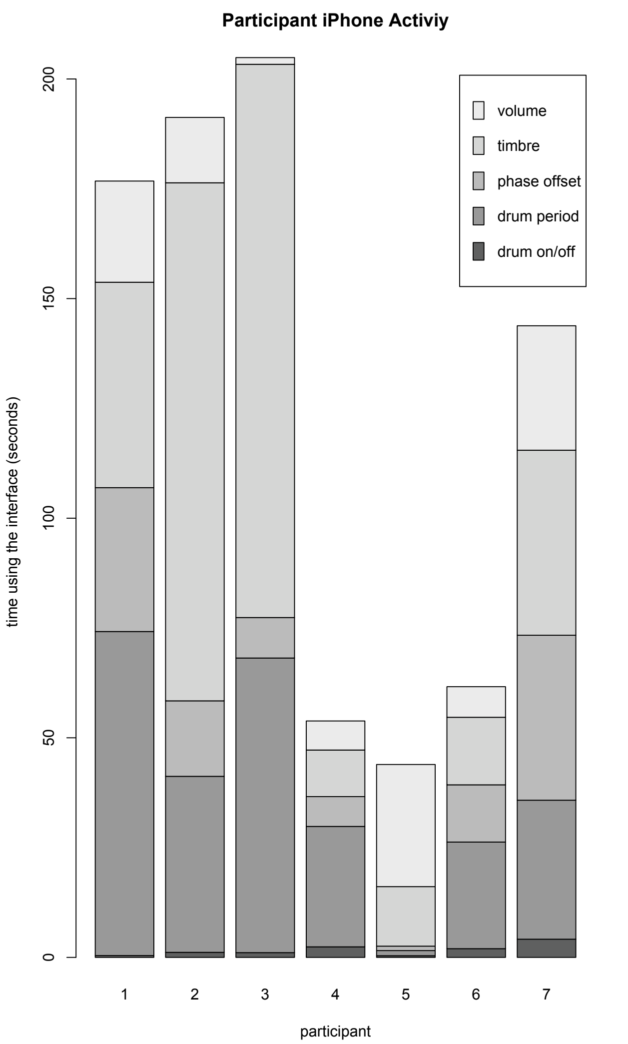


Figure 4. Audience interaction data from the *inMates* concert

In total, there were seven audience participants at the *inMates* concert. The performance lasted 25 minutes and all interaction data was logged. The interaction breakdown, showing the amount of time each participant spent manipulating each different parameter, is shown in figure 4.

Given the preliminary nature of this concert, we do not present any particular statistical conclusions from the interaction data gathered. However, it is interesting to note the different interaction styles of the different participants. Quantifying these differences will be a goal of future concerts. The primary aim of this concert was to test the system under load. In this regard, the concert was a success, with no audio dropouts or packet loss, despite the simultaneous interaction of the seven participants.

Following the concert, an informal feedback session was held amongst the participants. Overall, the participants enjoyed the the ability to influence the performance directly through the iPhones. One participant commented that *"There were some moments where I was really enjoying the drumming and hoping it would continue, but then somebody changed something and that moment was gone."* Participants commented that there were moments where they were particularly enjoying controlling the drum to which they had been assigned.

At times, participants found it hard to ascertain which drum they were controlling. Unfortunately, a planned visual feedback element was not completed in time for the performance, and some participants felt this could have helped them determine which aspects of the overall sound they were controlling. Overall, the participants felt that they would like to participate in future performances.

In future concerts we hope to refine the experience for both live coder and audience participants, including investigating the differences in audience interaction for participants with different levels of musical expertise. The possibility of audience members attending numerous concerts and developing their own particular interaction techniques is also an exciting one.

4. CONCLUSION

Our work on distributed performance in live coding has proven to be a fruitful source of inspiration for our creative practice. The ability to distribute control gives live coders a new source of intentional, sensitive creativity to harness, and allows the audience to engage more deeply with the creative process of the performance.

Live coding, as a discipline, cannot yet boast the same rich history of innovation and extended techniques that many conventional instrumental practices can. However, distributed performance is one opportunity to extend the current landscape of live coding performance, where the live coder and the audience are deeply interconnected in the act of performance. We hope to increasingly incorporate elements of audience involvement in our live coding.

5. REFERENCES

[1] "Apple iphone." [Online]. Available: http://www.apple.com/iphone/

[2] L. Barkhuus and T. Jørgensen, "Engaging the crowd: studies of audience-performer interaction," *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, Apr 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1358628.1358785

[3] A. Blackwell and N. Collins, "The programming language as a musical instrument," *Proceedings of PPIG05 (Psychology of Programming Interest . . .*, Jan 2005. [Online]. Available: http://www.cogs.susx.ac.uk/users/nc81/research/proglangasmusicinstr.pdf

[4] A. Brown and A. Sorensen, "Interacting with generative music through live coding," *Contemporary Music Review*, Jan 2009. [Online]. Available: http://www.informaworld.com/index/909056573.pdf

[5] X. Cao, M. Massimi, and R. Balakrishnan, "Flashlight jigsaw: an exploratory study of an ad-hoc multi-player game on public displays," *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, Nov 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1460563.1460577

[6] J. Croft, "Theses on liveness," *Organised sound*, Jan 2007. [Online]. Available: http://journals.cambridge.org/abstract_S1355771807001604

[7] C. Gould and K. Keaton, "The essential role of improvisation in musical performance," *The Journal of Aesthetics and Art Criticism*, vol. 58, no. 2, pp. 143–148, Apr 2000, improvisation in the Arts. [Online]. Available: http://www.jstor.org/stable/432093

[8] S. Gresham-Lancaster, "The aesthetics and history of the hub: The effects of changing technology on network computer music," *Leonardo Music Journal*, vol. 8, pp. 39–44, Jan 1998. [Online]. Available: http://www.jstor.org/stable/1513398

[9] C. Heath, P. Luff, D. Lehn, J. Hindmarsh, and J. Cleverly, "Crafting participation: designing ecologies, configuring experience," *Visual Communication*, Jan 2002. [Online]. Available: http://vcj.sagepub.com/cgi/content/abstract/1/1/9

[10] A. Mclean, N. Collins, and J. Rohrhuber, "Live coding in laptop performance," *Organised sound*, vol. 8, no. 3, p. 321, Jan 2003. [Online]. Available: http://dialnet.unirioja.es/servlet/articulo?codigo=862522

[11] C. Nilson, "Live coding practice," *Proceedings of the 7th international conference on New interfaces for musical expression*, pp. 112–117, 2007.

[12] E. Redlinger, "Mrmr osc controller." [Online]. Available: http://mrmr.noisepages.com

[13] J. Rohrhuber, A. de Campo, R. Wieser, and J. van Kampen, "Purloined letters and distributed persons," *Music in the Global Village Conference (Budapest)*, Jan 2007. [Online]. Available: http://www.wertlos.org/articles/Purloined_letters.pdf

[14] J. Sheridan, N. Bryan-Kinns, and A. Bayliss, "Encouraging witting participation and performance in digital live art," *BCS-HCI '07: Proceedings of the 21st British CHI Group Annual Conference on HCI 2007: People and Computers XXI: HCI...but not as we know it*, vol. 1, Sep 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1531294.1531297

[15] S. Smallwood, D. Trueman, P. Cook, and G. Wang, "Composing for laptop orchestra," *Computer Music Journal*, vol. 32, no. 1, pp. 9–25, Mar 2008. [Online]. Available: http://muse.jhu.edu/journals/computer_music_journal/v032/32.1smallwood.pdf

[16] A. Sorensen and A. Brown, "aa-cell in practice: An approach to musical live coding'," *Proceedings of the International Computer Music Conference*, Jan 2007. [Online]. Available: http://impromptu.moso.com.au/extras/aa-cell-icmc07.pdf

[17] A. Sorensen, "Impromptu." [Online]. Available: http://impromptu.moso.com.au

[18] R. Taylor, P. Boulanger, P. Olivier, and J. Wallace, "Exploring participatory performance to inform the design of collaborative public interfaces," *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, Apr 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1520340.1520561

ENABLING MUSICAL APPLICATIONS ON A LINUX PHONE

Greg Schiemer
Sonic Arts Research Network
Faculty of Creative Arts
University of Wollongong
2522

Eva Cheng
School of Electrical and Computer Engineering
RMIT Melbourne
3000

ABSTRACT

Over the past decade the mobile phone has evolved to become a hardware platform for musical interaction and is increasingly being taken seriously by composers and instrument designers alike. Its gradual evolution has seen improvements in hardware architecture that require alternative methods of programming. Dedicated I/O instruction sets for dealing with the idiosyncracies of various embedded peripheral devices are gradually being overtaken by I/O control using generic software that behaves more like operating systems developed for mainframe computers over three decades ago. This paper looks at the Neo FreeRunner, an open source mobile phone programmed using Linux. Its attraction as a platform for musical instrument development is that many musical applications created using open source cross platform software that once ran only on desktop computers can be now run in an embedded environment. The paper documents procedures we used in order to run musical applications effectively in the Neo FreeRunner. Musical motivations for using this platform can also be found in musical instrument development with j2me phones that provided a foundation for the creative work of the first author over the past 4 years.

1. INTRODUCTION

The Neo FreeRunner is a Linux phone that includes a range of hardware features such as wireless network connectivity, embedded peripherals and increased processing power offered by an advanced RISC processor. It offers developers an open source embedded platform with the scope to create a new genre of electronic music involving interaction with live performers.

Its operating system is not the only aspect of the Neo FreeRunner that is open source. Its circuit schematics and component layout can be downloaded making it possible for developers to customise hardware design. CAD files are even available for the casing of the Neo FreeRunner under a ShareAlike Creative Commons license.

Our approach has been to develop the musical application for the Neo FreeRunner, emulate this in a desktop environment and run it on the mobile platform without modifying the code emulated on the desktop.

The prospect of using compiled Arm9 native code offers a way to synthesise music using generic music software such as Pure data and Csound rather than interpretive languages like java and python which have been used in mobile devices [1, 2]. A similar approach to mobile synthesis has been adopted using the Symbian operating system [3].

The Linux environment is more suited to the development of new applications in embedded hardware than the j2me environment which had previously been used by the first author for developing musical applications [4, 5, 6].

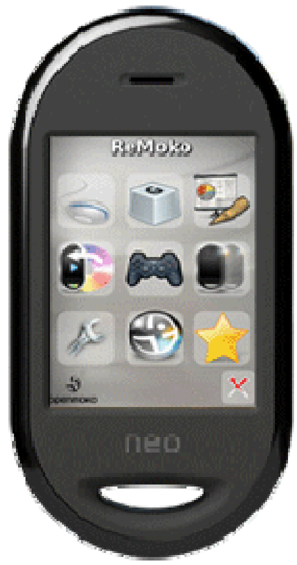


Figure 1. The Neo FreeRunner Linux Phone.

2. BACKGROUND

In many respects the Linux tool set resembles the firmware library developed for microcontroller hardware such as the MIDI Tool Box - hereafter called MTB [7]. However, the capabilities of the MTB and Neo FreeRunner differ in three significant ways.

Firstly, the Neo FreeRunner has a 400 Mhz 32-bit RISC processor, an engine theoretically capable of realtime synthesis; the MTB is based on an 8MHz microcontroller and is more suited to DIY musical applications using physical computing devices.

Secondly, the Neo FreeRunner RAM address space is large enough to run and store musical applications on a handheld device; moreover, it is possible to download

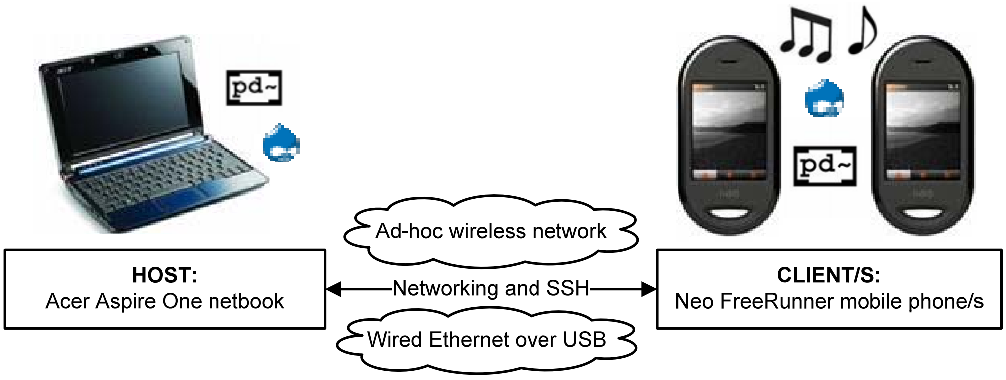


Figure 2. The Acer netbook host and FreeRunner client system paradigm

and run musical applications developed on a variety of standard PC platforms making it unnecessary to create an embedded library of firmware objects that allowed MTB users to program MIDI applications.

Thirdly, the low voltage power requirements of the Neo FreeRunner are suitable for a handheld or wearable device, whereas the power requirements of the microcontroller[1] at the core of the MTB were designed with automotive applications in mind.

3. DEVELOPMENT ENVIRONMENT

While development in future might potentially involve a host machine running a non-Linux operating system such as Windows or MacOS, the development environment we adopted was an Acer netbook used as a host terminal. This was used to communicate with the phone client via wifi or alternatively via USB as shown in Figure 2. A remote terminal was necessary to overcome the physical constraints that the limited screen size and stylus input places on most mobile phone users. Initially, we used the netbook's default Linpus Lite system, a derivative of the Fedora Linux distribution, but found it necessary to adopt the user-friendly Debian-based Ubuntu (version 8.10) instead. Ubuntu offered a richer set of libraries from the software repositories for application development whilst also providing optimisations for the netbook hardware.

Although the Ubuntu software repositories greatly aided the installation of music composition software, there were also several known bugs with Ubuntu on the Acer notebook that are currently being addressed by the Debian/Ubuntu Linux community, including: failing to suspend when the lid is closed (instead the screen just turns off, easily draining the battery); not recovering from hibernating, requiring a hardware reset; sound sometimes not working after suspending, requiring a logout or restart; putting the laptop into suspend which can sometimes corrupt the SD card partition table on waking up.

We also decided to replace the Neo FreeRunner's default OpenMoko operating system (version OM2007.2) with Debian in order to minimise problems that may arise with different Linux distributions in both the host and remote processors; further, similar to Debian-based Ubuntu on the Acer netbook, the Debian software repositories provided comprehensive package management for installing music composition software. To maintain the original OpenMoko operating system, the phone was dual-booted by installing Debian on the phone's microSD card and simply changing the booting order.

4. HOST

The Acer notebook is used as the host terminal. It establishes communication with the remote client via ethernet over USB or wifi. The Acer was setup to be used by non-expert computer users, with scripts automating procedures required to setup communications with the phone.

4.1. Logging In

For ease of use and quick startup time, Ubuntu on the Acer automatically logs in the user. However, users must enter their password when executing sudo commands or when bringing the laptop out of suspend mode.

4.2. Networking

All networking scripts print out the current networking configuration on completion so the user can see at a glance whether or not the script was successful. When enabling a networking interface (wifi or USB), that interface should have an Internet Protocol (IP) address, and conversely, when disabling an interface, the interface should be listed but without an IP address.

4.3. Ad-Hoc Wifi to the FreeRunner

Ad-hoc wifi (i.e., peer-to-peer) was chosen as the wireless connection paradigm as it enables multiple mobile devices (computers and/or phones) to join a secure wifi network without requiring any networking infrastructure such as access points. Note that running the ad-hoc wifi scripts provided disables the Ubuntu Network Manager which controls the Ethernet connection on the Acer, thus the wired Ethernet will not work! It is necessary to dis-

[1] Motorola MC68HC811E2

able the Network Manager to stop it from controlling the wireless interface.

4.3.1. WiFi On

To turn on ad-hoc WiFi from the Acer to the FreeRunner, a shell script was written to automate the process, to be run by the user by typing into the command line (in any directory):

```
./home/greg/Documents/adhocWifi_ON.sh
```

4.3.2. WiFi Off

To turn off ad-hoc WiFi to the FreeRunner, the user simply runs the provided script by typing into the command line (in any directory):

```
./home/greg/Documents/adhocWifi_OFF.sh
```

4.3.3. WiFi Properties

Ad-hoc WiFi connection properties are:

ESSID: `p2p`

Channel: `1`

WEP encryption key: 128-bit Hex key

IP address of Acer: `192.168.2.200`

IP address of FreeRunner: `192.168.2.202`

4.4. USB Networking to the FreeRunner

Internet access from the Acer to the FreeRunner can be enabled over the USB Ethernet connection. In order to do this there are also some settings that may need to be tweaked, including the Domain Name Servers (DNS) of the Internet Service Provider (ISP).

To change the DNS servers, edit the IP addresses in the following lines of the file `/etc/network/interfaces` on the FreeRunner (not the Acer!):

```
up echo nameserver 192.189.54.33 >
/etc/resolv.conf

up echo nameserver 203.8.183.1 >>
/etc/resolv.conf
```

It will be necessary for a user to know the IP addresses of the DNS servers on their local area network; DNS server IP addresses can be found by asking the ISP, network administrator, or ISP helpdesk. One or both IP addresses should then be edited into the `/etc/network/interfaces` file.

4.4.1. USB On

To turn on USB networking to the FreeRunner, to run the provided script the user simply types into the command line (in any directory):

```
./home/greg/Documents/usbNetworking_ON.sh
```

4.4.2. USB Off

To turn off USB networking to the FreeRunner, the user simply types into the command line (in any directory):

```
./home/greg/Documents/usbNetworking_OFF.sh
```

Note that exiting from an SSH session (see Section 4.5) with the FreeRunner will usually also automatically turn off USB networking on the Acer.

4.4.3. USB Properties

To distinguish the USB and wifi networks, different subnets are allocated in the IP addresses. USB networking connection properties are:

IP address of Acer: `192.168.0.200`

IP address of FreeRunner: `192.168.0.202`

4.5. Communicating with the FreeRunner over SSH

To use the terminal of the FreeRunner (as you would on the phone) over SSH, type into the command line (in any directory):

Over USB networking: `ssh root@192.168.0.202`

Over WiFi networking: `ssh root@192.168.2.202`

4.6. Using CSound 5.08

At the moment of writing there are occasional problems with CSound5 GUI which has crashed unexpectedly and without warning. This is a known bug with csound5gui on Ubuntu.

However, it is possible to run CSound 5.08 using a command line interface by typing into the command line (in any directory):

```
Csound test.csd
```

or

```
Csound -aiff -l test.orc test.sco
```

4.7. Using Pure Data 0.41-0

To run PD, click on the 'Sound and Video' item on the left hand menu on the Desktop, and click on the Pure Data icon. Or, type into the command line:

```
Pd
```

5. CLIENT

5.1. Installing Debian on the Neo FreeRunner

Debian was installed on the microSD card to preserve the factory install of OM2007.2. The only requirements for installing Debian on the microSD are Internet access to the FreeRunner and an existing Linux distribution in the FreeRunner's internal flash memory; the factory install of OpenMoko is sufficient for this purpose. Installation occurs over the SSH command line to the FreeRunner. Installation requires the following steps:

5.1.1. Prepare the microSD card

Obtain a microSD card of at least 1Gb capacity. Note that not all microSD cards are compatible. We experienced problems with a SanDisk 2Gb card, with OM2007.2 claiming that it could not read the partition table or boot sector. A somewhat outdated list of supported cards can be found on the OpenMoko wiki[2].

5.1.2. Prepare the networking

Boot the FreeRunner into the Linux distribution installed in the internal flash memory (probably OM2007.2) and SSH in to get a command line terminal interface. Setup USB networking on the OpenMoko distribution with Internet access (so Debian can be downloaded and installed) – see Section 4.4, the process for setting up Internet access on OpenMoko is the same as for Debian. The USB networking interface should be enabled by default, and the USB networking part of the file `/etc/network/interfaces` should read something similar to:

```
auto usb0
iface usb0 inet static
address 192.168.0.202
netmask 255.255.255.0
network 192.168.0.0
gateway 192.168.0.200
up echo nameserver 192.189.54.33 >
/etc/resolv.conf
up echo nameserver 203.8.183.1 >>
/etc/resolv.conf
```

5.1.3. Setting up WiFi

To setup the ad-hoc wifi, the wifi part of the file `/etc/network/interfaces` should read something similar to:

```
auto eth0

iface eth0 inet static
address 192.168.2.202
netmask 255.255.255.0
network 192.168.2.0
gateway 192.168.2.200
wireless-mode ad-hoc
wireless-essid p2p
```

Note that this is an unencrypted and unsecure wireless connection. We could not get encryption to work on OM2007.2; hence, Internet access is not enabled over the wireless interface under OpenMoko.

[2] http://wiki.openmoko.org/wiki/Supported_microSD_cards

5.1.4. Download and Install Debian

Installing Debian to run from the microSD card can all be done from an installation script consisting of the following step-by-step instructions.

Download the Debian install script on the SSH terminal command line:

```
wget -O install.sh http://pkg-
fso.alioth.debian.org/freerunner/install.s
h
```

Then make the script executable by typing

```
chmod +x install.sh
```

The install we did on the FreeRunner did not create a swap partition; fdisk failed to create the partition table. We also set the boot partition to be FAT so it became unnecessary to modify the FreeRunner's boot environment.

The command used to install Debian was:

```
SD_PART1_FS=vfat ./install.sh all
```

However readers are advised to consult the following URL for more up to date instructions:

http://wiki.debian.org/DebianOnFreeRunner

5.1.5. Boot into Debian

If Debian installs without errors, it's time to reboot from the NOR. Turn off FreeRunner and then press the Aux and Power buttons simultaneously to boot to the NOR boot menu, choose the second menu option with the Aux button to boot from MicroSD (FAT+ext2), then press the Power button to execute, and Debian should boot!

5.1.6. Login to Debian

At this point USB networking should be enabled by default (settings from OpenMoko are ported over during the Debian installation process) and upon booting into Debian the user may expect to see either a command line interface or a phone GUI interface (Zhone). The user can then connect to the FreeRunner using SSH: USB and Internet access is enabled by default. The default password for the root user is empty. Once the user is logged in this should be changed using the `passwd` command.

5.1.7. Install software from the Internet

Use Debian's `apt` package manager to add/remove software packages, as this will ensure that package dependencies are always resolved.

Useful commands are:

```
apt-get update
```

This updates the list of available packages from Debian repositories online – the list of repositories is stored in `/etc/apt/sources.list`

apt-cache search package/s

This searches the cache of available packages, though packages are not always named in an obvious manner!

```
apt-get install package/s
```

This installs the package

```
apt-get remove package/s
```

This removes the package

```
apt-get upgrade
```

This updates Debian with the latest versions of installed software, patches, etc.

The default installation is lean so users are advised to install some useful packages such as:

```
build-essential
gcc
g++
libc6-dev
make: useful packages for compiling code
joe: an easy-to-use text editor (as Debian
only comes with vi)
man: to read manual pages for Linux com-
mands
mplayer: command-line media player
```

When installing new packages, `apt` will check for dependencies and request confirmation if additional packages are required. Users should always check dependencies to ensure that no essential packages are unintentionally removed.

5.1.8. Install CSound

Install CSound; the latest Debian packaged version available is 5.08:

```
apt-get install csound
```

Check and agree to the dependencies; for extra functionality search for and install additional CSound packages by typing:

```
run apt-cache search csound
```

5.1.9. Install PD

Install PD; the latest Debian packaged version available is 0.41.4-1:

```
apt-get install puredata
```

Check and agree to the dependencies; for extra functionality search for and install additional PD packages by typing:

```
apt-cache search puredata
```

5.1.10. Enable Wifi Hardware

The latest version of the Debian kernel[3] (at the time or writing) does not enable the wifi hardware by default when the FreeRunner boots up. The script below was created to wake up the wifi hardware; this will need to be run every time the FreeRunner starts up.

```
joe turnOnWifiHardware.sh
   export
   sys_pm_wlan=/sys/bus/platform/drivers/gt
   a02-pm-wlan/gta02-pm-wlan.0
   export
   sys_wlan_driver=/sys/bus/platform/driver
   s/s3c2440-sdi
   echo 1 | tee $sys_pm_wlan/power_on
   echo s3c2440-sdi | tee
   $sys_wlan_driver/unbind 2> /dev/null >
   /dev/null
   echo s3c2440-sdi | tee
   $sys_wlan_driver/bind
chmod +x turnOnWifiHardware.sh
```

To run the script, type

```
./turnOnWifiHardware.sh
```

To do this the script must be in the current directory otherwise the full filename path should be used; the current directory pathname can be retrieved with the command `pwd`.

5.1.11. Setup Ad-Hoc Wifi Networking

Create the following script to setup an ad-hoc wireless peer-to-peer connection; the WEP key - the string of 26 hex characters – and the ESSID - the ad-hoc wireless network name - can both be changed:

```
joe adhocWifi_ON.sh
   ifconfig eth0 down
   iwconfig eth0 channel 1
   iwconfig eth0 key <128-bit hex> essid
   "p2p"
   iwconfig eth0 mode ad-hoc
   ifconfig eth0 up
   ifconfig eth0 192.168.2.202
   ifconfig
chmod +x adhocWifi_ON.sh
```

Create the following script to take down the ad-hoc wireless peer-to-peer connection:

```
joe adhocWifi_OFF.sh
   ifconfig eth0 down
   ifconfig
chmod +x adhocoWifi_OFF.sh
```

[3] 2.6.28-20090105.git69b2aa26

To run either script, just type

```
./adhocWifi_ON.sh
or
./adhocWifi_OFF.sh
```

To do this the script must be in the current directory otherwise the full filename path should be used. The scripts will need to be run every time the FreeRunner's ad-hoc wireless network connection is set up or taken down.

5.2. Booting Neo FreeRunner

The FreeRunner can boot two systems: the original OpenMoko 2007.2 distribution (factory installed) from the internal flash memory, and Debian, which is stored on the microSD card.

To boot into the original OM2007.2 system, press the power button until the handset vibrates a little; this interface can be used for phone and SSH terminal functionality. We have not modified the OM2007.2 install aside from updating it and enabling WiFi and USB networking.

To boot into Debian for SSH terminal and CSound/PD functionality, press the Aux button and then the Power button, and keep both buttons depressed. The phone should vibrate a little until a boot menu appears; select the second option (Boot from microSD (FAT+ext2)) using the Aux button and press the Power button to execute. Debian should load and present the user with a command line.

The Debian interface can only really be used by accessing the terminal through SSH; the phone GUI (Zhone/Illume) doesn't work at the moment and we have yet to address and test X-Windows.

5.3. MUSIC SOFTWARE IMPLEMENTATION

At the moment of writing, Csound and Pure Data run successfully on both the Acer host platform and the Neo FreeRunner client; both programs produce sound on the Acer host platform but currently not on the Neo FreeRunner client. The special qualities of microtonally tuned sound produced at low power levels using Csound running on an untethered battery-powered processing device have already been demonstrated in a recent concert workshop[4] and will be demonstrated as part of this paper using either the host platform or, hopefully, the Neo FreeRunner client. We are also currently working on a demonstration to communicate control information via UDP between Pure Data applications running on the host and the client using Pure Data commands Netsend and Netreceive. This will enable the musical gestures produced by the motion of the untethered client device to interact with synthesis software running on the host platform. Alternatively, the client may become a mobile sound source that is modified while it is in motion. And unlike the Nokia phones used for the pocket gamelan, no purpose-built pouch is required for swinging the Neo FreeRunner (a gesture used with the pocket gamelan for creating Doppler shift) as it comes with a ready-made hole that is ideal for attaching a cord.

6. REFERENCES

[1] Wyse, L. and Mitani, N. "Bridges for Networked Musical Ensembles", in *Proceedings of the International Computer Music Conference*, Montreal, CA, 2009, forthcoming

[2] William Duckworth and Nora Farrell 2007 'iOrpheus' August 31, http://www.iorpheus.com/

[3] Essl, G. Rohs, M. 2006."Mobile STK for Symbian OS in *Proceedings of the International Computer Music Conference*, New Orleans, USA, 2006

[4] Schiemer, G., and Havryliv, M. 2005 "Pocket Gamelan: a Pure Data interface for mobile phones" in *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05),* Vancouver, BC, Canada pp. 51-56.

[5] Schiemer, G., and Havryliv, M. 2006 "Pocket Gamelan: tuneable trajectories for flying sources in Mandala 3 and Mandala 4" in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06),* Paris pp 156-161

[6] Schiemer, G. and Op de Coul, M. 2007 'Pocket Gamelan: tuning microtonal applications in Pd using Scala' in *Conference Proceedings of the Australasian Computer Music Conference* June 19th-21st pp. 81-86.

[7] Schiemer, G. *MIDI Tool Box: interactive tools for music composition*. (Phd thesis) Sydney: Macquarie University, March 2009

[4] *Butterfly Dekany* performed at GAUNG 21st Century Global Music Education Bali, 29th April 2009

TOWARDS PARTICIPATORY CREATIVE SYSTEMS USING *MOBILE PROCESSING*

Christian Haines
Electronic Music Unit
Elder Conservatorium of Music
University of Adelaide
Adelaide, South Australia

ABSTRACT

In recent years interest has resurged in the field of creative networks and the participatory spaces they afford, especially in areas associated with computer music. Much of this interest has circled around small-scale networks consisting of cumbersome and specialised computer systems using sophisticated software. This investigation seeks to extend the scale of these networks, and hence participation, through the ubiquitous, intuitive and readily available technology of mobile phones. By extending the scale this author believes that unseen avenues of investigation will be revealed and new creative outcomes will occur.

1. INTRODUCTION

Historically the idea of a node, an interconnected point in space, represents a useful, although simplified, way to describe the players in creative participatory spaces. Classical musicians in an orchestra can be thought of as nodes, connected passively through a score and actively through a conductor and sound. An electro-acoustic ensemble may actively improvise around a set of principles or rules, whilst simultaneously having their contributions electronically manipulated by other players using varying degrees of active and passive engagement. Regardless of its simplicity, the concept of the node (or nodes) represents a suitable construct for the exploration of participatory creative structures moving beyond historical or normative models. Further, it enables players in these spaces to be considered 'unclassified,' where the notion of author shifts between audience, composer, performer and technology.

Nodal systems are an area of computer music that have undergone significant development and resurgent interest in the past decade or so. Drawing from the work of luminaries such as John Cage, with Imaginary Landscape #4, to David Tudor, and his various versions of Rainforest, researchers such as Gil Weinberg at the MIT Media Laboratory [21] and Atau Tanaka, Nao Tokui and Ali Momeni at the Sony Computer Science Laboratory in Paris [19], have sought to understand and chart nodal systems within participatory creative spaces. These systems include distributed and networked music, interconnected musical networks, interactive music systems and social music systems. Using this research as a point of reflection and drawing inspiration from the work of sound artists such as Mark Shepard and his Tactical Sound Garden Toolkit [14] and Scheimer and Havryliv's Pocket Gamelan [13], this investigation seeks to engage many of the same concepts and frameworks but shift them collectively to a larger, more volatile scale of investigation, embracing what Kim Cascone alludes to in the article, *The Aesthetics of Failure* [3].

2. PARTICIPATORY CREATIVE SYSTEMS

The investigation presented here is centred on the idea of developing a Participatory Creative System (PCS). A PCS can be thought of as large-scale creative works organised around a network of multiple nodes, where each node actively or passively contributes to the realisation of the work by permitting and/or promoting the interaction and exchange of ideas. A number of points will be drawn from a recently realised first incarnation of the work *[SOMETHING TO GO HEAR]* [5], a work for multiple mobile phones and software metronome which drew inspiration from the Fluxus-like work by György Ligeti *Poème Symphonique*[1] [10]. Essentially the work explores the notions of process music, distributed computing, spatialisation and technological determinism, with device variation yielding unforeseen sonic trajectories.

Essentially the work attempts to recreate aspects of the Ligeti piece. Audience members are provided with a software metronome, developed in Mobile Processing, for their mobile phone. The software on each phone produces a metronomic pulse with a differing tick, tempo and wind (potential energy that sets a duration limit on activity). At the beginning of the work audience members are asked to activate their software metronomes. Once activated each phone produces a short pulse of a sampled or MIDI-based metronome tick from its speaker. Over the course of the work a cluster of pulses is heard, with ticks phasing in and out of one another both temporally and spatially. Due to the varying tempos and winds the phones gradually stop at different times, revealing new sonic layers until all the phones have stopped.

[1] For 100 Metronomes.

2.1. Nodes

Each node in the network represents an individual or group of individuals contributing to the realisation of a creative work through means of a system. The individual is not necessarily artistically trained and may or may not have a clear sense of the overall direction or objective of the work. In essence, each node is a participant with their own specific function or role to perform, this role being determined both by the overall objective of the PCS and facilitated by the individual system used. Nodes may interact with one another passively, for example by simply performing together and having their musical or sonic results mixed acoustically. Subsequent variations and interactions can then be determined by the participants interpretation of that result. Finally, nodes may interact actively, for example in the case where the behaviour and interactions are a product of the sum of participant interactions and controls prior to acoustic propagation.

2.2. System levels

The system used in each node can vary significantly depending on the PCS but typically operates at two levels. The first of these is the node level system, a technological system designed to facilitate and encourage individual or group participation in the work; the second is the network level system which can either be considered as the sum of the node level systems, or as node level systems conducted and/or directed by other nodes.

2.3. Participation

Each aspect of the system makes use of a technology, combining software and hardware, whose inherent qualities and explicit design promote participation. Participants are empowered to contribute to the realisation of a creative work because:

(a) they have ready access to the hardware required to contribute;

(b) the software is provided for free, adheres to principles of perceptive interaction design and encourages play;

(c) system interaction at global network level is intuitive in terms of its behaviour.

2.4. Objectives

The key objective in developing a PCS is to foster the possibilities that participatory networks open up, the questions they raise and the inherent potential of the system. This includes the possibility of revealing pathways for new creative interpretation or activity through large-scale clustering of nodes (participants). In order to achieve this key objective, participation must be:

(a) (easily) *facilitated*, making use of readily available hardware and software, including intuitive interactions and providing explicit causality within the networks;

(b) *transient*, allowing participants to engage as part of a work but disengage readily without affecting the sum of the work;

(c) *scalable* (within reason), enabling the number of participants to reduce or increase whilst still producing a cohesive and relevant creative result;

(d) *socialised*, as discussed in *Facilitating collective musical creativity* [19], moving beyond the notion of participants as flaccid nodes, utilitarian in their role in providing technology in a work, to the idea that they themselves validly contribute and engage with other participants meaningfully;

(e) of a *high magnitude*, with network participation, and hence the amount of nodes, at a mass critical enough to begin to draw a sense of the potential of the work and its ability to raise new questions.

3. MOBILE PHONE AS AGENT

As can be noted with other technological platforms, a mobile phone[1] is a combination of hardware, operating system and software. The combination of these elements, coupled with the device's interaction design, determines the inherent capability and value of the device in terms of its creative agency. Creative agency in this sense relates to the kinds of creative outputs that can be produced using the mobile phone and the inherent participatory pull or allure that such devices have in relation to the socialised aspect of the work i.e. facilitating participation. In the case of *[SOMETHING TO GO HEAR]* participation was easily facilitated through hardware profiling, Internet and point-to-point network distribution. Further, users were able to readily test the software through a simple and direct user interface.

3.1. Hardware

In considering the hardware, a mobile phone is a ubiquitous[2] mobile computing device. As with other mobile computing devices, the hardware is based upon an integrated and cooperative array of components including a CPU, permanent memory, RAM, standard

[1] Although a mobile phone represents what some may consider a fairly standard profile of technologies, it is worth noting that such profiles vary enough in the case of this examination to be deemed very different technologies. As such the mobile phones discussed in this paper are those that utilise the Java Virtual Machine (JM) and at least in this sense provide a degree of cross-platform operability. Higher profile phones, such as Windows Smart Phones, Apple's iPhone and Google's Android were not considered.

[2] Ubiquitous in this sense refers to the idea the phone is readily available, common technology, used in abundance in modern industrialised societies.

inputs (keypad, buttons, microphone, Bluetooth, radio telephony and WiFi*)* and standard outputs (display, speaker, Bluetooth, phone signal and WiFi). As with other devices, the hardware itself places a number of restrictions on the kinds of creative works that can be produced. Specifically, mobile phones fit into the category of devices known as resource constrained.

3.1.1. Processor and memory

Of particular note in relation to the impact of hardware on the creative agency are the limitations on the kinds of processes that the CPU can deal with at a time and what is able to be stored in memory for such processes. Standard generation phones (i.e. not smart phones) vary in the speed from a few dozen to several hundred megahertz. Further, the kinds of instruction cycles that are able to be executed will likewise vary. Suffice to say the inherent processor strength of mobile phones is modest compared to PDAs, laptops and desktop computers.

3.1.2. Speaker

In terms of sound output, mobile phones usually provide options such as ear speakers, loud speakers, wired and Bluetooth headsets. In respect to frequency range and dynamics, headset options usually provide a better quality signal than a speaker. Speakers typically suffer as a result of the phone's physical structure, and hence acoustics, as well as the speaker size and design. Of paramount concern for some manufacturers is the cost-benefit trade-off of having recognisable 'voice' versus speaker price, thus usually resulting in low fidelity speakers. This said, in terms of performance logistics and simplifying participation, the speaker phone is the most suitable option as it provides a convenient and simple method of sound propagation.

3.2. Operating system

As with other computing devices, the operating system of mobile phones varies considerably between manufacturers and is usually proprietary. For example, Sony-Ericsson, Nokia, Samsung and Motorola all provide software development kits (SDKs) for use in developing on their mobile phone operating systems. In an effort to better support and encourage software development, particularly across platforms, many manufacturers have chosen to support the Sun's Java Platform 2 Micro Edition (J2ME)[15] in their operating systems.

J2ME "provides a robust, flexible environment for applications running on mobile and other embedded devices…. Applications based on Java ME (J2ME) are portable across many devices, yet leverage each device's native capabilities." [15]. J2ME runs within the Java Virtual Machine (VM) that sits on top of the mobile phone's operating system. Although this virtualisation simplifies many of the tasks for development, it introduces a notable performance sacrifice with respect to processing cycles [6]. The configuration of devices and compliance with respect to J2ME is determined by the Connected Limited Device Configuration (CLDC)[18], which determines the framework by which low level programming interfaces can interact with the phone's hardware and operating system. On top of this, and complicating matters further, is the Mobile Information Device Profile (MIDP)[16], a specification that determines a range of APIs to access the phones hardware and OS feature set. Simply put J2ME provides a cross platform means to develop software for resource constrained devices, such as mobile phones, but paradoxically does so in a manner that constrains those resources further.

3.3. Programming environment

Since circa 2000, for the purposes of creativity, there has been a proliferation in the number of programming environments that seek to provide higher levels of abstraction to low level features of devices. One environment that has received significant recognition, and a growing user base, is Ben Fry's and Casey Rea's *Processing* [4]. *Processing* "is an open source programming language and environment for people who want to program images, animation, and interactions" [4]. Building upon the foundation of *Processing* is Francis Li's *Mobile Processing* [7]. *Mobile Processing (*MP) seeks to extend the objectives of *Processing* to Java powered mobile phones. This is undertaken by providing a simple, intuitive environment through which mobile phone applications can be authored, tested and deployed. Although MP does open up the mobile phone for artistic exploration, it contains a reduced feature set of the environment *Processing*. Further, and similar to *Processing*, it has limited features in the areas of event timing, MIDI and sound input/output/playback. Some of these limitations have been overcome through additional base and third-party libraries, particularly those developed by Mary Jane Soft [11], which has spent considerable energy providing MP with access to new functions and processes in J2ME, such as extended MIDI, sound and communications.

3.4. System variability

Although MP provides a pathway for artists wishing to develop mobile phone applications without requiring specialised knowledge of programming, its stability is undermined by cross-platform compatibility and language abstraction levels. Stability can be defined as the consistent and uniform behaviour of a device in relation to its set tasks, regardless of variability. Variability describes the disparity that results from the interconnectedness between differing hardware profiles, the components that make up a hardware profile, phone carrier operability, operating system implementation of virtualisation, compliance and implementation of the virtualisation, and the programming environment's abstraction of the language.

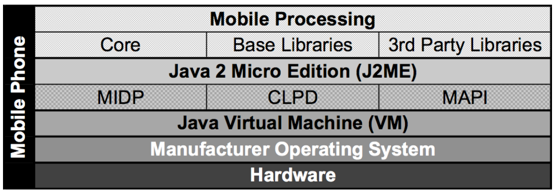


Image 1 - Mobile Phone System Layers

Given the number of layers, development can be wrought with difficulties, particularly in respect to producing consistent results. In this regard, developing multiple safe guards is a must to ensure processes can be *tried* and problems caught. Regardless, there have been instances where *trying* everything means *catching* everything, and no processes have been left functioning.

Obviously a choice could be made to directly engage with the phone and its system through manufacturer specific SDKs, or even by using J2ME directly. However, it is clear there is a learning curve to be addressed before any meaningful artistic product can be produced, and this may effect artistic participation. This raises a point of tension between the need-to-know, the need-to-do and the balance between flexibility, ease of use and whether or not to treat unintended consequences as issues or idiosyncratic and endearing personality traits presented by the technology. Further, even models from the same generational family or of exactly the same make have shown inconsistent behaviour.

For example, during *[SOMETHING TO GO HEAR]* two phones of exactly the same manufacturer and model (Nokia 6120) produced longer movements (ensuing in a duel lasting longer than the other fourteen participants) and one sounded a completely unanticipated rapid staccato of notes close to finishing. The timing of the movement was fixed, leading to the only explanation for this behaviour being that the phone had some kind of inherent problem.

4. ISSUES IN IMPLEMENTATION

Typically, each creative programming environment aims to serve either a particular niche or a broad set of creative objectives. In many instances where the prescribed range of objectives is narrow, for example visually-centric, there are often individuals who are keen to extend the functionality to other realms, for example music and sound. *Processing*, whose objectives are stated above, is situated in a visual context and has been extended through third-party libraries. Similarly, MP has undergone such an extension. However, regardless of the apparent evolution of the environment, some fundamental constraints built into the gene pool present ongoing challenges for those desiring to work with sound and/or music.

4.1. Frame based timing

The most notable issue with MP is that of timing. MP inherited a frame-based timing mechanism from *Processing*. In short, divisions of time are determined by the frame rate (FR) i.e. a number of frames per second (FPS). The FR determines the speed at which events are produced in the environment and the speed that the particular device is able to handle adequately.

4.1.1. Direct frame rate to BPM conversion

A direct frame to tempo conversion yields a basic range of tempos to work with. For example, a FR of 25 FPS can, in theory, produce 1500 frames or pulses or beats per minute. Therefore, shifting the FR up and down will produce different and somewhat awkwardly spaced, tempos (Table 1).

Frame Rate	30	25	20	15	10	5	4	3	2	1
BPM	1800	1500	1200	900	600	300	240	180	120	60

Table 1 –Frame Rate to BPM Conversion

4.1.2. Skip frames to diversify tempo

As has been discussed elsewhere by those who develop music applications using frame-centric technology (including Adobe's Flash and Director)[1], getting finer increments of tempo requires rethinking the paradigm of timing in terms of division of frames. This idea can be achieved by using Frame Skip (FS), which skips over a number of frames at a fixed FR before issuing a beat or pulse, and this provides finer increments (Table 2).

Frame Rate	25										
SKIP	30	25	20	15	14	13	12	11	10	9	8
BPM	50	60	75	100	107	115	125	136	150	167	188

Table 2 - Frame Rate and Skip to BPM Conversion

4.1.3. Frame-to-Tempo lookup table

A finer degree of tempo resolution can be further achieved by simultaneously varying both the FR and FS. A pre-calculated lookup table, indexed by FR and FS, provides the best apparent degree of tempo control within this restricted visual paradigm. Having said that, the implementation of this as a real-time control in *[SOMETHING TO GO HEAR]* introduced noticeable timing artefacts, such as the tempo never accurately resolving to the newly selected tempo during run-time operation. As a result, the method was withdrawn from use until further development could resolve the issue (Table 3).

		FPS					
		30	25	20	15	10	5
SKIP	50	30.0	24.0	18.0	12.0	6.0	4.8
	40	37.5	30.0	22.5	15.0	7.5	6.0
	30	50.0	40.0	30.0	20.0	10.0	8.0
	20	75.0	60.0	45.0	30.0	15.0	12.0
	15	100.0	80.0	60.0	40.0	20.0	16.0
	10	150.0	120.0	90.0	60.0	30.0	24.0
	5	300.0	240.0	180.0	120.0	60.0	48.0

Table 3 – Frame and Skip BPM Lookup Table

4.1.4. Thread sleep alternatives

Given that MP controls the FR via a system or thread sleep, and can in fact make use of this thread sleep for finer timing control, MP could become more attractive for those requiring precise timing control. As has become a familiar story, there are trade-offs in providing highly abstracted languages that promote ease of use, but reduce the level of control by abstracting to functions and classes. An example of this is embedding thread sleeps under the umbrella of frame rate control. Further, the degree of thread sleep is finite, given processor cycles require a certain period of time in which to operate.

4.1.5. Error offset in timing and absolute time

Although the above methods of tempo control at least appear to provide a stable basis by which events can be issued, even if increments are restricted, there is an error offset between those events. This varies as a ratio and is dependent upon the method chosen, such as FR to BPM conversion, FR with FS to BPM conversion and variable FR with variable FS to BPM conversion. For example, on a Nokia 6288 at a FR of 25, beat divisions should be 40 milliseconds but in actuality they vary from 36 – 44, representing an error of between 0 and 10%. The main issue resulting from this relates to the introduction of a 'swing feel' to the timing, which at certain FR and FS combinations becomes heightened and almost unpleasant.

Nokia 6288 with timing sketch (no screen drawing)								
Beat	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Interval (ms)	43	38	37	41	40	44	36	42
Offset (ms)	3	-2	-3	1	0	4	-4	2
Error (%)	7.5	5.0	7.5	2.5	0.0	10.0	10.0	5.0
Frame Rate: 25	**Frame Skip: None**		**BPM: 1500**		**Interval (ms): 40**			

Table 4 - Frame Rate Timing Errors

4.1.6. Interaction with display elements

As mentioned previously, the apparent sluggishness of the VM means that switching tempos dynamically mid-stream is not an elegant proposition. Further, and importantly, MP can suffer severe 'rate pull down' when utilising certain classes and functions, particularly those that generate display output. For example, on a Nokia 6288 the frame rate can operate optimally at 20 FPS or a pulse every 50 milliseconds. However, once user interface elements are included as part of the onscreen render, this degrades down to around 80 milliseconds or 12 FPS. This is a notable and unacceptable deterioration in performance and, in the case of using a tempo look-up table as discussed above, restricts the timing resolutions that can be used.

4.1.7. System variability and frame rates

Finally, as mentioned before, the makeup of the device and its variability dictates how efficiently and consistently tasks are handled. The issue here in terms of promoting a socialised use of technology, and hence participation, is that a device's tempo range and stability will differ significantly between participants, thereby typically producing nodes that are ineffective or simply non-cohesive. Although such consequences are unintended, there maybe instances, as mentioned before, were the results are desirable and not unwelcome.

4.2. Sound

4.2.1. General capabilities

The base level sound capabilities of MP are restricted, and are provided through a library. In terms of MIDI, only basic tone playback is provided with pitch and volume selection. Notably, the default tone that is selected for MIDI playback varies in type and quality between devices. Further, attempting to instantiate multiple voices can produce memory errors and/or device crashes. With regard to sound file playback, only certain audio file types are supported with basic volume control, and audio capabilities are clouded by the fact that devices vary in the file types they support. As with MIDI, multiple instantiations of audio file playback can also produce memory errors and device crashes. For example, in developing the software for *[SOMETHING TO GO HERE]*, two systems for metronome playback were developed (one MIDI-based and one audio-file based) in an attempt to provide a safe guard option if the mechanism did not work. In one instance, with a Nokia E51, neither option functioned, although the MP phone profiler [9] indicated both methods were supported.

4.2.2. Extensions

Mary Jane Soft's MSound, MSynth and MAudio3D libraries [11] provide welcome extensions to the base sound capabilities of MP and build upon the J2ME technologies tied with the MIDP 2.0 specification [17]. These provide access to MMAPI 1.1 (Mobile Media API) [16] and a number of functions including: mixing sound streams; performing audio capture and playback; a relatively extensive range of PCM and CODEC audio file types; simulation of 3D sound; basic sound generation and synthesis; and extended MIDI functionality including instrument selection and channel use. Again it can be noted that these libraries extend the range of layers above the system and could, in turn, produce their own idiosyncrasies within the system. For example, MIDI program change initialisation failure has required the rather ad hoc solution of loading a MIDI file and playing it back briefly.

4.3. Other issues

A number of other issues have arisen during the course of exploring the viability of MP and mobile phone technology as a socialised participatory device.

4.3.1. *Memory handling*

As mentioned previously, some of the memory handling with respect to voice creation and stopping, i.e. event timing or instrument usage, is problematic within certain families of phones such as the Nokia 6200 series and the Sony Ericsson K series. In the case of preparing *[SOMETHING TO GO HEAR]* this was notable and represented a fundamental reason for different sound generation systems being introduced.

4.3.2. *Array executions*

In some cases array instantiation produces what the system deems as malformed arrays, often leading to 'out of bound' errors on particular devices, whilst other devices perform the same function without issue. This is highly suspect given this issue can often occur upon initialising an application, which then proceeds to operate correctly after the next start-up. In the performance of *[SOMETHING TO GO HERE]* this required some participants to do a 'warm start' by starting their device and permitting the crash to happen, thereby allowing it to function correctly for the performance.

5. ADVANTAGES IN IMPLEMENTATION

5.1. Universal

Despite many of the reservations expressed, MP's implementation of J2ME provides a relatively universal platform to give programmers access to different devices. Most reservations concerning inconsistencies lie with device manufacturers and their implementation with respect to hardware. This is a key consideration in terms of facilitating the development of participatory systems, and perhaps allowing participation to extend to the area of application building or modification. This could take place, for example, by providing PCS frameworks that can be modified, changed and redeployed by participants or those wishing to implement their own systems.

5.2. Ubiquitous

The ubiquity of mobile phone / computing technology, coupled with platform universalisation, provides a relatively solid foundation by which applications can be optimistically deployed and run on a large scale. Although based on the experience of developing and realising *[SOMETHING TO GO HEAR]*, it is clear that one should proceed with caution and aim to provide a number of safe guards, such as alternate versions or capabilities. Similarly one should be prepared for some systems not to participate and gauge the effect this has on the overall realisation of a work.

5.3. MIDI and sound functionality

Although primitive, the inclusion of MIDI and sound functionality does provide a basis by which sonic or musical outcomes can be produced and, as mentioned, Mary Jane Soft's libraries have taken these functions further since 2007, providing avenues for exploration. The functionality available in mobile systems can be likened to looking at modern technology through a lens of times past. Although we see a modern communications device, the sound and music capabilities are seemingly as primitive as those seen and used by founders such as Tudor, Cage, Dodge and Lansky et al, between the 1960s and 1980s. Regardless, a PCS seeks to leverage such basic capabilities through the clustering of devices and revealing pathways of new creative interpretation or activity, rather than seeking totalised outcomes through an extensive and complex array of capabilities in one device.

5.4. Communications systems

One of the main attractions with mobile phones as a nodal participatory technology lies in its ability to communicate in a multitude of ways, such as radio telephony, short message service (SMS), multimedia messaging service (MMS), Bluetooth, WiFi and GPRS. These methods provide different mechanisms by which information can be transmitted, whether these mechanisms are: short range point-to-point, such as Bluetooth which is useful for rapid connectivity and low data exchanges, as seen with games; broader exchanges through intranets or the internet where centralised repositories of information or distant geographical interactions are required; radio telephony allowing for the feed of voice or other acoustic spaces to another location; the highly socialised use of SMS which has been exploited in range of mobile phone based artworks; and GPRS, which provides localised information and positioning. In short, the range of communication options available presents a pivotal advantage by promoting exchange or transmission of content. Further, due to participants typically being familiar with these systems, it provides an obvious means by which creative applications can encourage participation. Finally, the breadth of simple information types, such as voice/audio, text, photo and video, that can be collected and transmitted easily without training in new systems provides a range of possibilities for different creative works.

6. FUTURE DIRECTIONS

Although this examination of mobile phones, as a central part of participatory creative systems, is still in its infancy, it is clear from the explorations in *[SOMETHING TO GO HEAR]* that the majority of effort has been expended mediating the vast variety of devices. Bearing this in mind, the following core objectives have been identified:

(a) Reducing support by developing systems that provide safe guard positions which are intuitive, graceful and provide feedback to the participant. This will no doubt include profiling system functionality on a more detailed level and collating the information in a central repository, similar perhaps to the phone profiling used by Francis Li [9].

(b) Extending multi-system sound playback. As the development of PCS is situated from a context of working with sound, this objective is obvious and essential.

(c) Extending communicative interoperability between devices and beyond the socialised and acoustic realms, allowing the network to perform more like a traditional network and leveraging interactions similar to those discussed by Weinberg [21].

7. CONCLUSION

In summary, a participatory creative system seeks to build on and contribute to the field of network music and interactive music systems by extending the scale and level of participation within a localised and socialised space. By extending the scale it is believed that new creative opportunities and questions will be revealed. Further, by extending the scale it is duly noted that trade-offs have been made in terms of what kinds of systems can facilitate participation and also actively contribute meaningful creative capital to the outcome.

The mobile phone can be seen as a suitable candidate and facilitator for a PCS. However, the mobile phone does provide a point of tension by advancing a ubiquitous platform that promotes contributions from participants with a device that is inherently variable and reduced in capability when compared to other modern technologies. Rather than seeking absolute resolution of issues associated with variability, within reason, this investigation embraces variability as tenet of technological determinism and looks to the personality and potential that this variability can bring to the outcomes and social space of the participatory creative system.

8. REFERENCES

[1] Adobe. "Adobe". 2009. 20/4 2009. <http://www.adobe.com/>.

[2] ARTE. György Ligeti - Poème Symphonique for 100 Metronomes. Video. ARTE (France).

[3] Cascone, Kim. "The Aesthetics of Failure: “Post-Digital” Tendencies in Contemporary Computer Music." Computer Music Journal 24.4 (2000): 12-18.

[4] Fry, Ben and Reas, Casey. Processing. Vers. 1.03. Computer software, 2009.

[5] Haines, Christian. [Something to Go Hear] Rec 16/4/2009. Audio, Adelaide, 2008.

[6] Java User Group Community. "Tastephone Server". 2009. 20/4 2009. <http://www.club-java.com/TastePhone/J2ME/MIDP_mobile.jsp>.

[7] Li, Francis. Mobile Processing. Vers. 0007. Computer software, 2004.

[8] Li, Francis. "Mobile Processing Reference". 2009. 20/2 2009. <http://mobile.processing.org/reference/>.

[9] Li, Francis. "Phones". 2009. Profiling area for mobile phone compatibility with Mobile Processing. 20/3 2009. <http://mobile.processing.org/phones/index.php>.

[10] Ligeti, György, and Pierre Charial. Mechanical Music. sound recording. Sony Classical, New York, 1997.

[11] Manrique, Marlon.J. Mobile Processing Libraries. Mary Jane Soft 2009.

[12] Nokia. "Nokia 6288". 2009. Nokia. 14/9 2008. <http://www.nokia.com.au/A4519121>.

[13] Scheimer, Greg and Havryliv, Mark. "Pocket Gamelan: An Extensible Set of Microtonal Instruments." Australasian Computer Music Conference - Generate + Test. ACMA.

[14] Shepard, Mark. "Tactical Sound Garden". 2008. 17/11 2008. <http://www.tacticalsoundgarden.net/>.

[15] Sun Microsystems. "Java Me at a Glance". 2009. Sun. 20/4 2009. <http://java.sun.com/javame/index.jsp>.

[16] ---. "Mobile Media API". 2009. Sun. 17/1 2009. <http://jcp.org/en/jsr/detail?id=135>.

[17] ---. "Mobile Information Device Profile 2.0". 2009. Sun. 17/1 2009. <http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html>.

[18] ---. "Connected Limited Device Configuration 1.1". 2009. Sun. 16/1 2009. <http://jcp.org/aboutJava/communityprocess/final/jsr139/index.html>.

[19] Tanaka, Atau, Nao Tokui, and Ali Momeni. "Facilitating Collective Musical Creativity." Proceedings of the 13th annual ACM international conference on Multimedia. ACM.

[20] Tudor, David. Rainforest (Version 1). Compact Disc. Ellipsis Arts, 1990.

[21] Weinberg, Gil. "The Aesthetics, History, and Future Challenges of Interconnected Music Networks." International Computer Music Conference. ICMA.

A GENERATIVE ENVIRONMENT FOR PERFORMING CONTEMPORARY ELECTRONIC MUSIC

Sarah Keith

Macquarie University
Department of Media, Music, and Cultural Studies

ABSTRACT

Performing contemporary popular electronic music commonly incorporates audio playback, either from digital audio files or vinyl records. A software environment (Deviate) has been constructed, which employs generative methods for performance and control and allows new note-level musical material to be created in real-time. This discussion focuses on the contextual issues of using and designing this system, with regard to contemporary methods for computer-based and laptop performance, and critiques of these approaches. Standards for performance environment design and practice are introduced, which outline primary areas to be addressed in the construction of a performance environment such as Deviate. Audio examples of Deviate are located online, at http://www.cetenbaath.com/cb/about-deviate/.

1. INTRODUCTION

Contemporary electronic music, as it will be called in this paper, refers not to contemporary electro-acoustic music but to a diverse genre of music including and derived from popular electronic music styles of the 1980s to present, such as techno, industrial, and house. The performance of this music has historically taken several forms, ranging from turntable-based DJ sets to live programming and playing of sequencers and synthesisers. In recent years, music production and performance software programs have digitised DJ practice. The most prominent of these programs is Ableton Live, which allows users to cue loops and samples for selection and playback in real-time. The author's research concerns the incorporation of generative compositional methods to the performance of contemporary electronic music, where musical material is generated 'live' at the note-level according to stochastic processes and user-determined control parameters. This approach permits the creation of new musical material in performance and affords greater flexibility and scope for improvisation.

This performance model used in this research is that of a single performer with laptop computer, with no acoustic or instrumental input. A software performance environment (named *Deviate*) has been developed using Max 5 [1] and includes systems for percussive and melodic generation, synchronisation, and control of musical features. This paper focuses on the context surrounding this project, and will discuss its relation to other methods of generative performance, programmatic and stylistic formalism, and improvisation. Functional and lower-level aspects of *Deviate* are not covered here; see Keith [2] for a discussion of practical features.

This research is innately practical, and investigates possible solutions to a perceived issue, namely, the separation of the methods for production and performance in contemporary electronic music. To redress this, *Deviate* provides a means for producing note-level musical output in real-time, placing it closer to the role of a performable musical instrument rather than a storage and playback mechanism. The motivations for this project are therefore firstly founded on a musical and cultural basis, and secondarily on the fields of computer music, algorithmic composition, rule-based logic, and programming. This style-first, result-oriented approach aims to situate this research within a musical context, and develop generative techniques and performance practices to suit that practice. To date, *Deviate* has been used for numerous live recordings in various sub-styles of contemporary electronic music. For examples of these, see http://www.cetenbaath.com/cb/about-deviate/.

2. CURRENT APPROACHES

The motives for generative music composition and inquiry encompass music and information analysis [3], perception research and cognitive science [5], an exercise in programming [6], and the development of compositional tools [9]. The motivation underlying *Deviate* is comparatively prosaic in that it employs generative techniques as a practical method for augmenting live performance of contemporary electronic music. This endeavor is a response to the author's own perception that popular methods for laptop-based contemporary electronic music performance, such as Ableton Live [10], do not allow the user to extemporize or create new note-level material in performance.

2.1. Performing contemporary electronic music

As a production-based rather than performance-based musical form, contemporary electronic music has limited conventions for live performance. Turntable-based DJ mixing practice is perhaps the most well-known approach, where a volume crossfader is used to blend separate audio sources, merging the output of two vinyl records while maintaining a steady beat. In this scenario, the DJ exerts little input regarding the compositional structure of the sonic result. Virtuosic turntablism practices have evolved in hip-hop and other popular music forms, however, where sound sources are adeptly manipulated and recombined in order to generate an original work during performance [11], but for the most part performance constitutes playback of existing works. Other approaches may incorporate technologies of music production, including various configurations of software, sequencers, and synthesisers, but there is no unified method that exists across practitioners and sub-genres. In recent years, Ableton Live has come to the fore as a a software package that merges live performance and production capabilities, and is widely used by DJs as well as music practitioners in other genres. This research project is, however, examining only the model of single performer and laptop computer, and does not consider collaboration with other musicians, audio input, or additional musical instruments.

2.2. Non-linear and improvisatory approaches

An element that is underdeveloped in performance programs such as Ableton Live is the facility for improvisation, although it is patently successful in its approach to live music-making as a whole. The 'Session View' performance mode allows for real-time decisions to be made over phrase-level musical output and compositional form, by recombining pre-composed loops or via real-time control over audio effects and processing. Ableton Live's ethos focuses on this notion, as demonstrated by their "Defy the Timeline" slogan [12]). This concept is more accurately described as non-linear playback rather than improvisation, as output is wholly deterministic. Adding a measure of indeterminacy places the performer in an improvisatory and *interactive* rather than *reactive* role, as s/he is obliged to react and adapt to sonic output. Furthermore, the production of new material in performance is constrained by Ableton Live's approach. Although sections and loops can be selected, layered, recombined, and processed in real-time, creating new note-level material without recourse to external instruments is problematic.

3. MOTIVES FOR CHANGE

Given that the performance approach engendered by Ableton Live is acceptable to many DJs and performers of contemporary electronic music, the question might be raised: why improvise at all? There exist many concerns [13] raised in both academic and wider contexts that suggest that it may be useful to rethink current approaches to laptop performance. These concerns relate, in several forms, to the development and perception of skill. The areas discussed below highlight how *Deviate*'s approach and function in performance aim to address these issues.

3.1. Skill and effort

With the advent of the laptop as an instrument of performance came the decline of motoric skill. The performance template outlaid by turntable DJs has been implanted in a digital setting, retaining the fundamental practices of recombination and beat-matching while replacing the physical expertise necessary to achieve this with the "office-style user interface" [14] of mouse and keyboard. The ideal of skill in performance is crucial, and technology has had an undeniably complicating effect. Godlovitch, in his study of musical performance, cites both musical skill and "appropriately creditworthy physical skill" as pre-requisites for model performance [15]. Regarding technology, he states, "It gives anyone with minimal effort and skill the power to create the very results for which the musician has spent years in training…if society values musicians largely for their results, the value of musicians declines" [16].

This fatalistic view may not be universal, but nonetheless shows how value is tied to notions of physical and practical skill, and how skill differs within acoustic and technologised performance. Addressing this by attempting to develop physical skill on mouse, keyboard, or other control devices to a level beyond that of any standard computer user is unlikely to yield satisfactory results, although Collins [7] does suggest incorporating typing practice into a live coding exercise regimen. An alternative approach is proposed by d'Escrivàn, where effort and physical skill are circumvented by reducing the value of performance to intentionality [17]. The following question is therefore how intentionality can best be expressed in musical performance. To achieve this, the performer must demonstrate the capacity to make skilled, rather than routine or perfunctory, musical decisions, and also be able to realise new and original directives rather than be constrained by limited options. *Deviate* aims to address this by incorporating indeterminate and generative processes, formulating skill as the ability to interact with and improvise with new and complex material.

3.2. *Grain* and performance

An issue related to the perception of skill is the possibility for error and the notion of *grain*. This uncodifiable quality is defined by Barthes with reference to Kristeva's notions of pheno-text and geno-text [18], and constitutes "the body in the voice as it sings, the hand as it writes, the limb as it performs" [19]. Grain thus represents the elusive qualities that are created through the performance of a work [20], generated materially and resistant to systematisation. As digital audio files present a flawless representation of themselves in each instance of playback (speaker and mechanical error notwithstanding), there is minimal space for grain in this type of performance. Notions of effort and virtuosity can be framed in terms of grain, where surmounting the inherent difficulties and idiosyncrasies of a musical instrument translates to the perception of skill, and thus to appreciation by an audience. The cultural antipathy towards technology in musical performance can be expressed as a tension between the desired and elusive grain (the "human element" of performance) and the perfection of the machine. By incorporating generative processes resulting in output that cannot be wholly foreseen, *Deviate* aims to introduce grain by creating music that is unique to each instance of performance.

3.3. The significance of spectacle

The effectiveness of any approach is additionally contingent on the performer's actions being understandable to the audience. Croft [21] suggests that developing scrutable relationships between performer action and sonic response is essential to instrumental laptop performance. Cascone [22] introduces the notion of "counterfeit", a term which refers to the audience perception of falseness that occurs"when a performer generates music by a process unknown to the audience; using technology more at home in an office cubicle than a musical performance". This concern is echoed by Davis [23], who declares that most laptop musicians are "boring to watch" and "often there isn't even a visible link between a keypunch and a specific change in sound. Is it live or is it Memorex?" Given that the majority of home computers contain pre-installed music playback and music-making software, and that the laptop screen enforces a physical (acousmatic) barrier between performer and audience, it is understandable that audiences would assume the use of pre-recorded samples or sequences in laptop performance. This perceived use of playback devalues laptop performance to audiences. Incorporating perceivable improvisatory practices to performance is one method for reducing "counterfeit" and the perceived externality of the performer to laptop performance; this project serves as an exploration of this possibility.

4. STYLE SYNTHESIS AND MODELLING

The discussions above cover the wider cultural and contextual reasons for incorporating generative and improvisatory practices into laptop performance. Adapting these practices to musical constraints and goals is another issue, but one that which needs concerted attention to ensure this project's success.

Situating *Deviate* within a specific musical genre implies that style modelling is a concern, though its intended use in performance environment adds further constraints. A well-known example of musical style modelling is Cope's Experiments in Musical Intelligence (EMI) project, started in 1981 [24]. The data-driven programming techniques used in this project involved "analyzing a database of musical works, and then, using this analysis, replicating new music in some manner appropriate to the user's wishes." [4]. EMI has resulted in the development of compositional programs capable of composing new works in the style of composers including Bach and Mozart. Other style modelling methods have been similarly based on analysis of large bodies of musical data, parsing existing musical texts into a lexicon of phrases or patterns, and selecting progressive musical objects according to context [25]. Modelling using analysis of musical input, rather than a pre-built corpus, has been undertaken by Kippen and Bel [26] in their exploration of improvisations of north Indian tabla players. Each of these data-driven approaches requires analysis of existing works within an established musical tradition. The analysed works must be of a distinct style to ensure consistent results, thus limiting the variety of possible output. As a result, data-driven modelling is more attuned to creating new works according to specific conventions, or replicating the style of individual composers, rather than composing within a broader stylistic framework.

Data-driven style modelling is not immediately suitable for this project, as a core aim of the project at hand is to build a performance environment capable of sustaining a range of musical results. A more informal analytical approach is demonstrated by generative music programs geared more towards popular styles, including Collins' BBCut [27] sample-cutting breakbeat library. Collins describes this approach as "'active style synthesis' rather than 'empirical style modelling'." [27]. Similarly, programs such as Koan (now succeeded by Noatikl) incorporate more general musical rules within a user-defined modular structure to generate note and control data [28]. An approach based on listening analysis and heuristic tests thus focuses more on creative input (either from the developer of the program or the eventual user) in developing new works, rather than rigorous stylistic analysis and recreation. Further examples of generative programs for contemporary or popular music forms include LEMu [29], Bloom [30], and Infno [31].

4.1. From modelling to generative creativity

The link between music-making and generative processes, in a psychological sense, is well established and underlies music performance, improvisation, and composition [32]. The word 'generative', in this context, refers to Chomsky's linguistic theory of grammars and syntactical structures [33] rather than computational music production. Generative frameworks are applicable to musical structures including tonality [34] and form [35]. Given that musical structures can be analysed in generative terms, it follows that such structures may be actively used in the generation of musical material. These generative principles describe broad musical processes, rules and conventions, providing leeway for variation while constraining output within acceptable parameters.

Musical rules and conventions exist at an absolute and empirical level (as used in data-driven style modelling) as well as in a more informal and descriptive capacity. Culturally based concepts such as genre, style, and subculture describe musical and performative practices defined by rules, although these rules may be mutable and not formally set out. A high degree of stylistic formalism exists in contemporary electronic music; Bogdanov identifies 63 varieties of electronica [37], 14 within

Jungle/Drum'N'Bass alone [38]. His description of ragga reads:

> *Ragga jungle is characterized primarily by fast, complex beat patterns, deep, tight bass, and the use of sound system-type MC chanting sampled from old reggae, ragga, and dancehall records. Ragga also makes jungle's connection to African and Caribbean traditional and popular musics most evident, with rhythms recognizably descendent from nyabinghi and calypso-style drumming.* [39].

The above description is naturally based on a categorisation of existing musics rather than an attempt at compositional or computable formalism, but nonetheless demonstrates how existing musical conventions can inform generative structures. The translation from stylistic formalism to strict programmatic formalism in the case of *Deviate* has been undertaken heuristically, based on listening analyses of existing artists and works, as well as analyses of the resulting output.

A more weighty concern relating to algorithmic and generative music performance in general is raised by Lerdahl, who proposes that all artificial compositional grammars, including music composed using algorithmic and generative methods, must be based on a 'listening' grammar in order to contain meaning. Lerdahl suggests a set of 17 constraints relating to musical events, structures, and pitches, and uses them to demonstrate how serialism subverts these constraints and is thus difficult or impossible for a listener to comprehend musically [36]. From this perspective, incorporating generative models derived from existing musics reinforces the aesthetic success of any musical system. Generative musical models based on analysis are thus not confined to instances where modelling and close style synthesis are the goal, but relate to all instances of algorithmic and generative composition.

5. LIVE CODING

Live coding is the foremost approach to laptop performance that addresses the contextual and practical issues cited above, particularly the lack of transparency and the inflexibility of conventional approaches to digital performance. Building algorithmic and generative musical structures from scratch in performance and modifying them in real-time places the performer in an incontrovertible position of responsibility. The projection of the coding screen likewise makes performance practice more transparent. Some drawbacks to the medium of code as an integral part of performance are highlighted by Collins, including "obscurantism and intellectualism" [8], while the complexity of composing code live implies that "in practice most composers would content themselves with modifying pre-tested snippets" [40]. Though live coding laudably addresses many problematic aspects of laptop performance, the intensive process of writing code live impedes the creation of complex musics that adhere to the stylistic characteristics of existing musical genres. The analytical processes and data structures required to output generative music within the constraints of contemporary electronic music are too burdensome to realistically build from scratch in a live performance. Furthermore, coding, live or otherwise, is not a practice that is indigenous to contemporary electronic music production (Aphex Twin and Autechre's occasional use of Supercollider and MaxMSP notwithstanding), and therefore does not logically present itself as a medium for composition, let alone performance. Live coding thus attends admirably to problematic aspects of laptop performance, while introducing new concerns regarding the complexities of using the medium of code for live music creation.

6. STANDARDS FOR DESIGN AND PRACTICE

The issues raised above are the primary matters that need to be answered by *Deviate*'s function in practice. Firstly, the performer must be able to exert sufficient control to direct musical output, but not be obliged to input more control than is feasible. Likewise, the performer needs to fulfill a transparently performative role, rather than a role that is either perfunctory or obscure to the audience. Finally, the system needs to provide a navigable space between the consistency of style modelling and potential for the generation of original works. The following section expands on these assertions, proposing a number of criteria that must be met by *Deviate* and similar systems, and how these criteria may be tested.

Directability: Given that the aim is to construct a performance-oriented environment, there must be sufficient scope for interaction with and influence over output in real-time. The user must be able to direct musical processes to realise his/her own aesthetic goals within broad constraints of musical genre. Any analysis, while necessary to create works conforming to a particular style, must therefore be sufficient to place generated works within genre constraints. Conversely, they should be indefinite enough to allow the creation of new works that are not apparent recreations of a specific style or composer. Directability may be ascertained by composing and performing within diverse musical forms and styles, and according to predetermined aesthetic and musical parameters.

Responsiveness: Aside from the ability to direct musical processes, the environment must be capable of generating material and responding to input with low enough latency to cement the relation between action and result for both performer and audience. The importance of feedback and immediacy in musical performance, and improvisation in particular, is highlighted by Pressing [41], who asserts, "feedback is a vital component in improvisation for it enables error correction and adaptation". A low response time is also a valuable musical criterion and aids in developing skill. Determining responsiveness is a more subjective process, dependent on the performer's impressions and the quality of musical output.

Comprehensibility: The generative structures and processes used, and the controls designed for real-time interaction, need to be intelligible to the performer. This criteria is additionally important given the possible complexity of musical output, and given that the environment is to be used in a performance situation. Developing comprehensibility is handled through interface design, system design, and practice, and through continued refinement of each of these areas in relation to the others. Comprehensibility is, again, a subjective issue dependent on the performer and his/her exposure to and practice with the environment at hand.

Suitability: Musical output needs to be evaluated to ensure that it falls within the constraints of the chosen context and genre. This aspect is contingent on the above criteria of directability, responsiveness, and comprehensibility, and includes aesthetic judgement as well. Apart from being able to produce suitable output, all generated material must be provably consistent to allow its use in performance. This can be objectively informed by formal or informal comparison with existing musical works, as well as by the performer's own judgement.

Reusability: *Deviate* must be sustainable across a range of musical tasks and capable of producing varied output, as it is a genre-based performance environment rather than a tool for realising a specific musical work. A related concern is the ability to extend and expand the environment's abilities. This criterion is dependent on meeting each of the previous criteria, and requires long-term evaluation.

This list is an attempt to objectively determine the standards that must be met for *Deviate* to successfully achieve its goals, and it is constantly reviewed in light of these criteria. Evaluation takes place in terms of both machine functioning and judgements that are aesthetic and subjective in nature, and its function is evolved through a combination of practice, experience, and continual refinement.

7. CONCLUSION

Deviate aims to provide a performance environment for electronic music incorporating note-level improvisation and generative methods. This environment is situated not only within the context of generative music systems, but also within the existing practices of contemporary electronic music. The approach presented here outlines the motivations underlying this research project, as well the contextual issues informing its construction. Given that this undertaking is practical in nature, and has a specific stated outcome, it is essential that the benchmarks for success are clearly elaborated. The criteria above have been devised with regard to current practices in contemporary electronic music and laptop performance, and attend to aesthetic critiques and technical limitations of these practices. These criteria have been useful in designing and refining *Deviate* while attending to larger contextual, musical, and cultural concerns, and it is hoped that future projects bridging generative music and existing musical forms likewise benefit from this approach.

8. REFERENCES

[1] *Max 5* 2008, Cycling '74, San Francisco, CA, USA.

[2] Keith, S 2009, 'Controlling Live Generative Electronic Music with *Deviate*'. *Proceedings of the 2009 International Conference on New Interfaces for Musical Expression* (NIME09), Pittsburgh, PA, USA, 2009 (forthcoming).

[3] Cope, D 2005, *Computer Models of Musical Creativity,* MIT Press, Cambridge, MA, p. vii.

[4] ibid., p. 62.

[5] Wessel, D 2006, 'David Wessel', *Contemporary Music Review*, vol. 25, no. 5/6, pp. 425-8.

[6] Collins, N 2007, 'Live Coding Practice'. *Proceedings of the 2007 International Conference on New Interfaces for Musical Expression* (NIME07), New York NY, USA, 2007 pp. 112-7.

[7] ibid., p. 115.

[8] ibid., p. 112.

[9] Dahlstedt, P 2009, 'Thoughts on Creative Evolution: A Meta-generative Approach to Composition', *Contemporary Music Review*, vol. 28, no. 1, pp. 43-55.

[10] *Ableton Live 7* 2008, Ableton AG, Berlin, Germany.

[11] Hansen, KF 2002, 'The Basics of Scratching', *Journal of New Music Research*, vol. 31, no. 4, pp. 357-365.

[12] Ableton AG 2009, Ableton AG, Berlin, Germany, viewed 9 April 2009 <http://www.ableton.com/live-defy-the-timeline>.

[13] Collins, N 2003, 'Generative Music and Laptop Performance', *Contemporary Music Review*, vol. 22, no. 4, pp. 67-79.

[14] Zicarelli, D 2001, *ICMC 2001 Keynote Speech by David Zicarelli, President, Cycling '74*, ICMC 2001, viewed 9 April 2009, <http://people.finearts.uvic.ca/~aschloss/ICMC_2001/after/keynote.php3>.

[15] Godlovitch, S 1998, *Musical Performance: A Philosophical Study*, Routledge, London, UK, p. 49

[16] ibid., p. 49.

[17] d'Escriván, J 2006, 'To Sing the Body Electric: Instruments and Effort in the Performance of Electronic Music', *Contemporary Music Review*, vol. 25, no. 1, p. 187.

[18] Barthes, R 1977, *Image, Music, Text*, trans. S Heath, Fontana Press/HarperCollins Publishers, Hammersmith, London, p. 181.

[19] ibid., 188.

[20] ibid., 182.

[21] Croft, J 2007, 'Theses on liveness', *Organised Sound*, vol. 12, no. 1, p. 64.

[22] Cascone, K 2002. *Laptop Music - counterfeiting aura in the age of infinite reproduction*, Texts - MicrosoundWiki, viewed 20 April 2007,

<http://interdisciplina.dmethods.com/microsound-wiki/index.php/>.

[23] Davis, E 2002, *Songs in the Key of F12,* Wired Magazine vol. 10 no. 05, May 2002, viewed 13 December 2006, <http://www.wired.com/wired/archive/10.05/laptop.html?pg=1&topic=&topic_set=>.

[24] Cope, D, no date available, *Experiments in Musical Intelligence,* UC Santa Cruz Arts Faculty, viewed 15 March 2009, <http://arts.ucsc.edu/faculty/cope/experiments.htm>.

[25] Dubnov, S, Assayag, G, Lartillot, O, & Bejerano, G 2003, 'Using Machine-Learning Methods for Musical Style Modeling', *IEEE Computers*, vol. 36, no. 10, p. 4.

[26] Kippen, J & Bel, B 1994, 'Computers, Composition and the Challenge of "New Music" in Modern India', Leonardo Music Journal, vol. 4, pp. 79-84.

[27] Collins, N 2001, *Algorithmic Composition Methods for Breakbeat Science*, in Proceedings of Music Without Walls, De Montfort University, June 21-23 2001, ISBN 1857213319, viewed 9 April 2009, <http://www.cogs.susx.ac.uk/users/nc81/research/acmethodsforbbsci.pdf>.

[28] Intermorphic Ltd 2009, *Noatikl,* Intermorphic Ltd, Bucks, UK, viewed 9 April 2009 <http://www.intermorphic.com/tools/noatikl/>.

[29] Wooller, R, Coleman, N, Brown, A, no date available, *LEMu*, Clatterbox, viewed 9 April 2009, <http://www.clatterbox.net.au/instruments/lemu>.

[30] Eno, B & Chilvers, P, 2008, *Bloom*, Opal Limited, viewed 9 April 2009, <http://generativemusic.com/>.

[31] Collins, N, 2007, *Nick Collins - generative music*, Informatics: University of Sussex, viewed 9 April 2009, <http://www.informatics.sussex.ac.uk/users/nc81/music/infno.php>.

[32] Sloboda, J A (ed.) 1988, *Generative processes in music: the psychology of performance, improvisation, and composition*, Oxford University Press, NY, p. vi.

[33] Chomsky, N 1957, *Syntactic structures*, Mouton, The Hague.

[34] Krumhansl, C & Kessler, EJ 1982, 'Tracing the Dynamic Changes in Perceived Tonal Organisation in a Spatial Representation of Keys', *Psychological Review*, vol. 89, no. 4, p. 334-368.

[35] Lerdahl, F 1992, 'Cognitive Constraints on Compositional Systems', *Contemporary Music Review*, vol.6, no. 2, p. 97-121.

[36] ibid., p. 115.

[37] Bogdanov, V (ed.) 2001, *All music guide to electronica: the definitive guide to electronic music*, Backbeat Books, San Francisco CA 94105, pp. vii-xiv.

[38] ibid., p. 637.

[39] ibid., p. 638.

[40] Collins, N, McLean, A, Rohrhuber, J, Ward, A 2003, 'Live coding in laptop performance', *Organised Sound*, vol. 8, no. 3, pp. 321-330.

[41] Pressing, J 1987, 'Improvisation: Methods and Models' in Sloboda, J (ed), *Generative processes in music*, Oxford University Press, pp. 129-178.

ADAPTIVE MUSIC TECHNIQUES

Andrew R. Brown
Creative Industries Faculty,
Queensland University of
Technology

Thorin Kerr
Australasian Cooperative
Research Centre for Interaction
Design

ABSTRACT

There are many interactive media systems, including computer games and media art works, in which it is desirable for music to vary in response to changes in the environment. In this paper we will outline a range of algorithmic techniques that enable music to adapt to such changes, taking into account the need for the music to vary in its expressiveness or mood while remaining coherent and recognisable. We will discuss the approaches which we have arrived at after experience in a range of adaptive music systems over recent years, and draw upon these experiences to inform discussion of relevant considerations and to illustrate the techniques and their effect.

1. INTRODUCTION

Adaptive Music reorganises existing musical material so as to produce new musical experiences based on that material. Adaptive music employs many techniques similar to those used by notational composers to conduct musical variation. Therefore our techniques for adaptation work with familiar and recognisable musical concepts such as pitches and rhythms and operate, primarily, on symbolic representations of music. This is significantly different to Adaptive Audio systems which largely rely on the scheduling of audio recordings and and manipulation of the sound by digital signal processing. An early example of adaptive music systems is the iMuse engine developed in the 1990s for use in video games by LucasArts [1].

Our Adaptive Music techniques have been implemented as algorithms which manipulate symbolic representations of music and respond to instructions in real-time. This gives rise to many possibilities for developing interactive performance environments and collaborative musical experiences involving interaction with symbolic musical material.

Music production environments have typically offered relatively basic interaction with symbolic musical material. MIDI sequencers, for example, typically offer editing facilities on individual notes, or groups of notes in ways which have little to do with actual musical transformations. Notational packages can offer useful manipulation techniques, although the design of these tools revolves around expediting note level composition (to printed score), rather than re-interpreting existing work.

2. BACKGROUND

There are a variety of uses for adaptive music. Our motivation was the provision of interactive music jamming software for musically inexperienced users; typically children or the disabled. Adaptive music techniques are also ideal for artistic installations where the musical material needs to respond to user interaction or to changes in the environment. One of the most ideal uses of adaptive music is in computer games because these are entirely digital environments that are inherently interactive and involve unpredictable real-time state change. The techniques have been used as early at 1987 by Toshio Iwai in the *Otocky* game by Famicom [2]. More recently, the Electronic Arts game *Spore* uses adaptive music extensively, for example varying the instrumentation and rhythmic density of parts in response to game state changes.

There have been a number of approaches to real-time adaptive music using symbolic representations that are worth reviewing by way of contextualising the techniques described in this paper. Interestingly a number of these have been associated with commercial software applications. These include, M and JamFactory [3], DirectMusic Producer [4], Karma [5], MadPlayer [6], Notakle [7], Synfire Pro [8]. We are not including in this review more general-purpose music environments, such as Max/MSP that can be used for creating adaptive music systems, nor are we including generative systems, such as Band-In-A-Box, that tend not to be interactive or adaptive. There are a range of algorithmic techniques used across these systems.

2.1 Templates

The provision of a harmonic template can go a long way to providing a musical backbone for generated music. In a similar way that chord progressions are prescribed in Band-In-A-Box, chord progression templates can be declared in DirectMusic as ChordMaps, and in SynFire harmonic progressions provide a structural basis for surface level changes to phase materials. Structural templates can also be used to dictate large scale form. Often templates are manually described but can also emerge algorithmically as the music progresses as described for harmonic material by Sorensen and Brown [9].

2.2 Abstraction

The uses of abstraction as a method of isolating musical structure has a long history, evident in the analytical musicology of Schenkerian analysis [10] and the Generative Theory of Tonal Music [11]. While most real-time music systems are somewhat limited in their ability to describe, create or even track hierarchical abstractions (with a few exceptions [12]), some adaptive music systems use abstraction as a means of generalisation to inform generated variations in surface level output. For example, Cogitone uses gesture contours to describe musical phrases. Contours can be manipulated by process and by hand. Mappings to and from these contours are not necessarily deterministic and surface details reflect the current context (such as key and metre) when rendered at run time. Another abstraction often used are pitch set and series. For example, M uses lists of pitches, rhythms, and dynamics that can be looped/phased and reordered. Noatikl supports the creation of pitch class sets (scales) and rhythm sets and allows manipulation of materials based on those.

2.3 Recombination

Variation can be provided by selecting from alternative tracks or patterns. This technique is used within DirectMusic's 'style' data type that can contain track alternatives that can be selected based on some state variable or interactive controller. Timbral recombination allows for the re-voicing of musical tracks, and the MadPlayer makes significant use of this such that generated tracks can select from a range of alternatives for each instrument part.

2.4 Transformation

A long-established composition technique is to modify a motif by transposition, arpeggiation, expansion, contraction, inversion, and so on. A number of systems have employed these techniques to varying degrees including DirectMusic, Cogitone and Noatikl. The Karma system makes much of these kinds of transformations, often based around arpeggiation, as a way of elaborating on the musical input (such as a chord or phrase).

2.5 Probability

The most common functions used to control probability are random selection and markov tables. These processes have been part of computer music composition since the computer-assisted composition of the 'Illiac Suite' [13]. David Zicarelli made extensive use of probability in M and of Markov process in JamFactory. While probability is an effective an effective real-time technique, widely exploited in the Koan system for example and still a significant technique in Noatikl, the use of Markov chains requires analysis of data that can limit its responsiveness to change, but in the right context such as used interactively in The Continuator [14], can be quite effective and efficient.

In this paper we demonstrate the techniques which we have found useful in developing adaptive musical systems in performance-oriented environments.

3. MUSICAL ELEMENTS AND MANIPULATION

Music theory identifies numerous musical concepts which arise though common techniques employed by composers to organise musical material. Thus we have concepts which exist beyond individual notes such as melody, harmony, rhythms, or even form. These techniques have evolved as elements of compositional practice, however it is also worthwhile considering other musical elements which aren't expressly identified as compositional techniques.

George Pratt [15] in his work on educating music perception defines a relatively wide range of music elements that include obvious ones, like pitch and rhythm, but also less traditional ones such as spatialisation. Music perception identifies elements of music that are significant in effecting people's responses to music. Some of these are traditionally 'compositional' parameters and other 'performative.' Amongst the most commonly cited are tempo, key, and use of rubato [16].

Our criteria for selecting elements for our adaptive techniques includes that they must be;

- able to be performed in real-time (a logical and computational constraint),
- robust across a range of musical works and styles,
- make a significant perceptual difference to the musical output.

The elements of music that are are available for adaption is also constrained by the parameters in the music representation system being employed.

4. MUSICAL REPRESENTATIONS

Horacio Vaggione has observed that "there is no musical composition process (instrumental, electroacoustic, or otherwise) without representational systems at work [17 p.58]. 'At work' here implies that the representation system in use carries with it, and influences, a mode of musical understanding. Some computer-based music representation systems draw from Common Practice Notation, such as MIDI or Csound scores. Others reflect a signal processing mentality as applied to musical parameters (MAX/MSP, PD). Some other kinds can include graphic scores (UPIC) or piano-roll views. Suffice to say, the representation of musical elements in a software system becomes an important factor to the understanding of music and musical transformations adopted by the software user. In our work we are using a MIDI files of compositional material as the initial material. This means that the immediate representation makes available elements such as event onset and end times, chromatic pitch, velocity, instrument number, and so on.

5. TECHNIQUES FOR MUSICAL ADAPTATION

The techniques described here are those we have found useful over a range of projects, in particular we have implemented them in an algorithmic music application for the XO Laptop designed and distributed by the One Laptop Per Child Program, which we will defer detailed discussion of to a forthcoming publication. Suffice to say that the system relies on MIDI data as the basic material for a musical description that is performed on a sample-based synthesizer.

Based on the material provided in the MIDI file, adaptive algorithms conduct significant symbolic manipulation of the musical data. As will be seen, there is comparatively minimal adaptation of audio synthesis or rendering parameters. By way of example the bass line in figure 1 will be used as a basis for transformation examples in this section.

The transformation of each musical element is discussed below.5.1 Pitch Range

A common and simple technique to affect pitch material is through transposition. Key modulation within a work can be used to produce an increase in tension, or a resolution to the form of a work. However, beyond this scope the effectiveness of transposition proves relatively limited. Furthermore, novice users have difficulty using transposition in musically sensible ways.

Our approach to pitch transformation concentrates instead on the pitch range of a part. We have found that this provides comprehendable and controllable variation. The basic technique is to expand or contract the range covered by the pitches of a part. This is equivalent to the stretching or compression of the pitch contour. Such range variation requires some constraints. A useful pivot pitch, relative to which the range is calculated, which can be a root pitch either at or near the bottom of the original pitch range.

When the range is at its minimal level, the phrase contour is a flat line. All notes in the phrase have the root pitch. There is theoretically no limit to the amount of expansion, but practical limits include the playable range of the instrument, and what expansion amount is musically sensible. The resultant expansion is dependent on the original pitch range of phrase, but in our experience an expansion multiplying the original phrase range between 0 - 2 times is generally sufficient. Figure 2 shows the example with expanded pitch range.

A final constraint is to quantise the resultant pitch to a given pitch set. Effective approaches include the use of a pitch class based on the pitches used in the original phrase, or an appropriate scale or mode for diatonic music.

Percussion instruments respond differently to pitch control. Selecting different 'notes' in a MIDI drum kit, or alternatively, adjusting the audio playback pitch of percussive sounds doesn't provide conventionally musical results. Instead, we have taken the approach of adjusting the selection of drum instruments already in use by the music. An expanded pitch range then takes advantage of the entire drum-kit range. A reduced range will concentrate the drum selection toward a particular drum sound, typically a bass drum.

5.2 Rhythmic Density Thinning

A significant contributor to the 'mood' or 'energy' of the work is the frequency of events. Put more musically, this equates to the number of notes per time period which we call the rhythmic density (not to be confused with the textural density which equates to the number of concurrent independent parts).

Rhythmic density thinning takes the approach that the given piece of music is the 'most dense' version of the music. Assuming the music conforms to regular metric structures, the rhythmic density can be reduced by removing notes that occur in less-dominant metric positions. For example, in simple quadruple time (4/4) we assume a hierarchy dominance beginning with beat1, then 3, then 2 and 4 followed by further subdivisions of the beat which follow a similar hierarchy.

Events (notes) are then filtered based on their metric position, and where that position sits along a continuum from least to most dominant. Figure 3

Figure 1. The original musical phrase.

Figure 2. The musical phrase with expanded pitch range.

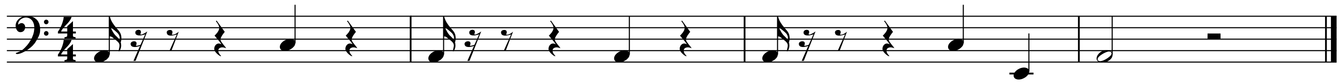


Figure 3. The musical phrase with reduced rhythmic density.

shows an the example phrase with a reduced rhythmic density.

While this technique in its raw form is effective, we have found that the musical change from stage to stage along the filtering continuum can be too sudden. To compensate for this effect we soften the change from stage to stage by implementing a 'porous' filter that 'lets through' a number of notes that would otherwise be filtered. Another side effect of this filtering is that note durations need to be adjusted (lengthened) to compensate for the 'space' created by deleted notes. This duration operation occurs independently of other processes which affect the duration of notes (see Articulation below).

5.3 Articulation

In musical performance the articulation of a note can mean a number of things including tuning, attack, dynamic level and so on. In this situation we use articulation in a more limited sense to simply mean the performed duration of each note. Whether a phrase is played legato (sustained notes) or staccato (abbreviated notes) can make quite a difference to the 'mood' of the music. The algorithmic aspect of this is quite trivial, to scale the note duration by some articulation factor. This factor when set at 1.0 will playback the note as 'recorded' in the MIDI file (or as adjusted by some other algorithmic process such as rhythmic density), a factor near zero will produce the shortest note length while factors greater than 1.0 will extend the note. Given that most human performances have note lengths around 80% of the written duration, a factor of 1.2 or more will start to produce overlapping notes, creating an effect not unlike holding the sustain pedal down on a piano.

5.4 Dynamic Level

The dynamic level in musical performance is largely a matter of loudness, but is also associated with timbre, attack and other factors that correspond with more energetic playing of an instrument. In implementing this technique we confine ourselves to a simplistic notion of dynamic as loudness. Of course given that we use MIDI velocity as a carrier of the dynamic value it is possible to use a synthesizer which responds with more sophistication than simple volume adjustment. Similarly, all scaling in our examples are linear and we leave it to the synthesizer to convert these into the logarithmic loudness curve typical of musical instruments.

We employ two dynamic controls, ‘overall’ and ‘pulse dynamic’ variations: Our overall adaptive dynamics simply scale the existing dynamic value of each note from the MIDI file. Given that most performance in MIDI files average at around 70-80% of the available dynamic range some capacity to increase dynamic level can be expected within the limits of legal parameter ranges

Pulse-based dynamic change mimics the way performers add emphasis to certain beats in regular metrical music. We use a simple periodic function (summed cosines) to imitate this emphasis. The frequency of the periodic function needs to match the pulse or beat duration of the current metre. Controlling this function means controlling the amount of emphasis given to accented beats. Increasing emphasis, increases accents on certain beats. Note that a setting specifying no emphasis will provide no additional emphasis on beats, however accents recorded in the original performance are retained.

5.5 Tempo

The speed of music has been shown to be a significant contributor to musical affect. Adjusting the tempo involves a relatively trivial change in playback speed. As with the dynamic level, research has shown that quasi-periodic variations in tempo are common in human musical performances [18, 19]. To imitate this and provide adaptive control over it we implement a subtle periodic tempo change, again using summed cosine functions, that approximates this effect. Here the style of the music is important: For example, the effect is quite effective when applied to Western music of the Romantic period which utilises rubato quite liberally. It is less effective when applied to modern or electronic music styles with rigid tempos.

5.6 Timbre

Timbre manipulation is not the focus of adaptive music techniques, however some simple timbre manipulation is provided. In particular, a resonant filter is provided which is mostly effective for the manipulation of electronic styles of music. Filter cutoff is the parameter adjustable by the user. A useful amount of resonance can be calculated as a function of cutoff, peaking at a mid range of between 700 to 1000 hz, and tapering at audible extremes. If a resonant filter is not suitable for the style of music, or the instrument, then the quality of the sound can be changed by selecting an alternate instrumentation for the part.

There are audio examples of all these techniques available online at http://www.explodingart.com/acmc2009/.

6. INTERACTION, CONTROL AND MAPPING

There are a variety of ways to represent changes in musical parameters through an interface. On the XO Laptop users control the amount of musical transformation by controlling the spacial position of an instrument. Movement along each axis adjusts a particular group of musical transformations. Here users understand the transformations in terms of a 'stylistic' change to the music. A similar approach is provided with the Jam2Jam network jamming software on the Macintosh. In this case users are given access to the specific musical parameters along each axis. With this model, users gain an understanding of each individual transformation, and how each transformation contributes to overall stylistic qualities. Interaction can also be collaborative through the use of supplementary controllers such as MIDI controllers or iphone applications, or with networking features available in the software.

7. TESTING AND EVALUATION

These algorithmic processes have been developed through field trials with several systems, in particular four iterations of the Jam2jam network jamming software that have been tested in numerous locations around the world [20, 21]. Observations provided by participants in the trials has guided the direction of development of the algorithms and the mechanisms for interaction with the algorithms. Of key importance was gaining an understanding of the relationship between the users experience, the mechanism for interaction and the effectiveness of the algorithms. For example, through trials we found that, although the interface was easy to interact with, this did not always translate into understanding musical concepts in play, as illustrated through this observation from a teacher:

> "when being used with students, some training needs to occur. Although the software is intended to be experimented with, students find it confusing to just pick up and use. They need to have it demonstrated so they know what it's possibilities are" (Personal correspondence, 2009).

In observations like this, it is unclear whether modifying the algorithms or the interface will improve the experience. In fact, the solution to this feedback, and others like it, was to change the representation of a musical parameter. Rather than simply providing an interface with controls to a 'black box' musical application, a user could visually see settings on the screen which corresponded with the actual musical parameter settings.

As well as observations of users interactions with these systems that have informed the effectiveness and significance of these changes, the various jam2jam iterations have involved three re-implementations using different programming languages (Java, Scheme, and Python) and music systems (jMusic, Impromptu, and Csound). This process has also refined the way in which musical adaptations can be most efficiently be represented as algorithms.

8. SUMMARY

Adaptive music techniques offer a means to enhance the expressive terrain of existing music by translating musical concepts into performable parameters. In the course of this research we have found that the representation and effectiveness with which to interact with musical concepts is crucial in enabling creative engagement with those concepts. This paper has presented some of the techniques, and refinements of those techniques, which have proved effective.

Audio examples of all the techniques described in this paper are available online:

http://www.explodingart.com/acmc2009/

9. ACKNOWLEDGMENT

This research was supported by the Australasian CRC for Interaction Design, which is established and supported under the Australian Government's Cooperative Research Centres Program.

10. REFERENCES

[1] Land, M. and P. McConnell, *iMUSE*. 1994: Berkeley.

[2] Collins, K., *An Introduction to Procedural Music in Video Games*. Contemporary Music Review, 2009. **28**(1): p. 5-15.

[3] Zicarelli, D., *M and Jam Factory*. Computer Music Journal, 1987. **11**(4): p. 13-29.

[4] Fay, T., *DirectMusic Producer*. 1998, Microsoft Corporation: Redmond.

[5] Kay, S., *Karma*. 2001, Korg USA: Melville, NY.

[6] MadWaves, *MadPlayer*. 2004, MadWaves: London.

[7] Cole, T. and P. Cole, *Noatikl*. 2008, Intermorphic: London.

[8] Schnoor, A., *Synfire Pro*. 2008, Cognitone: Hamburg.

[9] Sorensen, A. and A.R. Brown, *A Computational Model For The Generation Of Orchestral Music In The Germanic Symphonic Tradition: A progress report*, in *Sound : Space - The Australasian Computer Music Conference*, S. Wilkie and A. Hood, Editors. 2008, ACMA: Sydney. p. 78-84.

[10] Schenker, H., *Harmony*. 1980, Chicago: University Of Chicago Press.

[11] Lerdahl, F. and R. Jackendoff, *A Generative Theory of Tonal Music*. 1983, Cambridge, MA: MIT Press.

[12] Gifford, T. and A.R. Brown, *Automated Improvisation*, in *Australasian Computer Music Conference*. 2009, ACMA: Brisbane. p. (in this volume).

[13] Hiller, L.A. and L.M. Isaacson, *Musical Composition with a High-Speed Digital Computer*. Journal of the Audio Engineering Society, 1958. **6**(3): p. 154-160.

[14] Pachet, F., *The Continuator: Musical Interaction with Style*, in *International Computer Music Conference*. 2002, ICMA: Göteborg, Sweden. p. 211 - 218.

[15] Pratt, G., *Aural Awareness: Principles and Practice*. 1990, Buckingham: Open University Press.

[16] Livingstone, S.R., et al., *Controlling Musical Emotionality: An Affective Computational Architecture for Influencing Musical Emotions*. Digital Creativity, 2007. **18**(1): p. 43-54.

[17] Vaggione, H., *Some Ontological Remarks about Music Composition Processes*. Computer Music Journal, 2001. **25**(1): p. 54-61.

[18] Juslin, P.N. and J.A. Sloboda, *Music and Emotion: Theory and research*. 2001, Oxford: Oxford University Press.

[19] Friberg, A., *pDM: An expressive sequencer with real-time control of the KTH music-performance rules*. Computer Music Journal, 2006. **30**: p. 37-48.

[20] Dillon, S., et al., *Communities of sound: generative music making and virtual ensembles*. International Journal of Community Music, 2008.

[21] Brown, A.R. and S. Dillon, *Networked Improvisational Musical Environments: Learning through online collaborative music making*, in *Music Education with Digital Technology*, J. Finney and P. Burnard, Editors. 2007, Continuum: London. p. 96-106.

GESTURE AND MUSICAL EXPRESSION ENTAILMENT IN A LIVE CODING CONTEXT

Alistair Riddell
Photography and Media Arts
ANU School of Art
The Australian National University
Canberra, ACT
alistair.riddell@anu.edu.au

ABSTRACT

This paper considers issues and implications of gesture and musical expression for live coding performance. In reviewing the broader relevant literature and its interpretation in digital music research, it seemed logical to reflect on how musical gesture and expression might evolve in such generative sound practices. Indirectly this research raises the question of what live coding is capable of facilitating in the context of realtime performance and how and why live coding languages could evolve in this respect.

1. INTRODUCTION

The practice of live coding reflects a curious disjunction in the evolution of computer music. Although it might seem retrogressive, the practice does reflect maturity, re-asserting the primacy of language and extending that into the realm of extemporized dialogue between performer and instrument as public performance. This creative practice runs parallel to the general trend of explicit functionality through Graphical User Interfaces or more explicitly, patching. The debate over which is a more effective engagement in creative practice continues with important arguments from both sides[1] if the wider context of computer music activity is considered. Let's say, however, as a matter of closure on this, that for some the idea of live coding is profoundly intriguing.

Several important texts, Nilson [12], Blackwell and Collins [3] and Brown and Sorensen [6] discuss live coding from the central positions of programming languages, mathematical formalisms, generative systems, historical evolution, performance practice, relation to the traditional instrumental skills and wider implications for computer music. From these seminal texts, the prospects for live coding can be extrapolated to a number of important creative prospects. Perhaps the most curious of these is a view articulated by Whitelaw [22] where live coding is essentially data manipulation in realtime. Whitelaw states about live coding:

> "Whatever else it says, it also says, "watch what I do with this data." It displays a data literacy, an ability to acquire, munge, filter, process, map and render. Since it's primarily operating as art, rather than functional visualization / sonification, it also demonstrates a process of translating or mediating between these domains."

While it is possible that live coding could singularly embrace such a data aesthetic (infosthetics) practice, of interest here is the extent to which the practice inspires lateral thinking and expectations of new creative directions.

Returning to the musical context, Nilson [12] provides one of the most illuminating and personalized accounts of what it is to be a live coder and the parallels with traditional instrumental practice. In fact, much of the discourse found in the three seminal texts mentioned earlier, concern the pragmatics and public reception of live coding as a musically creative activity.

Although a relatively new practice[2], live coding does draw upon a wealth of knowledge in language development and major advances in computational hardware. Consequently, it does seem a logical development in the evolution of computer music. Indeed, it appears veritably fecund with possibilities beginning with the number of software applications that can be used and those that have been developed explicitly for that purpose. Apart from addressing existing and particular issues surrounding interactive music, live coding arouses curiosity in regard to what further creative possibilities await to be discovered. An example of this is the collaborative practices of *PLOrk* [14] and *aa-cell* [18] that suggest a more sophisticated outcome, intensification and contemporaneity in the "live" dynamic.

It is obvious that live coding concerns circumscribe a particular view of the mind/body creative interaction, which is somewhat unprecedented. Nilson [12] observes, "It is helpful to first disassociate control and physicality." This has huge ramifications even within the electronic music fraternity with those who believe the human body crucial to all musical expression. Yet

[1] A Coding vs. Patching thread on the SuperCollider Users mailing list from 24-04-09 debated issues including computational and cognitive efficiency, process comprehension and pedagogy. The discussion petered out with refutations from both perspectives that advanced neither approach over the other.

[2] See toplap.org/index.php/HistoricalPerformances for an evolutionary overview.

creative production and indeed, musical expression also require levels of abstraction that “don’t have an immediate physical analog” Nilson [12]. It is clear that greater abstraction–the construction of complex sound events–is a key strength of live coding. In this respect the public experience has been augmented with a simultaneous visual presentation of the coding process even though as Sorensen and Brown [18] point out, “However, even with a strongly technical audience, complete comprehension of the generative ramifications of the source code being run during a performance is challenging”. Nevertheless, the evidentiary spectacle does serve a purpose as confirmation of interaction and causal relations. While there is clearly something inscrutable about dynamic code presentation, for some of the audience it can become quite engaging as the performance evolves and the musical result becomes compelling.

2. WHY EXPRESSION

What is absent in certain practices of electronic music is evidence of a physical relation between the body and sound production as is widely understood in traditional musical practice. There are two approaches to looking at this. Across electronic music genres, it is either not important or not relevant to the nature and style of music or it is. So not all electronic music benefits from or is enhanced by a correlation between human physical agency and the sound. But the instrumental paradigm is difficult to ignore. Watching someone playing piano is to observe a physical engagement in the production of sound, which we identify as “effort” and in a more refined state becomes externalized emotion mapped to the sound. Sometimes it is genuine and sometimes affectation but the behaviour is considered mandatory for most performance to achieve expressivity. The question of “Liveness” as discussed by Croft [9] reflects on how electronic music evolved with the absence of “body presence” in many of its contemporary genres since the middle of the Twentieth century. It is worth noting that this status is probably undergoing a state of reversal as various performers are currently returning to some type of human performance or instrumental engagement with computer technology. Having an identity as a performer maybe putting some balance back into the electronic music studio production context. Not to mention simply reinvigorating live electronic music even if live coding is seen as a controversial practice.

A corollary to this is whether it is necessary for musical expression to be based on traditional instrumental practices at all. In recorded music, without visual referents, it is an allusion. The listener must interpret and respond to the nature of the sounds. So is interpretation and appreciation of expression possible when there is a human presence but no visual correlation of human action? Laptop performance is a classic example of this, where the process of sound production and physical interaction appear to have no connection with human effort. Perhaps it is more appropriate to say that little attempt is made to establish a connection. In contrast, live coding takes a different view of this issue in a performance context and thus opens discussion on the extent to which creative effort can be manifest.

Thus intrigued by the prospects of live coding, the idea for this paper arose from discussions about how certain sound events could be given more expressive characteristics during a live coding improvisation. Now, it should be understood that the approach to gesture generation under consideration may not be appropriate for all musical forms or for all musical material being constructed within a given performance. Specifically, it is directed towards musical material deemed to be of an expressive nature and understood to require such nuance. For example in a simple case, melodic material without such tweaking might simply sound mechanical and characterless. A point of reference for this research was performance characteristics in certain Jazz styles and idioms. On reflection, the prospect of live coding practice accommodating and attaining such a level of expressivity and individualism is too compelling to pass without comment.

Given that any kind of expressive superimposition on musical material implies an additional creative burden for the performer, a central objective of this research is to discuss the value of including such performance overhead cognizant of the demands of creating new sound structures. This would seem a profound consideration with even a partially generative approach to live performance. In challenging live coding performances, effective expressive coding might be too demanding to implement in a given time.

For a live coding performance in which sound material is defined numerically or by predetermined structures or generative processes before execution, it begs the question as to how one would know firstly, if the material needed such nuance and secondly, what such expressive control would be? If we think in classical terms of a melody for example, typically it is a more or less fixed form, can be understood during rehearsal and in the composition usually repeated at various times. It is therefore possible to focus on an expressive articulation of the melody. With live coding it is perhaps possible also if there is a reasonable window of opportunity. So it is conceivable, given enough practical experience, that an understanding of how to shape particular sound events could become simply another live coding skill. After all, that is exactly how it works with traditional instrumental music.

3. GESTURE AND MUSICAL EXPRESSION FROM THE PHYSICAL PERSPECTIVE

A review of some of the literature surrounding gesture and musical expression reveals the extent and diversity to which the subject has been researched in recent years. A sample review covers analysis, trends and synthesis [1, 2, 7], mapping and controllers [4, 10, 11, 13, 15, 20], and beyond sound [10, 15, 21]. This is to gaze into only a fraction of the research. Many researchers consider the most effective solution to be based on external controllers with a fundamental understanding derived from an analysis of traditional musical instruments characteristics [8, 17, 19]. This is a logical starting point, as external interface devices have a direct relation to musical instruments and an implied connection with musical expression at the physical level. In this respect, models of gesture and musical expression can be analyzed and understood from a wide variety of traditional musical instruments and performance practices. Of interest at this point is, whether the expressivity of such controllers is largely predicated on existing instrumental skill and whether that impacts on the prospect of electronic music developing unique forms of expression. There are also controllers that have no connection with traditional instruments but explore the idea of gesture from other aspects of human movement and language. Often the extent of such research is too rarified to have a significant impact on the electronic music community.

The question that arises here is are controller paradigms constraining the development of contemporary musical expression in the electronic domain? Must the future of musical expression be conditional on human physical gesture as the context for musical expression? This question permeates the intention of this paper, as clearly this would be somewhat limiting. The concept of musical expression, in generative music, may need to be appreciated through a more abstract understanding of the nature of control as experienced through sound events uniquely improvised by the live coder.

Of particular interest regarding gesture analysis from an instrumental perspective, is the research of Levitin et al [10] and Rovan et al [15]. Levitin and colleague's analysis of a musical event provides a fundamental framework that is almost universally applicable. In short, they are considering the control of a tone in a monophonic context, the *attack, steady-state* and *decay* or beginning, middle and end form as axiomatic to how we perceive musical events, whether as one note/sound or a melody. The mapping strategy used by the authors is "the linking or correspondence between gestures or control parameters… and sound generation or synthesis parameters." These "should exploit some intrinsic property of the musician's *cognitive map* so that a gesture or moment in the physical domain is tightly coupled – in a non-arbitrary way – with the intention of the musician."

The research of Rovan et al, discusses gestural mapping strategies with a specific focus towards a practical outcome employing the Yamaha WX7 wind controller interfaced with IRCAM's FTS software running an additive synthesis engine. Of interest here is their discussion of a 2 dimensional expressive timbral subspace mapping. This is defined as four quadrants formed from the intersection of points where X is pitch and covers a 2 octave range (F3, F4 and F4) and dynamic levels (pp, mf ff). While they point out that the approach has existing parallels with sample synthesizers, they observe, "By considering the additive method, we consider interpolation not between actual sounds but between models, and thus the issue of modeling is central to our work."

In both research cases, implementations use an external controller. So although the data from the controllers is digital, the important aspect of timing in musical expression can be derived from the performer, as well as visual performance cues for the audience. If computer music controllers are considered in a more experimental context, such as those described by Cook [8], musical expression could take some interesting new directions. While Cook's constructions are predicated on some kind of musical instrument other systems like that by Overholt [13] and Van Nort [19] are based on sophisticated and novel physical interaction that have no immediate association with traditional instrumental practice. Yet the aspirations of use are quite specific, if not ambitious, Overholt writes:

> "Instead of requiring excessive musical knowledge and physical precision, performers will be able to think about music in terms of emotional outputs and the gestures that feel natural to express them. This should let the musician to go much more directly from a musical idea or feeling to sound, thinking more about qualitative musical issues than technique or physical manipulations."

Exactly what music the author is referring to cannot be appreciated from the paper but can be deduced to have at least reached initial expectations.

Research into musical expression and gesture based around controller development would appear to be universally accepted and in fact encouraged at an experimental level. Sometimes it seems with indifference to Cook's [8] *Principles for Designing Computer Music Controllers*, for example, number 5, "Make a piece, not an instrument or controller" which seems an important and astute observation for experimental research with creative aspirations. Reflecting on that principle a bit further, the implication is that the construction of a controller should not be based purely on the nature of the controller itself, without a vision of what creative outcome could be achieved. Yet who would know what that might be? The unexpected is fundamental to the prospects of experimentation.

Development of physical controllers has the initial and time consuming stage of constructing the device itself before progressing to issues of data output, mapping strategies, sound production and composition/performance. In this respect, approaches to gesture and musical expression are implicitly considered in physical terms from the outset.

Another view of gesture is from the musical perspective. Arfib et al [2] comment, "Using musical gesture relates to the musical meaning given by the sound performer, meaning which we can extract from the signal, with long integration time, for example transition type–portamento, legato, pizzicato–and modulations–vibrato, roughness." So what are the prospects for gesture and musical expression when a physical device is not implicit? Possibly good but ambivalent, consider Arfib et al again, "Note the effect of physical gesture appears in musical gesture, even though we cannot always notice it." It should further be noted that this research into physical gesture and its correlation with musical gesture was intended "to permit the design and development of human-computer interfaces better adapted to interpretation and improvisation." Although in this case there are physical controllers involved at the research front end, why should musical gesture research not be undertaken independent of such a starting point?

4. LANGUAGE AS INTERFACE TO EXPRESSION

How might sound applications that employ a programming language as a means of performance, negotiate gesture and musical expression? Throughout this text what is meant by gesture or musical expression has been left solely to the readers experience and understanding. It seems fair to say that most informed views begin with experiences from instrumental music and pedagogy. For example, the term, 'cantabile' can be understood to encapsulate a very specific relation between gesture, as originating through physical action and musical expression as an invocation of emotion in classical music. Its application and relevance to the evolving world of computer music might seem limited and perhaps anachronistic but the ability to effectively execute that well understood musical term in a live computer music context raises some interesting questions.

While live coding is an empowering means of production, it brings with it a formalism whose origins are not from conventional music practice. It is therefore, a unique form of cognitive engagement at a creative level but issues of extensibility and sophistication are not yet widely understood. Knowledge of live coding practices that lead to more sophisticated and nuanced performance will coalesce over time and through wider use and experience of creative outcomes. If multiple applications exist for live coding performance, what points of commonality do they share? This question has yet to be systematically addressed but has beginnings in discussions by those who are active practitioners. Sorensen and Brown [18] observe:

> "Our approach revolves around setting up generative processes, and the dynamic nature of live coding allows the performer to direct these processes. Live programmers not only write the code used to generate the music, they also constantly change and modify the behavior of that code dynamically throughout the performance. In this way the live programmer controls higher level structure, directing processes like a conductor directs an ensemble."

No matter what languages or formalism are used for live coding, there will be a tendency towards refinement that facilitates the construction of sophisticated sound events and hopefully, their expressive articulation. One might expect that this would lead to defining means of gesture and musical expressivity commensurate with the nature of the mode of engagement.

5. PROPOSITIONS FOR CODING EXPRESSIVITY AS PERFORMANCE

After negotiating the wider context of musical expression in the computer/electronic music context and considering pertinent aspects of live coding, the following propositions are intended to address issues specific to defining what musical expression might entail in this context. The relative importance of these propositions is a matter of practical consideration and engagement, and not of overall concern here. There are clearly points that overlap and converge and are bilaterally influential, but again that is more pertinent to later implementation discussions. It also has to be acknowledged that these points need wider discussion and explication.

5.1. Musical Expression as a Universal Condition

Understanding gesture and musical expressivity depends on experiencing and acknowledging the emotional effect of musical structures. These can range from a single sound of arbitrary duration to complex sound aggregations. In the case where sounds are uniquely electronic in nature, the concept of musical expression is fundamentally abstract and may require a specialized interpretive approach. But recognition in the listening experience that sound contains something of the humanity of the performer (in this case a cognitive state) one that it is transfigured, is something to aspire to.

In much contemporary electronic music, expression that is traditionally understood may not be relevant, applicable or achievable. This condition should open up the possibility of a meta-level condition arising as a maturing consequence of the practice.

5.2. A Language Framework for Expressivity

Live coding is a trajectory from the submission of a symbolic notation to interpretation to sound in a spontaneous act. An excellent overview of necessary conditions for a live coding language can be found in Blackwell and Collins [3], Brown [5,6] and Wang [21]. Within the nature of the symbols, syntax, grammar and overall language implementation lie the potential for defining the production of sound deemed to be infused with expression.

Therefore, a language specification that permits development of constructs for gesture and musical expression is a reflection of the sophistication of that language. They also need to reflect expectations of the effect on the sounds to which they are applied. This is challenging because they could be personal and additionally inscrutable. Experience becomes the only means of predicting a reception of success.

5.3. Pre-configuration of Expressive Structures

The pre-configuration of functions and structures for a given performance is likely to be a necessary and evolving condition to the production of more sophisticated and subtle performances. This suggests that some initial thought as to what will be performed and how, will require follow up coding in the form of specific macro structures. A language of any sophistication would allow and facilitate this.

Further to this point, the nature of expressive structures are likely to be defined in the context of control data influencing audio signal amplitude, frequency, event timing and diffusion. This is not an unfamiliar activity to electronic musicians. However, to consciously consider performance as control and at a symbolic level, independent of sound production, raises some interesting thoughts, particularly in the way one might have learnt the language in the first place.

5.4. Expression in a Visual Form

An integral part of live coding is the visual presentation of the code itself. The dynamics of this evidence is remarkably revealing of the performer as much as it is of the sound. The idea of presenting in code, expression being applied to the sound, is a seductive extension to the performance.

Evidence of expression could be deployed to another representative level within the visual context. Perhaps in a distinct form and location to the code itself. This could however, be too distracting or confusing. Alternatively, integrating a graphic representation under the code might be more subliminal. This representation could fade over time and may only be present to indicate a shift in the performer's creative focus. This may complicate the relation with the code, which may not be responsible for that expression. However, it might work well with re-entrant non-linear code blocks.

This visual form has been in part realized already. Andrew Sorensen, in performance in 2008[3], employed descending coloured patterns synchronized with the sound to some effect. Extending this to reflect expressive intentions in performance therefore might not be that difficult to implement and a first step towards the visualization of expressive intention.

5.5. An Idiomatic Terminology

The historical and musical implications of gesture are undeniably influential in all musical activity but in the context of live coding the term's physical connotations are clearly less relevant and accessible. The term, 'musical expression' seems more inclusive and flexible in the evolution of electronic sound. It is conceivable that the practice and results of a meta-level shaping of sound in live performance might be understood through the establishment of a terminology more applicable to the genre. Such terminology might be in the form of code, and would significantly distinguish it from the terminology of traditional musical practice but it may lack universal coherence.

5.6. Prospects for a Unique Genre

Ideally, live coding would facilitate a particular style of music over time. It is conceivable that it would have less of an emphasis on sound and technology, and more on contemporary idioms of composition, improvisation and general music making. However, this is likely to depend on the performer, the language they use and the context in which they perform.

Another critical matter concerns pedagogy and the cultivation of a community of users. In the current climate of electronic music practice, a clear incentive to embark upon the learning curve of live coding would be a clear potential for creative individualism beyond what is possible with the common performance applications. This remains a matter of reaching a critical point in public musical awareness of what live coding can express.

6. CONCLUSION

This paper was written with a personal vision in mind of what musical expression, in the context of live coding, might entail. It had its origins in preliminary thoughts around the idea of a physical, indeed, tangible surface that displayed contours, which would be mapped to structures controlling expression. However, thinking this through resulted in a return to consideration of the primary mode of engagement, the language itself. This paper has not had a specific language implementation underpinning it rather it has taking a deliberately abstract and philosophical view. Consequently, as preliminary research, it is intended to inform approaches to later technical developments.

In the course of writing this paper I have personally reflected on the cognitive relation between programming and the ability to produce and appreciate sound events that achieve a sense of humanness or expressive agency through abstraction set in varying degrees of immediacy. Interestingly, a tradition instrumentalist must constantly make a sound while simultaneously listening and adjusting in order to appreciate its expressive effect. A live coder, while also being able to alter evolving sound, has opportunities for reflection while their generative constructs unfold in time. In this unique boundary world between performance and composition, perhaps the meaning of musical expression will be in the form of a deep awareness of and creativity with complex event sequences, a fluidity of response and the imposition of a unique identity in creative outcomes. In essence, those indefinable properties: style, experience and character.

[3] *Transmissions in Sound.* ANU. October 15, 2008.

The only effective way to reveal such aspirations is through reflective and sustained performance practice. This practice would inevitably lead to an intimate understanding of engagement, and personal extensions and refinements to the formal language. Whilst a fascinating prospect for this distinctly cognitive practice, inscrutability may become idiosyncrasy, and ultimately through a cognoscenti, acquire public acceptance and recognition. Although live coding seems counter intuitive, it uniquely combines the reflectiveness of composition with the spontaneity of improvisation. Such a singular and evolving creativity at least parallels and rivals the intellectual dimensions of traditional instrumental practice.

7. REFERENCES

[1] Arfib, D. and L. C Kessous. "Gestural Control of Sound Synthesis and Processing Algorithms", in Lecture Notes In Computer Science; Vol. 2298. Springer-Verlag. Berlin. 2001.

[2] Arfib, D. and J. M. Couturier, L. C Kessous, V. Verfaille. "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces", in *Organised Sound*. 7(2). 2002.

[3] Blackwell, A and N. Collins. "The Programming Language as a Musical Instrument". In *Proceedings of the Psychology of Programming Interest Group* (PPIG05), 2005.

[4] Bongers, B. "Physical Interfaces in the Electronic Arts", in *Trends in Gestural Control of Music".* IRCAM book and CDROM.

[5] Brown, A. R. "Code Jamming". M/C Journal 9, 6, 2006. http://journal.media-culture.org.au/0612/03-brown.php

[6] Brown, A. and A. Sorensen. "Interacting with Generative Music through Live Coding", in *Contemporary Music Review.* 28(1). 2009.

[7] Cirotteau. D, G. De Poli, L. Mion, A. Vidolin, and P. Zanon. "Recognition of Musical Gestures in Known Pieces and in Improvisations", in *Gesture-Based Communication in Human-Computer Interaction*. Springer-Verlag, Berlin. 2004.

[8] Cook, P. "Principles for Designing Computer Music Controllers", in *Proceedings of the 2001 conference on New Interfaces for Musical Expression*. 2001.

[9] Croft, J. "Thesis on Liveness", in *Organized Sound.* 12(1), Cambridge University Press. 2007.

[10] Levitin, D. and S. McAdams, R. Adams, "Control parameters for musical instruments: a foundation for new mappings of gesture to sound", in *Organized Sound.* 7(2). Cambridge University Press. 2002.

[11] Kim, J. H. and U. Seifert. "Embodiment and Agency: Towards and Aesthetics of Interactive Performativity", in *Proceedings of the 4th Sound and Music Computing Conference.* Lefkada, Greece. 2007.

[12] Nilson, C. "Live Coding Practice", in *Proceedings of the 2007 Conference on New Interfaces for Musical Expression*. New York. 2007.

[13] Overholt, D. "The MATRIX: A New Musical Instrument for Interactive Performance", in *Proceedings of the International Computer Music Conference*. Havana, Cuba. 2001.

[14] Trueman, D. and P. Cook, S. Smallwood, G. Wang. "PLOrk: The Princeton Laptop Orchestra, Year 1", in *Proceedings of the International Computer Music Conference.* 2006.

[15] Rovan, J. and M. Wanderley, S. Dubnov, P. Depalle, "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance", in *Proceedings of Kansei - The Technology of Emotion Workshop.* 1997.

[16] Schacher, J. "Gesture Control of Sounds in 3D Space", in *Proceedings of the Seventh International Conference on New Interfaces for Musical Expression*. New York. 2007.

[17] Sha, X. Wei, and G. Iachello, S. Dow, Y. Serita, T. St Julien, J. Fistre. "Continuous Sensing of Gesture for Control of Audio-Visual Media", in *Proceedings of the Seventh IEEE International Symposium on Wearable Computers*. 2003.

[18] Sorensen, A. and A. Brown. "aa-cell in Practice: An Approach to Musical Live Coding", in *Proceedings of the International Computer Music Conference.* 2007.

[19] Van Nort, D., D. Gauthier, S. Xin Wei and M. M. Wanderley. "Extraction of Gestural Meaning from a Fabric-based Instrument", in *Proceedings of the 2007 International Computer Music Conference,* Copenhagen, Denmark. 2007.

[20] Wanderly, M. "Gestural Control of Music", in *Trends in Gestural Control of Music*. http://recherche.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/kassel.pdf. 2000.

[21] Wang, G. and P. Cook. "ChucK: A Concurrent, On-the-fly, Audio Programming Language", in *Proceedings of the 2003 International Compute Music Conference.* 2007.

[22] Whitelaw, M. "Data, Code & Performance". *The Teeming Void*, 21, September 2006. teemingvoid.blogspot.com/2006/09/data-code-performance.html

LINKING ANALYSIS AND CREATION: AN IMPROVISED STUDY

Matthew Hill
School of Creative Arts
James Cook University
Townsville, Qld, Australia
mattsound22@gmail.com

ABSTRACT

This paper discusses the application of a compositional process linked to the analysis of specific improvised musical works incorporating new technological elements. A parametric analysis template is utilized to examine nine works and insights from the analyses are applied to the creation of an improvised study. The paper draws from doctoral research where an analysis/creation model was developed and applied across a range of musical genres including electronic, electronic dance music, improvised and rock.

INTRODUCTION

Hill (2007) reviewed a range of existing analytical methods and identified the tendency for analysis to be directed towards musicological, rather than music making, ends. Furthermore, many existing analytical methods are inadequate in dealing with the processes and outputs of new and emerging music technologies. Existing analytical methods focusing on electroacoustic, electronic dance music, jazz and popular music provide useful analytical tools. Of particular importance is Middleton's (2000) notion of a *participant analyst*. It is the premise of this research that extending the role of Middleton's *participant analyst* beyond that of the musicological domain, to encompass a *participant analyst/creator*, has the potential to facilitate a more comprehensive investigation into the compositional/musical creation domain.

ANALYTICAL METHOD

The text-based, parametric approach incorporates observations made by the analyst (mostly on an aural basis) alongside insights from the literature related to particular parameters. The first section of the analysis template (see Table 1) addresses the shaping factors in the creation of a musical work. According to numerous authors (e.g., Brown, 1997; Ferrara, 1984; Hubbs, 2000; Tagg, 2000), the consideration of a broad range of extramusical parameters is necessary for any comprehensive analysis this research incorporates this principle. The consideration of shaping factors also enables an investigation of positionality and individual agency in relation to technology. The second section of the template addresses the inputs of a particular work and includes a range of traditional parameters (e.g., pulse, metre, dynamics, pitch selection, texture, etc.) in addition to parameters more suited to electronic and electroacoustic works such as, sound objects, spatial elements, and programmatic association. The range of parameters is intended to include those that impact on the creative process.

Shaping Factors:

Parameter			
Musical	**Theoretical**		*Philosophical, music theoretical.*
	Practical	**Technical**	*Tools for realisation e.g., equipment, studios (constraints/potential).*
		Play	*Bottom up - type approaches, e.g., jamming, software/synthesiser exploration.*
		Practice	*Instrumental/vocal/studio etc.*
	Listening		*Influences, musical or other.*
Other	**Environmental**	**Macro**	*Time, place, culture etc.*
		Micro	*Room, ambience, etc.*
	Budget/ Resources		*Source of funds, amount.*
	Intended Audience		*Explicitly stated? Implications due to marketing?*
	Timeframe		*Time for realisation.*

Inputs:

Parameter		Experiential	Literature
Sound	**Sources**	*E.g., electronic, concrète, instrumental, vocal.*	
	Objects	*Individual elements – e.g. durations/amplitude envelopes, frequency ranges/pitches, wave types, description of samples, text fragments, composite objects (e.g., additive synthesis). (Roads, 2001, following Schaeffer)*	
	Object processing	*Types, e.g., delay, chorus, reverb, compression, gate, filtering, distortion, speed variation, LF modulation.*	
Rhythm	**Pulse**	*Present? Constant/variations? Tempo – constant/varied, sudden or gradual changes? (Solomon, 2002).*	
	Metre	*Ametric, polymetric, multimetric? Regular meter? Constant or changing? (Solomon, 2002)*	
	Patterning	*Description of patterns, changes, note values, accents, effects on other parameters (e.g. pitch) (Solomon, 2002).*	
Pitch	**Selection**	*Tonal, atonal, microtonal, polytonal, modal, chromatic, aleotoric? (Adapted from Solomon -"Tonality", 2002)*	
	Vertical structures	*Chord structures, voicings, intervals (Solomon – "Harmony", 2002).*	
	Vertical patterning	*Chord sequence/repetition/ variation/ rate of change/ root movement/ pedal point/drone. (Solomon - "Harmonic Motion", 2002)*	
	Horizontal structures and patterning	*Melody/Phrase/motive/riff – structure/contour/range/length. Description of patterns. Repetition/variation? (Adapted from Solomon - "Thematic/motivic structure", "Thematic/motivic development" and "Pitch range", 2002).*	
	Dynamics	*Constant/Changing? Independent of texture? (Solomon, 2002)*	
	Texture	*Monophonic, homophonic, polyphonic, contrapuntal, heterophonic? Homorhythmic? Thin/thick? Changing? Polarization (melody/accompaniment, solo/tutti, antiphony. etc.) (Solomon, 2002).*	
	Timbre	*Quality(ies). Static/evolving. Sound combinations/blending (Solomon, 2002). Vocal timbre – throat/head/chest and gender construction (Shepherd, 1991, p. 163).*	
	Spatial elements	*Nature of perceived acoustic space. (Wishart, 1996, p. 140) Stereo - left/right, near/far, static/dynamic, local/diffuse (Wishart, 1996, ch. 10). Reference to "Sound box" (Moore, 1993, p. 106) Multispeaker – placement, static/dynamic, trajectories, local/diffuse (Wishart, 1996, ch. 10).*	
	Programmatic Association	*Perceived meaning/connotation. Implicit/explicit, Intrinsic/contextual recognition (Wishart, 1996, p. 150). Sound image as metaphor? (Wishart, 1996, p. 165) Reference to physical gesture? (Middleton, 2000, p. 108) Signification/coding (Middleton, 1990, ch. 6). Syntax/semantics/ontological levels (Ferrara 1984, p. 359).*	
	Structure	*Form type(s). Small and large scale relationships. Derivation of structure – principle? (Solomon, 2002).*	
	Interaction	*Description. Rules? Standard types of interaction? (Rinzler, 1988)*	
	Score	*Existing? Type?*	
	Presentational Format	*Recording (format?)/ Live performance (venue?)*	

Table 1. Analysis Template

COMPOSITIONAL METHOD

A compartmentalized and hybridized approach to composition is adopted whereby various composition practices are identified during analysis forming a storehouse from which new compositional work is undertaken. The relationships between compositional methods identified via analysis and those utilised in the creation of new works occur on a range of levels and are particular to the subjective position of my role as participant analyst/creator. At the shaping factors level, some parameters are beyond my control. These include certain *technical, environmental, budgetary* and *timeframe* considerations. By framing my position at the shaping factors level in terms of resonance, links can be made both between works analysed and to the new creative works developed. For example, whilst not having the specific technical limitations involved in a particular work, the general notion of 'technical limitation' affords a point of departure for a discussion of such in the creative process. Similarly, the comparison of the shaping factors between works and to my own position offers the possibility of insight into aspects of continuity and/or disruption both within and across musical genres.

Some inputs can be directly incorporated into new works whereas others can be utilised at a more general conceptual level. For example, the use of a quadraphonic spatial arrangement can be directly transferred to a new work, whereas a concept such as the absence of a gestural connotation for the generation of sounds in an electroacoustic work can be transferred at a general level, regardless of the particular sound utilised in a new work. Similarly, certain inputs identified in the analysed works may provide a point of departure for the development of inputs for new works. The development of integral serialism in the 1940s and 1950s, whereby serial procedures were applied to a range of parameters (e.g., rhythm and dynamics) beyond pitch, provides one such example.

WORKS ANALYSED

The selected works analysed are listed below:

1. Pauline Oliveros: "Bye Bye Butterfly" (1965)
2. Miles Davis: "Bitches Brew" (1969)
3. Herbie Hancock: "Chameleon" (1974)
4. David Behrman: "On the Other Ocean" (1978)
5. Joel Chadabe: "Valentine" (1987 – 94)
6. George Lewis: "Voyager" (1987 – 95)
7. Courtney Pine: "Oneness of Mind" (1997)
8. *Interface*: "Scrb" (2000)
9. Dave Douglas: "November" (2003)

APPLYING ANALYTICAL METHODOLOGY

Completed analyses of improvised works are presented in Hill (2007). Most of the works analysed are recordings of live performances, with the exception of "Bitches Brew", "Chameleon", "Oneness of Mind" and "November" where some degree of overdubbing and/or editing of performance has occurred in the production of the recording. The limitations of a desk-based (i.e., using the recording as analysis object) approach to analysis include Butterfield's (2002) objection to the lack of context, or in the case of jazz performance, the 'carnival atmosphere' surrounding the event. Given that most of the works analysed were recorded live in a studio, not in front of an audience (except "Voyager" and "Scrb"), the 'carnival' is somewhat restricted to the happenings within the studio.

The reliance on a recording for analysis presents difficulties in the analysis of interaction. Firstly, the ability to identify sound source with a performer is crucial for any discussion of interpersonal/machine interaction. However, in the case of Chadabe's "Valentine", the situation is further complicated because the composer deliberately sought to obscure the differentiation between the computer and human performances. Secondly, in the case of ensemble interaction, visual cues can provide insight into levels of interaction that may not be apparent aurally.

It is possible to group the nine improvised works into two sub-groups: (a) works emerging from a 'jazz' tradition (including "Bitches Brew", "Chameleon", "Oneness of Mind", and "November"), and (b) works emerging from an 'art music' tradition (including "Bye Bye Butterfly", "On the Other Ocean", "Voyager", "Valentine" and "Scrb"). This division is made primarily on the basis of the stylistic history of the artist and is useful in the consideration of genre terrain.[1]

ASCERTAINING COMPOSITIONAL METHODS/ MAPPING GENRE TERRAIN

In terms of *shaping factors*, a somewhat diverse range impacts upon the creation of the improvised works analysed herein. At a *theoretical* level, one broad similarity between the composers/performers of the works is university or conservatorium study. Whilst most citations in the literature of such study pass unremarkably, neither Davis nor Oliveros view such study as particularly instrumental in developing their own careers. Oliveros, for example, identifies the disjunction between her own creativity and the instruction occurring in academia:

> I had a struggle for years fending off the structures that were being brought forward by instructors in academia … they had no relationship to what I was hearing. I resisted following the instructors' models.

[1] I acknowledge the problematic nature of this division, particularly the blurring of such boundaries by, for example, George Lewis' involvement with many artists working in a 'jazz' tradition, and also Joel Chadabe's album of reworked 'standards' - *After Some Songs* (1987-94).

> Somehow or other, the listening inwardly created the space to go ahead and the courage to do what I felt was important to do (Oliveros, 1993, p. 377).

Davis echoes this disjunction, withdrawing from the Julliard School of Music as he was finding more value in playing with and listening to New York jazz artists (Carr, 1998).

The need for some degree of performer autonomy is prevalent amongst the composers/performers of the improvised works. Remarks to this effect are found in the *theory*, *play* and/or *practice* parameters of the analyses. The notion of a 'shared creation' is posited by Behrman (1997) as a contrast to the composer/performer division common to the European art music tradition.

> There's the model especially in the European tradition of the Creative Superperson (the Composer), and the lesser worker musician (the performer) which I've wanted to get away from. I like the idea of sharing in the creation of something and don't mind getting less than 100% of the credit for it. I like designing software which can be lifted off the ground, so to speak, by a wonderfully imaginative musician who does something with it that I never would have dreamed of… The tradition of 'unfinished composition' of course is not new. Much of Jazz and other musics primarily designed for live performance have a lot to do with that kind of idea. You could say that when the composition is unfinished, authority is being questioned (Behrman cited in Gross, 1997).

Chadabe suggests that he composes 'activities' rather than 'pieces':

> A 'piece', whatever its content, is a construction with a beginning and end that exists independent of its listeners and within its own boundaries of time. An 'activity' unfolds because of the way people perform; and consequently, an activity happens in the time of living; and art comes closer to life (n.d.).

The degree to which performer autonomy is embedded within each of the improvised works discussed herein varies and appears to correlate with the extent to which such autonomy is discussed by the artists themselves in the literature. For example, in "Voyager", Lewis is explicitly interested in creating a non-hierarchical environment where the computer 'performer' displays a similar degree of autonomy to that of the human performer. On the other hand, in comments from Davis, from musicians playing with him, and from those influenced by him (e.g., Douglas), a picture emerges of an artist ('creative superperson?') whose whole 'musical conception' (Gitler, 2002) included the careful selection of side-people and musical materials alongside a "knack of pulling things out of musicians that they might not normally be aware of" (McLaughlin cited in Carr, 1998, p. 263-4). Thus although the performance of Davis' ensemble was experimental and improvised, there is a sense that Davis was in control over the direction of the sonic outcomes. This is at odds with both the non-hierarchical approach of Lewis and the clear delineation of roles outlined by Chadabe and Behrman above.

In addition to a range of contemporary musical and extra-musical influences, many improvising musicians also explore the potential of new technologies. Amongst the works analysed for this study, the rationale for, and nature of that exploration are quite disparate. Behrman and Lewis present a similar attitude towards the development of interactive computer works. For example, for Behrman, technology as such is amoral and dependent on the motivation of the user/developer (Behrman cited in Gross, 1997). Lewis (2000b) echoes this by linking broader social and cultural structures to software development:

> Musical computer programs, like any texts, are not 'objective' or 'universal', but instead represent the particular ideas of their creators. As notions about the nature and function of music become embedded into the structure of software-based musical systems and compositions, interactions with these systems tend to reveal characteristics of the community of thought and culture that produced them (p. 33).

Interaction between sound sources and/or performers occurs on a range of levels and via a range of parameters. A solo and accompaniment model is common to the jazz tradition works ("Bitches Brew", "Chameleon", "Oneness of Mind" and "November"). However the degree to which the accompaniment instruments are 'locked' to particular parts varies. In contrast to the solo/accompaniment model, collective improvisation involving all or some of the sound sources/performers is an aspect of most of the works and features most prominently in "Voyager" and "Scrb".

Rules for interaction are established in the interactive computer works and embed varying degrees of human control. Chadabe uses the software application M (which he developed), where basic MIDI information is recorded and then pitch, rhythm and timbre are manipulated by collections of algorithms accessed via graphic display by the user (Chadabe, 1997). The applications utilised by Behrman and Lewis operate in real-time without direct human control and are based on the computer 'listening' to the human performance and responding in certain ways. For example, the "Voyager" system, when multiple performers are present, decides which performer to listen to and whether to match, oppose or ignore various parameters (Lewis, 2000b). The development of such rules in some way represents a 'top down' process of creation. However, the sonic outcomes of such systems are determined by the performers and hence represent, simultaneously, a 'bottom up' approach. The traditional model of playing jazz standards, (i.e., the model used for many of the works considered here) where a soloist improvises over a set form, represents a similar merging of 'top down' and 'bottom up' processes.

APPLYING COMPOSITIONAL METHODS/ DEVELOPMENT OF IMPROVISED STUDY

Particular *resonances* observed in shaping the creation of the Improvised Study are listed in Table 2. The aspects of compositional methods, *elements of personal interest*, are identified in the second column of Table 3 below. The third column in Table 3 provides the detail of the study developed.

In terms of *shaping factors*, particular resonances observed in the creation of the study include *theoretical, technical, practical, environmental* and *listening* elements, these listed in Table 2. Of the elements listed, the most pertinent are *technical* aspects, particularly an interest in computer interactivity, and *listening* and *environmental* aspects, including an interest in a range of musical styles and live performance. On a *micro environmental* level, the Improvised Study was realised as a live performance in a studio setting, a process common to most of the improvised works discussed herein. In terms of the practical element, *play*, my interest in creating works that involve some degree of performer autonomy reflects the concerns elaborated by Behrman and Chadabe above.

Parameter			Resonances
Musical	**Theoretical**		• University music study (all).
	Practical	**Technical**	• Computer interactivity (Behrman, Chadabe, Lewis). • Use of computer as tool which effects creation process (Chadabe). • Belief in 'amorality' of technology (Behrman).
		Play	• Long history of live performance (all). • Interest in performer autonomy, i.e., breakdown of traditional composer/performer split (all).
		Practice	
	Listening		• Acknowledged range of influential music styles (Davis, Douglas, Hancock, Oliveros, Pine). • Interest in environmental sounds (Oliveros).
Other	**Environmental**	**Macro**	• Acknowledged desire to reflect current social/culture climate, particularly popular aspects (Davis, Hancock, Pine, Oliveros). • Interest in exploring experimental means (Lewis – AACM).
		Micro	• Live performance in studio setting (all except Lewis and Interface).
	Budget/ Resources		
	Intended Audience		
	Timeframe		• Initial recording of live performance in matter of day/s (all).

Table 2. Resonances Observed Between Factors Shaping Selected Key Works and Factors Shaping the Creation of the Improvised Study

Parameter		Elements of Interest	Elements of Study
Sound	**Sources**	• Traditional instruments (all). • Combination of acoustic instruments and electric/sampled sound sources (all). • Electric pianos (Davis, Hancock, Pine). • Computer generated samples/synthesis (Behrman, Chadabe, Interface and Lewis). • Turntable/vinyl sounds/samples (Douglas, Oliveros, Pine).	• Computer generated samples/synthesis, constructed in Reason and 'remixed' in Max/MSP. • Between one and three loops of original track heard.
	Objects	• Live instrument sounds (all). • Synthesised/sampled instrument sounds (Chadabe, Douglas, Hancock, Lewis, Pine). • Range of homogenous objects (Behrman, Chadabe). • Range of heterogeneous objects (Interface, Lewis). • Sustained wide bandwidth tone throughout (Oliveros). • Noise elements, including vinyl 'hiss' (Douglas, Oliveros, Pine).	• Synthesised/sampled instrument sounds including drumkit, acoustic bass, electric piano (Rhodes), acoustic guitar played at various speeds. • Range of homogenous 'meta-objects' created by use of up to three layers of same source.
	Object processing	• Delay (Davis, Oliveros). • Filtering (Douglas, Pine).	• Variable speed playback. • Delay.

Rhythm	**Pulse**	• Constant (Davis - mostly, Chadabe, Douglas, Hancock, Pine). • No pulse (or multiple, unmetrically related) (Behrman, Interface, Lewis, Oliveros).	• Mostly constant throughout with multiple pulse often. • Main loop 173 bpm (heard at 0'1'37" and 2'43 – 3'50"). • Middle loop 133 bpm (at 1'15 – 2'08")
	Metre	• 4/4 (Davis & Hancock – mostly, Pine, Douglas). • 5/8 (Chadabe). • Variations (Davis, Hancock). • Ametric (Berhman, Interface, Lewis, Oliveros).	• Main and middle loops 2/4.
	Patterning	• Repetitive, syncopated elements (Chadabe, Davis, Douglas, Hancock, Pine). • Constant 1/8th or 1/16th note phrases in solos (Chadabe, Davis, Douglas, Hancock, Pine). • Free (Behrman, Interface, Lewis, Oliveros).	• Repetition and syncopation in main groove. • Repetition of rhythms throughout, with variable speed creating isometric effect.
Pitch	**Selection**	• Limited pitch set (Behrman). • Tonal with some chromatic elements (Chadabe, Douglas, Hancock, Pine). • Polytonal elements (Davis). • Microtonal elements (Lewis). • Mostly non-pitched or indeterminate pitch (Interface, Oliveros).	• Tonal, polytonal and microtonal elements.
	Vertical structures	• Chord structures and voicings based on extensions and alterations of diatonic harmony as per jazz style (Davis, Douglas, Hancock, Pine). • Somewhat random vertical alignment of pitch elements (Interface, Lewis, Oliveros). • Range of intervals from semitone to octave utilised (all).	• Chord structures and voicings based on extensions of diatonic harmony as per jazz style. However multiple loops heard simultaneously and therefore actual vertical alignment of pitched events somewhat random. • Three modes of vertical alignment heard in study and result from various playback speed formula: '(a) matching' - where computer generated playback speed (*cps*) is a factor or multiple of player playback speed (*pps*)(i.e., 0.25, 0.34, 0.5, 0.67, 0.75, 1, 1.5, 2, 3 or 4 times player speed); (b) 'opposing' – where *cps* is determined by following (if *pps* < 2, *pps* + *cps* = 2, or if *pps* >2, *pps* + *cps* = 500); or (c) 'ignoring' – where cps changes at regular intervals by semi-random amount (using Max drunk object). • Intervals vary according to playback speed. At normal speed (i.e., speed of main loop) mostly stepwise or third intervals with some larger intervals up to and including a perfect fifth.
	Vertical patterning	• Static harmony or pedal point (Behrman, Davis). • Repeating chord progression (Chadabe, Douglas, Hancock, Pine). • Random (Lewis). • Nil (Interface, Oliveros).	• One layer provides static harmony for most of work. Main riff - contrapuntal elements outline Eb Minor 7 (dorian) tonality. Middle riff (at 1'15" – 2'08") – pitches include C – Eb – G (electric piano) and Gb – Db (guitar). • Somewhat random patterning from other layers and presents polytonal/microtonal elements at times. (Three playback speed modes – see above - determine actual combinations).
	Horizontal structures and patterning	• Motive development, use of sequences, diminution, augmentation etc., (Behrman, Chadabe, Davis, Douglas, Hancock, Lewis, Pine). • Apparent random elements (Behrman, Chadabe, Interface, Lewis).	• Repetition of elements throughout with isorhythmic shifts due to changes in playback speed. • Main 'loop feature repeating Ab – Gb - Eb descending figure. • Middle riff features repeats Eb – C (descending) – G (ascending).
	Dynamics	• Shifts due to texture (all). • Fairly constant throughout (Behrman, Chadabe). • Peaks and troughs creating tension and release (Davis, Hancock, Interface, Lewis, Pine).	• Shifts due to texture and performance. • Decrescendo on main loop from 1'20" – 1'37", fade to silence. • Some fades on other layers.
	Texture	• Instrumentation includes: drumkit, bass, guitar/keyboard, sax/trumpet (Davis, Douglas, Hancock, Pine).	• Varies between 1 – 3 homogenous 'meta-layers'. Each meta-layer includes drumkit, bass, electric piano and acoustic guitar samples. (See form

		• Homogenous (Behrman, Chadabe) • Thick diverse, heterogenous (Davis, Interface, Lewis). • Major shifts marking sections or start/end of solos (Davis, Douglas, Hancock, Pine).	diagram below, Fig. 1).
	Timbre	• Mostly static timbres throughout (Behrman, Chadabe, Davis). • Foregrounding of timbral transformation (Hancock, Interface, Lewis, Oliveros). • Use of extended instrument techniques (Interface, Lewis).	• Mostly static timbres throughout with changes due to variations in playback speed.
	Spatial elements	• Placement of sounds in stereo field (all). • Blurring of source of multiple timbrally similar sounds (Chadabe). • Lack of foreground/background relationships (Behrman, Interface, Lewis, Oliveros). • Foregrounding soloists in solo/accompaniment texture (Davis, Douglas, Hancock, Pine).	• Sounds mostly centred with some spread (via reverb) across stereo field. • Delayed sound heard at far left and right. • Lack of clear foreground/background relationships with consistent elements (i.e., main and middle loops) mostly at same volume as other elements. Exception to this at beginning (0 – 40”) where main loop is perceived as foreground.
	Programmatic Association	• Movement from centered to dispersed achieved by range of parameters including spatialisation, instrument roles, melodic/harmonic aspects, structure and effects (Davis). • Use of vinyl hiss adding ‘warmth’ to track (Douglas, Oliveros, Pine). • Random, ‘non-human’ feel at times (Behrman, Chadabe, Interface Lewis). • Repetitive, syncopated rhythmic aspects providing somatic/dance aspect (Davis, Douglas, Hancock, Pine).	• Juxtaposition of constant and varying elements (echoing centred/dispersed movement of Davis) created by maintaining one layer (main loop or middle loop) for most of work. • Random nature of changes in playback speed combined with ‘skipping’ feel of loops create non-human feel. • Repetitive and syncopated aspects provide somatic/dance aspect albeit at very fast tempo and with somewhat ‘unhuman’ feel.
	Structure	• Sectional with solo/accompaniment sections (Chadabe, Davis, Douglas, Hancock, Pine). • Free (Behrman, Interface, Lewis, Oliveros).	• Sectional: ABA Coda, with sections determined by presence of main (A) or middle (B) loops. Coda without main/middle loop. See form diagram below. • Length of sections freely determined during performance. • Main and middle loops not precomposed or predetermined, resulting instead from chance placement of start and end points on sample.
	Interaction	• Solo/accompaniment sections with some accompaniment parts ‘locked’ (Davis, Hancock, Pine, Douglas). • Collective improvisation sections between two or more sources (all). • Blurring of role between soloist/accompanist (Behrman, Chadabe, Davis, Lewis, Oliveros). • Non-hierarchical environment (Behrman, Lewis). • On timbral and textural levels (Interface, Lewis). • Computer interaction (Behrman, Chadabe, Lewis). • Computer interaction based on rules – match, oppose or ignore (Lewis). • Tension and release principle achieved via variety of parameters, e.g. pitch, dynamics, rhythm, texture, and/or structure (all).	• Computer interaction system developed in Max/MSP. Multiple sample (one to four) playback with computer controlling playback speed of two samples. Rules based on ‘match’, ‘oppose’ or ‘’ignore’ principles (Lewis). However player controls other parameters of computer playback (e.g., interaction mode, sample selection, volume and loop start and end point selection). • Level and nature of interaction varies between sections. See form diagram (Fig. 1) for graphic representation. • ‘Matching’ and ‘ignoring’ modes of interactive system utilised. • ‘Locking’ of some parts (e.g., main and middle loops). • Tension and release achieved by: removal and return of main loop; alternating between ‘matching’ and ‘ignoring’ modes of computer (e.g., 2’58” – 4’22”).

Table 3. Elements of Personal Interest and Elements of Study

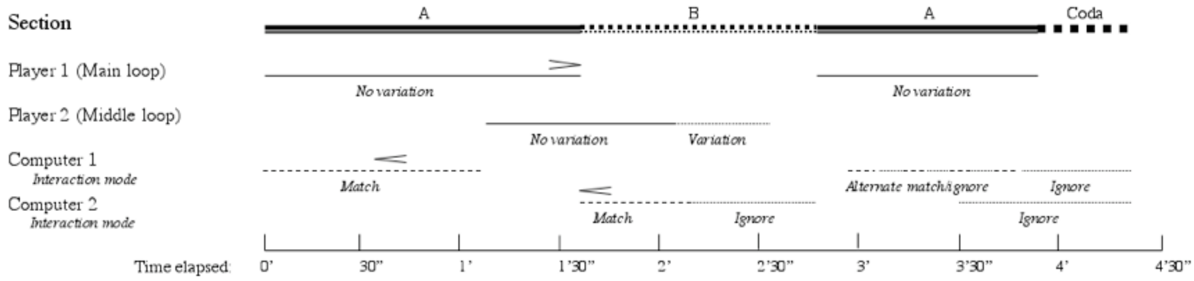


Figure 1. Improvised Study: Form diagram

In terms of *inputs* for the study, many of the *elements of interest* (listed in the second column of Table 3) were incorporated into the study (listed in the third column of Table 3). This extends to the incorporation of somewhat disparate elements of interest within individual parameters. For example, the *sound source* utilised for the study is a draft of an instrumental work for drumkit, electric piano, double bass and guitar that was realised using samples within the Reason software application. For the study, multiple short sections of this work are sampled and played at various speeds. Thus, in some way, all of the sound sources listed as elements of interest in Table 3 are incorporated into the study. Samples of traditional instruments, including electric piano, are replayed in the study in a manner echoing the use of a turntable as source in both Oliveros' "Bye Bye Butterfly" and Pine's "Oneness of Mind". Furthermore, as in the Pine example, samples of my own performance are used as a sound source.

Other examples of the incorporation of disparate elements of interest into the study include elements of *rhythm* and *pitch*. In terms of rhythm, for the works analysed, two categories can be established; (a) works with a constant pulse and (b) works without or with multiple or unmetrically related pulse. For the study both categories are explored by foregrounding a constant pulse in some sections and overlaying irregular/unmetrically related fragments at other times. In terms of *interaction*, I was particularly interested in the interactive computer systems developed by Behrman, Chadabe and Lewis. An integral component of all of these systems is analysis and computer performance on a note-to-note level, and the incorporation of some traditional pitch elements. However, given my ongoing interest in utilising field recordings and sound fragments above and/or below note level in terms of timescale the study incorporates computer interactivity at a conceptual level, not a note-to-note level. Instead, the study is an attempt to incorporate computer interaction within a system of multiple sample playback modules in an improvised performance context.

REALISATION OF IMPROVISED STUDY

The performance system developed in Max/MSP for the Improvised Study extends from my previous work with field recordings where various parameters such as start and end points of loop, playback speed and spatialisation of various sound fragments, between approximately three and thirty seconds in length, were manipulated in real time. The Max/MSP abstractions utilised were further developed for performances throughout 2004 and 2005 with the interactive aspects constituting the new component developed for the Improvised Study. For the study I sought to explore the computer automation of playback speed in terms of the rules of interaction utilised by Lewis in "Voyager", i.e., the three modes, matching, opposing or ignoring. However, whilst the system developed for the Improvised Study could be utilised in a 'non-hierarchical' (Lewis, 2000a) manner, my own exploration of the system involved some degree of control of the computer's output. This included at a basic level, sound source selection and volume control. Figures 2 and 3 present the basic audio and interactive control flow charts for the abstractions developed in Max/MSP, with key objects indicated where appropriate.

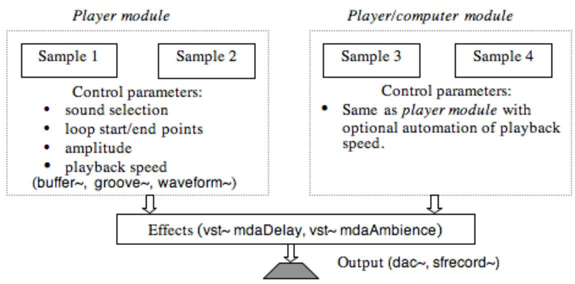


Figure 2. Improvised Study: Basic audio and control flow chart

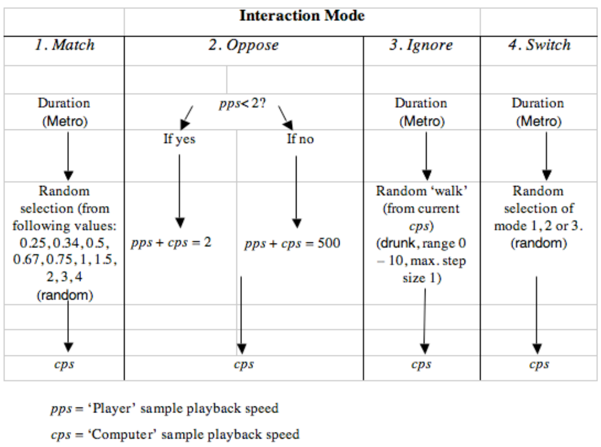


Figure 3. Improvised Study: Interaction mode flow chart

Given that the system developed was an extension of my earlier work, minimal system testing was required. However, once operational, a lengthy play process was undertaken where a range of sound sources, interaction modes and performance techniques were explored. Numerous performances were recorded and reviewed, most featuring the utilisation of only one interaction mode. Through this initial process of minimisation where, for example, only the matching or opposing modes of interaction were utilised, a more thorough exploration of the various modes and potential sonic outcomes was possible. Having established some fruitful performance avenues, further reflection on the various compositional methods identified through the analysis of the selected key works was conducted before completing the study. The Improvised Study was selected from numerous performances conducted in a home studio environment, monitoring with headphones and utilising the computer mouse, trackpad and computer keyboard for control. The study was recorded directly within Max/MSP and transferred to Pro Tools for normalisation with no further editing or effects added.

CONCLUSION

Having completed the analyses discussed herein and another twenty-seven in other genres, refinement of the analytical template is recommended by subsuming *timbre* within the description of individual *sound objects* and an expansion of *programmatic association* to include sub-parameters such as *gestural reference* (following Middleton, 2000), *signification/connotation* and *intrinsic/ contextual recognition* (following Wishart, 1996). In addition to the desk-bound method presented here, an ethnographic approach could also be used to probe collaborative processes. The relationships that develop within a studio environment between musicians, engineers and producers warrant detailed study in order to establish a thorough picture of the nature of music creation.

The analysis/creation model developed for this research makes explicit the connection between musical analysis and music creation; a connection not made in the majority of existing analytical methods. The compartmentalisation of musical inputs as discrete elements enabled the creation of a storehouse; an accumulation of discrete elements (i.e., compositional methods) at/from which I could ponder, select, combine or appropriate in the creation of the new practical works. The relative fixity of the shaping factors elements highlight individual agency and offer a starting point for consideration of the impact of such factors as gender, class and race on music creation.

REFERENCES

Bahn, C., Cook, P., & Truman, D. (2000). *Alternative voices for electronic sound: spherical speakers and sensor-speaker arrays (SenSAs)*. Paper presented at the International Computer Music Conference, August 27 – September 1, 2000. Retrieved May 24, 2006, from http://www.music.columbia.edu/%7Edan/alt_voices/index.html

Brown, M. (1997). "Little Wing": A study in musical cognition. In J. Covach, & G.M. Boone (Eds.), *Understanding rock: Essays in musical analysis* (pp. 155 – 170). New York: Oxford University Press.

Butterfield, M. (2002, Spring). Music analysis and the social life of jazz recordings. *Current Musicology,* 2001-2002, 324-352.

Carr, I. (1998). *Miles Davis: The definitive biography*. London: HarperCollins.

Chadabe, J. (n.d.). *Homepage*. Retrieved June 30, 2006, from http://www.chadabe.com/

Ferrara, L. (1984). Phenomenology as a tool for musical analysis. *Musical Quarterly, LXX*, 3, 355-373.

Gitler, I. (2002, August). Backstage with Dave Douglas. *Down Beat, 69*, 8:12.

Gross, J. (1997). David Behrman [Interview]. *Perfect Sound Forever*. Retrieved June 26, 2006, from http://www.furious.com/perfect/behrman.html

Hill, M.F. (2007). *Hit Scrape Click Drag: Analysis and application of compositional methods at the intersection of conserved and emergent technologies*. PhD Thesis, James Cook University.

Hubbs, N. (2000). The imagination of pop-rock criticism. In W. Everett (Ed.), *Expression in pop-rock music* (pp. 3 – 30). New York: Garland.

Lewis, G. (2000b). Too many notes: Computers, complexity and culture in "Voyager". *Leonardo Music Journal*, *10*, 33-39.

Middleton, R. (1990). *Studying Popular Music*. Milton Keynes: Open University Press.

Middleton, R. (2000). Popular music analysis and musicology: Bridging the gap. In R. Middleton (Ed.), *Reading pop: Approaches to textual analysis in popular music* (pp. 104 – 121). Oxford: Oxford University Press.

Moore, A. F. (1993). *Rock: The primary text*. Milton Keynes: Open University Press.

Oliveros, P. (1993, Autumn). Cues. *The Musical Quarterly 77*, 3, 373-383.

Rinzler, P. (1988). Preliminary thoughts on analyzing musical interaction among jazz performers. *Annual Review of Jazz Studies, 4*, 153-60.

Roads, C. (2001). *Microsound*. Cambridge, MA: MIT Press.

Scaruffi, P. (1993). *Pauline Oliveros* [Interview]. Retrieved March 30, 2007, from http://www.scaruffi.com/interv/oliveros.html

Shepherd, J. (1991). Music and male hegemony. In R. Leppert, & S. McClary (Eds.), *Music and society: the politics of composition, performance and reception.* Cambridge: Cambridge University Press.

Solomon, L. (2002). *Music parametric analysis*. Retrieved February 15, 2006, from http://solomonsmusic.net/paramet.htm

Tagg, P. (2000). Analysing popular music: Theory, method, and practice. In R. Middleton (Ed.), *Reading pop: Approaches to textual analysis in popular music* (pp. 71 – 103). Oxford: Oxford University Press.

Wishart, T. (1996). *On sonic art.* Amsterdam: Harwood Academic Publishers.

TOWARDS A DEFINITION OF THE PERFORMING AUDIOVISUALIST

Lloyd Barrett
Queensland University of Technology
Music

Andrew R. Brown
Australasian CRC for Interactive Design
QUT

ABSTRACT

The evolution of the laptop computer as a musical instrument in the 1990s heralded a rising wave of digital performance activity that highlight a vastly different relationship between performer and instrument to traditional embodied (instrumental) or disembodied (acousmatic) approaches. As notions of performativity are subjected to constant critical reconfiguration, artist and software producers have responded with a range of innovations that allow for enhanced expressive possibilities and the ability to define personalised modes of performance. The increasing ubiquity of digital media combined with the power of current generation notebook technology has provided the perfect platform to realise integrated audio-visual toolsets that respond to musical controllers and provide mixed-media results. Despite emerging practitioners increasingly availing themselves to the musical affordances of this technology, theoretical discussion in the field ignores the various approaches a solo musician might take in developing integrated media works for performance. In an increasingly crowded niche there is a clear compulsion to consider expanded modes of performance, yet lacking any formal framework these integrations can easily alienate an audience, distract from performance and lead to criticisms of novelty for novelty's sake.

As an emerging area of practice where the interoperability between sound, image, text and gesture is dissolved, this audio-visual form of expression requires a formalised typology that extends beyond novelty and technological determinism; beyond the restrictions of a traditional musical performance and the confines of disciplinary boundaries towards an integrity of context, form and function suitable to a hybrid medium; to a new audiovisual practice that emerges from histories of live performance and media production. This paper will outline a framework for audiovisual creation by applying the musical theory of Morphopoeisis, devised by Panayiotis Kokoras, to a variety of hybrid media practices and by outlining a number of performative approaches towards a definition of the performing audiovisualist.

1. INTRODUCTION

In September 2008 the following discussion took place on the Audiomulch mailing list:

Korhan Erel: ...the audience in Turkey is fairly new to experimental, avant-garde music and there is a danger of alienating them when you give them nothing to relate to except for the music. Using video or working with VJs provide a practical solution to this, but I would prefer either to prepare the video myself or design the whole performance from scratch with the VJ.

John Smith: By using visuals it's like admitting that music just isn't enough to sustain the audiences interest. Working with a visual artist for a specific concept is always a good idea but again I feel that music loses its unique power and the opportunity to live the excitement of "pure" sound. A friend who doesn't use visuals told me once, in sarcasm: "I can watch TV at home!"[65]

This dialogue illuminates the theoretical and practical interplay between sound and image in the field of digital media performance. Disagreements over instrumental and acousmatic approaches to digital media performance and the "novelty" inclusion of hybrid media certainly inform current practice, highlighting a need to examine the effect that the growing integration of hardware, software and cultural practice is having on the performer.

In 'Haunted Weather' David Toop describes the dream of "creating an overwhelming synaesthesia" as being subservient to the "false assumptions or deep seated needs [to see a clear] discernible link" between visible actions and sound production that creates a "warm glow of communication" with the audience. [62] Musicians and sound artists have historically deployed any number of tools to define, communicate and distribute their creativity; at times directly challenging rather than reinforcing the preconceptions of an audience. Steve Dixon states in the preface to Digital Performance, "...music was one of the first artistic fields to experiment significantly with and embrace computer technologies, and in terms of both creative production and commercial (as well as illegal) distribution, music has arguably been more radically revolutionised by the 'digital revolution' than the other performance arts."[19] Instead of relying on traditional models of music performance the new breed of performer is exploring the unique possibilities that a partnership with digital technologies can offer to composers and performers.

The performing audiovisualist, strives to achieve an overwhelming synaesthesia, through the integration of a variety of expressive forms in contemporary performance. Both the D.V.D. Trio and the duo of Sabine Ercklentz and Andrea Neumann received "Honorary Mention" awards at the Prix Ars Electronica Cyber Arts 2008 festival. [44] Both integrate audiovisual components to suit completely divergent performative needs. At the same festival the Golden Nica was awarded to the Reactable [39]; an instrument/interface that adopts an extensible visual music approach to live performance that has been used in concert by high profile artists like Bjork. The synthesis of audio and vision practices is also evident in cultural commentary through blogs like Create Digital Motion [14] and Skynoise [33] that output a constant source of commentary on digital hybrids, soft/hard hacks and performative revelations. These activities highlight the growing integration of sound and image as a performance practice that extends the notion of the musician and musicianship in the twenty-first century.

These integrative trends have been noticed in scattered academic reflections at least since Marshall McLuhan's publications in the 1960s [49] and have been acculturated by the isomorphism of digital data. As a keen observer of these trends Stephen Holtzman states, “an expression is an expression of its time [and also] an expression of the idiomatic nature of the medium by which it is realized.”[34] When the computer considers all media as numeric data, and software exists that makes little to no distinction between media types, the implication seems to be that interdisciplinary practice is the natural outcome of computational arts.

As an emerging area of practice where the interoperability between sound, image, text and gesture is dissolved, this emergent new form of expression requires a framework that extends beyond novelty and technological determinism; beyond the restrictions of an instrument/performer/audience approach and the confines of disciplinary boundaries towards an integration of senses and an integrity of context, form and function suitable to a hybrid medium; to a new audiovisual practice that emerges from histories of musical performance and media production.

2. A FRAMEWORK FOR AUDIOVISUAL CREATION

Emerging approaches to audiovisual performance may draw on the richness of established musical performance practices but they distinctly move in many directions away from what might be considered the traditional instrumentalist approach. In defining the foundation of what audiovisual performance is, and how it differs from a more traditional musical performance, it is worth identifying the deeply rooted performative approaches that resonate with this emerging form. To facilitate this approach we will draw on the theory of Morphopoiesis [42] a model that was originally designed to assist the perception of musical structure in electroacoustic works. The model lends itself particularly well to a generative, as well as its intended analytical, function.

Morphopoiesis describes compositions as being based on atomic sound transformations that have a characteristic spectromophological trajectory. These sound objects are animated or set in motion through musical space until the trajectories are integrated into a sonic fabric. As a framework for audiovisual creation this model can easily include audio and visual elements; events that we call projected objects, which become animated objects as they are transformed over time. These animated objects can have varyingly independent or coordinated trajectories in audiovisual space that eventually come together as narrative objects around a common theme or purpose with an ambition to achieve aesthetic coherence and affect.

2.1.Projected Objects

Among historical examinations of audiovisual performance the legacy of the Magic Lantern is a recurrent theme. While the mechanics of the device were designed to project images to support story telling; in practice their application played on superstitions of the time by projecting corpses, ghosts and supernatural apparitions to shock the audience. Illusionists like Johann Georg Schröpfer experimented with projecting onto mirrors and smoke to create disembodied figures that appeared to float in the air above the audience. He also used hidden hollow tubes to distort vocal utterances to further project the illusion that these phantasms were more than just clever tricks. [28]

The floating phantasms of the Magic Lantern would rise again in expanded cinema experiments taking place from the late 1960s. A key work from this period is "Line Describing a Cone"(1973) by Anthony McCall. A 2005 screening presented by Other Film in Brisbane took place in the garage of a local art gallery; an appropriate space for a work that does not require a screen, only darkness and space. A single beam of light extends across the space to create a dot that slowly forms a circle. As the circle is projected, the light extending from the projector creates a cone that evolves to a hologram compelling the audience to interact and explore. [37]

The audiovisual work of Robin Fox has progressed from initial work developing tones that produce complex replicable mandalas on an oscilloscope. [24] Fox has since relocated the visualisation to lasers, which extend the interaction of sound and image visibly and physically into the audience. As a solo artist he generates these results from a relatively compact setup (the major components being notebook, mixer and laser) and has the portability required to travel the world like the archetypal showmen of the Magic Lantern period. Though his work comes from an experimental music background it has been licensed by the likes of The Chemical Brothers [43] and there are signs that the VJ/Dance scene is starting to catch up. In 2007, French visual architects Exyzt produced an astonishing projection mapping for the Transmusicalles Festival set of house producer, Etienne De Crecy with a projector, well-defined spatial coordinates and some simple software enabling them to manipulate vertices to provide a spatial illusion. [54] De Crecy DJ's in the centre of the scaffolding which appears to twist and morph like a hybrid of M.C. Escher and a Rubix Cube.

In the examples above the key similarity is the projection of one or more abstract, figurative or cultural objects into space and the solidification of an illusion generated in partnership with the audience in that space. Despite the consumer availability of surround-sound technology, Adam Donovan is one of the few artists working on the focused projection of sound objects. His parametric acoustic arrays are lenses that focus a highly directional, 3˚ beam of sound across a space of approximately 200 metres. [10] While Donovan's research is generated under the auspices of the Defence Science and Technology Organisation, this technique has potential for the performing audiovisualist in further integration of sound and image in live delivery.

2.2. Animated Objects

A ventriloquist act involves the animation of an inanimate object through the projection of voice and motion. The ventriloquist puppet is also representative of a partnership with technology [3] where technologies are a ‘double’ that can be reflecting, conflicting, transcending or replacing the human performer in modern dance production. [18] The Swedish electronic group The Knife literalise the concept of a technology ‘double’ through use of a projected digital avatar in concert that sings both solo and backing parts, as well as appearing to manipulate an imaginary machine. [2] The notion that these puppets develop a "life of their own" is a horror explored in literature and a dream explored in code as autonomous artificial life entities determine an engagement with the real and virtual world via algorithms.

Deeply ubiquitous automated processes spawn beneath our feet, behind any program we use and in the infrastructure that runs both our virtual and real cities. A transparent use of this automata can be seen in programmable environments like Max/MSP [15] and Impromptu [60] that allow for autonomous processes to occur and respond to the musical feedback generated between user and whichever process is the focus of their interaction. Animata, by contrast, is a program that explores the puppet metaphor more directly, allowing the generation of 2D "puppets" from bitmaps that can be then made to move and interact based on sonic input. [51] Dave Griffiths creates multiple automated avatars in live coding performances with his software Al Jazari. [30]. In this environment he literally animates the computational processes; but the animated object need not always be so literally set in motion.

Oskar Fischinger's pioneering animation of simple shapes and colours moving with sympathy to music is a realisation of the links between colour, shape, sound and mood that Wassily Kandinsky proposed in "Concerning the spiritual in art.”[40] Musical sounds are associated with visual symbols for the audience, producing a meaning that is neither exclusive to sight nor sound but to what Michel Chion calls "transensoriality". [11] Fischinger and New Zealander, Len Lye not only pioneered Visual Music but repurposed the Wagnerian convention of the leitmotif in the application of sound to (abstract or geometric) objects creating higher-level meaning and personality. This idea would resonate throughout 20th century art, animation and even multimedia with the introduction of graphic user interfaces and the icon (earcon).

Approaches to animating objects can link together audio and visual elements, either directly as sonification or visualisation or through more subtle or indirect mappings. Systems like Metasynth [63] generate sound from image in linkages that lead back to the colour organ inventions of the 1800s and synaesthetic composition experiments by the likes of Schoenburg and Scriabin. Where the basic premise of a colour organ is the projection of colour in relation to note and keypress, the aforementioned programs connect sound and image through the science of spectral analysis; converting colour within the spectrum to a relative frequency. The Processing software [26], among its many pragmatic uses, allows the user to construct live painting interfaces where gestural control, close to traditional action painters like Jackson Pollack, can be achieved with instantaneous image/sound results using sound and music library extensions such as Minim and SoundCipher. [4] The ability to create complex visuals has encouraged groups like the Anti-VJ collective to design their own complex systems for interactive painting. [55] The virtualisation of these forms through digital technology allows for endless prototyping and adjustment in order that the animation of the inanimate through sensory integration can be controlled by process and/or gesture in a manner best suited to context and the stimulation of meaning for the audience.

2.3. Narrative Objects

Experiments with integrating music and song with other forms of media to expand the construction of narrative has a strong history, from the leitmotifs and arias that outline the libretto in an opera to the recontextualisation of Power Point software by David Byrne. [6] The Residents were early adopters of CD ROM technology and used it not merely as an extensible song format but in the creation of artistic experiences where interaction is the key to unraveling layers of meaning. A DVD reissue of The Commercial Album [Residents] features a labyrinthine 3D gallery that the viewer navigates with the DVD remote in order to access the variety of videos produced by the band and their fans.

The invention of the remote control and the interactivity it affords is, according to Peter Greenaway, the moment that cinema died. [29] His argument for the radicalisation of filmic art is in part due to his disagreement with a form of cinema that is reliant on text over image; particularly its fundamental basis in treatments, storyboards and scripted narrative. [29] Lev Manovich notes that Greenaway's work 'The Falls' demonstrates an early rejection of traditional narrative form [46] in favour of a case study documentation of 92 lives affected by an imaginary event. With the assistance of the No-TV collective, Greenaway has converted his Tulse Luper project into a form of performance cinema by adopting the performative approach of the VJ with a touch-based gestural editing system. [53] Working alongside live electronic musicians, originally static forms are opened to the possibility of live remixing and recontextualisation, where the audience is taken on a journey unique to the performance.

Brisbane interdisciplinary artist Chloe Cogle uses recontextualisation of sound and image to frame her performances; combining atmospheric mechanical sounds by Luke Walsh with archival slide projection and a faux 'authorial' narration that bends truth just enough to maintain audience intrigue. [67] The performative reconfiguring of discarded objects is also at the centre of The Labyrinth Project; a growing work of 'database cinema' that evolves from the rescue of archival footage and interacted with via a tagging system allowing artists and audience to define their own narrative structure. One of the artists involved in this project, Marsha Kinder disagrees with Manovich's assertion that database cinema is anti-narrative [46] arguing that databases and narratives are "compatible structures whose combination is crucial to the creative expansion of new media" [41] She refers in particular to the "dual processes of selection and combination" that are the foundation of all stories, songs, and narrative forms regardless of their linearity or chance construction. By appropriately tagging audio/visual data, the resulting construction can still be made to follow a developmental path whilst also "challenging the notion of master narratives whose selections are traditionally made to seem natural or inevitable." [41]

One of the preeminent cinematic influences on modern audiovisual performance is the structuralist aesthetic pioneered by Dziga Vertov in his "Man With A Movie Camera". [17] This multilayer film exhibits strong rhythmic approaches to editing and juxtaposition that also inform Godfrey Reggio's "Qatsi" trilogy [56]; which express a battle between humans and technology that resonates not only with the organisation of images but through the Phillip Glass scores that mesh synthesised arpeggiations with classical instruments and choral voices. Despite the absence of a narrative voice channelled through narrator or characters, authorial intent is made clear through the choice and combination of sound and image. While works may resist a traditional narrative arc, the connections apparent behind the choices made allow the user to draw their own conclusions and meanings. The narrative may be implied, improvised or have aleatoric elements but structure is never absent.

2.4.Summary

A digital musician constructing an audiovisual work must consider the trajectory from audio/visual object through animation and development towards the construction of a meaningful dialogue with the audience. Tom Ellard (Severed Heads) believes that the "…performance video has yet to escape a trivial 'eye candy' level. It is still assessed in terms of equipment/technique…[something that] mature artforms, such as film have been able to escape."[32] The development of tools that integrate the projection, animation and construction of audiovisual objects should discourage the reliance on arbitrary wallpaper visuals. Adapting the theory of Morphopoiesis to a framework for audiovisual creation is particularly useful in that it does not constrain the artist to a fixed modality; rather it provides a foundation for the construction and deconstruction of audiovisual works that support the divergent approaches that lead to a signature work.

3. APPROACHES TO AUDIOVISUAL PERFORMANCE

In composing an audiovisual work the practitioner may be influenced by any number of cultural artefacts and practices. Translating these ideas into performative works requires a focused set of skills extending from musicianship into various other hybrid media capabilities. The following sections outline key roles that the digital media performer engages in whilst translating a creative idea into a manageable practice.

3.1.The Solo Artist

Gone unmentioned to this point is that the term 'Audiovisualist' is singular and aligns with a tradition of 'one man bands' and other largely solo practices. This is in contrast to, and in recognition of, collaborative and interdisciplinary audiovisual works, such as Opera, that have a well-established history. 'Total Art Work' (Gesamtkunstwerk) as outlined by Richard Wagner and explored via large-scale operatic works performed at the specially designed Bayreuth Festspielhaus, might well be considered the forerunner to interdisciplinary performance, yet from a pragmatic perspective any definition of the performing audiovisualist is situated on the opposite end of the industrial relations scale to these lavish production projects. While perhaps lacking the prestige that serves as an important cultural cache, the performing audiovisualist is a much more sustainable proposition, being identified for the purposes of this paper as a singular solo digital artist that operates on a much more personal and portable scale.

By way of example, Bruce McClure has been described as a "moving-image magician" [25] but he himself prefers the term "one man band". [9] His work exemplifies a unique approach to solo AV practice as he uses 16mm film projection loops and guitar pedals to create trance inducing minimal beats for the eye and ear. Ryoji Ikeda explores a similar form of synaesthetic minimalism from a digital perspective in his work documented on the DVD, "Formula". [38] As a sound artist he is informed by his work with the equally experimental Dumb Type theatre company; it is acousmatically driven and his absence from the stage is covered by simple visual elements that are synchronistically integrated with his sparse sound-works. Due in part to the minimalist nature of his source material, the form of his pieces develop a natural structure expressed equally through sound and image. A more instrumentalist approach is that taken by Yoshimitsu Ichiraku aka Doravideo; who attracted an honorary mention at Ars Electronica 2007 for his integration of VJ mash-up aesthetics triggered via live drumming. [36]

While there are variable physical and psychological limits to the solo performers ability to control parameters in performance, the use of prepared materials and computational processes as performance prosthetics combined with the potential for significant fluency and skill development by practitioners means that the horizon of solo performer capability can seem quite distant. David Saltz states that the use of pre-rendered audiovisual material makes the live performer "subject to the tyrannical inevitability of the linear media... which saps live performance of its most critical values: spontaneity and variability."[27] In contrast Dixon embraces "the playful and transparent illusion of interaction" as a major reason to include pre-recorded media in a performance. This inclusion does not "trouble or blur the ontological distinctions involved; rather, they get their performative punch by highlighting them."[27] In designing the software platform Isadora, Mark Coniglio, a student of Morton Subotnick [60], has attempted to address the issues faced by the Troika Ranch dance group he works with: "Digital media is wonderful because it can be endlessly duplicated and/or presented without fear of the tiniest change or degradation. It is this very quality (the media's "deadness") that is antithetical to the fluid and ever changing nature of live performance."[12] From his perspective "the performers must have latitude to improvise if they are to take advantage of interactivity [and] the audience must have some understanding of the interaction [and/or] the instrument with which the performer controls the manipulation in order to complete the loop between audience and performer."[12] The performing audiovisualist must consider interaction as an essential element of performance design to ensure that they can actively demonstrate control of the material to the audience.

The minaturisation of various recording, production and distribution technologies has provided the performing audiovisualist an extraordinary amount of freedom to locate and generate source material. Veteran audiovisualist Tim Gruchy notes that his "working practice is a seamless folding of the concepts of work and life."[23] Being able to instantly capture and store material with one of several pocket sized HD cameras on the market for later reuse has the potential to connect solo audiovisual practice with that of the roving bard or indie-media journalist; constantly recording, remixing and presenting AV material in a semi-permanent state of creative autonomy. With a small amount of thought and effort, the bus ride home can translate to an alluring abstract visualisation with appropriately contextual sound/image manipulation.

3.2.The Technologist

In working towards a definition of the performing audiovisualist, it is necessary to understand that we are describing a creative practice that is enabled, but not replaced, by new tools and technologies. When media theorists such as Manovich suggest that computer scientists are the greatest artists of today and the greatest artworks are new technologies [47] we should be conscious of the temptation to fetishise "the technology without regard for artistic vision and content." [19] Dixon goes on to suggest that "Manovich's formulation encapsulates an indiscriminate techno-postmodern aesthetic theory of infinite (yet always-already recycled) possibilities..." that serve to "mar rather than advance critical understanding of relationships between technology and art."[19] Michael Faulkner, VJ theorist and director of D-Fuse argues that technological redundancy establishes art, using the elevation of painting to "high art" around the invention of photography as an example. He suggests that artists are more likely to use redundant technology as an affordable means to express their creativity. [50] We believe that technology is a significant, but not dominant, influence on the performing audiovisualist; that motivations, creative skills, knowledge, and sensibilities largely define and shape this emerging practice, as they do all creative practices.

Having made our precautions clear, artist/technicians like Robert (Ableton Live) Henke demonstrate the responsibilities of the solo performer to turn their hand to many aspects of the practice, as technologists, and possibly inventors of the tools and techniques necessary to realize their performances. This can make the distinction between tool developer and tool user a very blurry one. Examples of early audio visual inventions include the Frederic Kastner's Pyrophon, which worked by igniting gas in different sized glass chambers, producing strange tones and light effects. [16] Steve Langton and Hubbub produced a variation on this design for the REV festival at the Brisbane Powerhouse in 2002 [35] that was also demonstrated in pirate-ship form in 2009 at the Melbourne Docklands. The interlacing of science and art in performance can also be seen in the many variations on the Color Organ principle. In the 1800s Daniel Vladimir Baranoff-Rossiné perfected and exhibited the Piano Optophonique; an instrument that utilised coloured disks, mirrors and lenses to project abstract moving colors when the traditional keyboard was played. The link between instrument and visualist gave Daniel "an unusual freedom in exploring dynamic painting that I could hardly have dreamed of before." [66] Also of note is the buried history of synaesthetic invention by Percy Grainger, uncovered by Warren Burt in a paper delivered at the Australasian Computer Music Conference in 1994; he describes the "Electric Eye Tone Tool" as a "seven voice instrument with seven sine wave oscillators controlled by variations in light on a series of 14 photocells. Patterns painted on a large plastic sheet pulled across the plate of the instrument [causing] the variations in light."[5]

The technological foundations for the performing audiovisualist were laid in the 1980s when the synthesised image became a ubiquitous presence on TV in the form of motion graphics, and the advent of MTV brought with it a commercialised aspect that owed more to advertising than audiovisual experimentation. Around this time Stephen Jones debuted on Metro TV in Sydney with Severed Heads, providing a demonstration of his video synthesiser used to mix colorized patterns with taped footage and played much like the other members of the band played their audio synthesizers. His influence was demonstrable in the development of visualisations for rave events in the late 80s/early 90s, particularly with artists like Subvertigo, who modified Panasonic MX10 video mixers to allow the use of a Luminance Key; mixing disparate footage into pixellated surrealism whilst maintaining a political edge. [32] Software tools like VJAMM [7] integrated the ideas of these experimenters with the needs of the modern VJ in respect to the electronic dance music scene. While these tools finally allowed a more synaesthetic integration in composition and performance, they were restricted by the demands of a relatively narrow set of usage requirements; looping, sampling, mixing, projecting; and were limited by processor hungry video demands that easily exceeded the abilities of then current technology. These boundaries however often work to focus creativity as in the case of Botborg [1], who utilise a simple audio/video feedback system combining digital sound with analog vision mixing, providing the flexibility to improvise a direct audio/visual synchresis through tightly organised bursts of epileptic malfunction. While their work harkens back to early 90s VJ work, their approach is akin to electroacoustic improvisation and has been adapted to suit contexts as broad as Australia's What Is Music festival and European 'Electroclash' club nights.

3.3. The Entrepreneur

In the May 2009 issue of Prospect magazine, Brian Eno notes that "digital technology has made music easier to make and copy, with the result that recorded music is about as readily available as water, and not a whole lot more exciting" this has in turn meant that "unable to make a living from records sales, more and more bands are playing live." (22) The ubiquity, power and portability of notebook technology is one of the primary economic reasons why audiovisual performance is emerging as a low-risk, low-cost, itinerant and flexible form powered by the increasing democratisation of technology and the use of both proprietary and open source platforms to generate performative solutions that adapt to varied live contexts.

In an academic forum addressing the sustainability of musical forms, Huib Schippers explained that maintaining the committed interest of a community and audience is one of the key challenges within any art form [58]. Attunement to preferences and trends is essential in a field where innovation is a commodity. Paul Spinrad expands on this by placing awareness of the audience at the centre of the development in audiovisual performance. "As an audient, you don't just hear the applause and laughter... You're part of one enormous brain that, among other things, is working out the problem of how people in the surrounding culture and at the present moment react to things, and what reactions are and are not appropriate." [62] It is interesting to note that, despite being birthed in the realm of dance music and clubs, the place of a VJ within that scene is heavily problematised. Aside from playing second fiddle to the whims of the DJ and promoters there is also little room to stand out performatively as VJ Anyone (Olivier Sorrentino) notes " ...if people stop dancing and just stand there with their jaws dropped, staring at your visuals and drooling, then you're also not doing your job. If the crowd is watching the screen as cinema, then they're not enjoying the rest of the experience, interacting with other people..."[46] While there are certainly audiovisualists attempting to challenge this notion, Spinrad cautions that "our expectations and habits around being audience members have atrophied ever since movies became popular. [They] taught us to sit together and pay attention to a dead, unchanging recording rather than something living and responsive."[62] As Henry Warwick, curator of the San Francisco Performance Cinema Symposium notes "VJ is doomed, so long as it's carried by the dance scene. To evolve, the form needs to break away..."[61] not only from the dance scene but also from the legacy of Cinematic and Visual Arts it so desperately clings to for justification. A reconsideration of performative context and appropriate venue for audiovisual performance is drastically overdue.

Experimental art communities remain the most tolerant of audiovisual work as they feature an inherently interactive construction of performative dialogues where the links between audiovisual performance and its applied heritage make the most sense. The strong link between organisations like Other Film and the experimental 'underground' music scene reiterate and expand upon models of artistic collectives that were often central to the initial development of expanded and experimental cinema. Localised experimental communities offer the opportunity to perform in a more appropriate venue than the dance club. Having a background in Architectural studies, Tim Gruchy [31] put his understanding of spatial design to work with the recontextualisation of space for audiovisual performance at the Recreational Art Theme parties in late 80s Australia. Outside of the usual art gallery / cinematheque environments, and in spite of public liability laws, there is a consistent drive to the use of smaller venues with projection setups (Glitch in Melbourne) reuse of abandoned cinemas (The Globe in Brisbane) and radical (and legally ambiguous) recontextualisation of any space from warehouses to car parks to city drains. Cindi Drennan has moved on from VJ work with Tesseract [21] to what she terms "Illuminart - installations, screen sculptures and structures featuring projection art for festivals, theatre, public spaces and corporate events."[20] The movement towards multiscreens and projection mapping follows a clear (albeit often neglected) lineage from expanded cinema works by the likes of Corinne and Arthur Cantrill in the 1970s [8] that emphasises an entrepreneurial use of space and the potential for an audiovisualist to expand from the boundaries of stereo sound and 4:3 vision.

3.4.The Activist

The role of the pioneer in the activation of new artistic forms can be seen as both scientific and entrepreneurial, and in many cases also political. As a noted auteur with a history of provocation, Peter Greenaway can afford to be outspoken in this regard:

"...my complaint is that now, after 108 years of activity, we have a cinema that is dull, familiar, predictable, hopelessly weighed down by old conventions and outworn verities, an archaic and heavily restricted system of distribution, and an out-of-date and cumbersome technology." [29]

His 'rant' highlights the fact that an activist might choose to use revolutionary techniques to start the revolution, not merely comment on it.

The organisation of sound and visual objects into a political narrative is central to audiovisual performance and evident from the early days of VJ. Reminiscent of media collagists Negativeland, there is a tendency towards textual juxtapositions that critique and entertain equally. While the "Natural Rhythms" series by Hexstatic/Coldcut provide chainsaw subtle montages of National Geographic and rainforest deforestation footage, their Panopticon AV work directly integrates sound bites and footage from the Undercurrents and Reclaim The Streets movements. [65] Linking political activism with digital media production serves to highlight one fundamental affordance of the democratisation of technology; a dismantling of the hegemony of meaning in favour of an independent multitude of voices.

Wade Marynowsky is a potent example of an audiovisual artist performing political texts that also challenge the nature of the audiovisual medium in performance. Frequently connecting sound to image processes, ‘Apocalypse Later’(1994) uses sound to destroy image (and vice versa) in an abstract work that also serves as a critique on the manipulated history of Australia's violent past. ‘Geek From Swampy Creek’(1997) is an amusing parody of the laptop performer where Marynowsky, portraying the "Geek" clad in a brown suit, coke-bottle glasses and sporting a megacephalic brain, sits at a laptop swaying distractedly. The granular emissions pixellate, mozaic and distort the visuals until they morph into an abstracted swamp. In contrast to many audiovisualists, Marynowsky presents himself as a performer at the centre of the audiovisual performance, both physically and by extension thematically within the context of the work. [48]

The image of the audiovisualist as pioneer also reaches into being an early adopter of technologies and the vision that particular technologies could carry with them attitudes that challenge dominant paradigms. NATO was one of the first AV program to present itself as a work of art beyond utilitarian concerns. Built from the basis of Ircam’s MAX the various builds arrive like mail-bombs wrapped in a situationist-style distribution scheme. With the central creator, Netochka Nezvanova being represented in public by several women, “[the] project presented itself as a sectarian cult, with its software as the object of worship."[13] Digital media artist Alexei Shulgin considers N.N. to be an elaborate performance; "a corporation posing as an artist, reciprocal to artists who had posed as corporations before.”[13] From NATO a genealogy of both open and proprietary tools emerged (Jitter / Gem / Processing) where the focus shifted from predefined tools for production to tools for the production of tools, instruments, interfaces and personalised artworks. As Netochka outlines "NATO.0+55 has altered the Max demographics by introducing new media artists, Internet artists, video and VR/3D artists to Max. In contrast to the largely aging + medicated male audience that Max has traditionally attracted, a significant number of NATO.0+55 operators are women” [52] This shift has almost certainly helped drive the integrated media approach and opened up avenues previously closed to a new practitioner demographic

3.5.Summary

At some point new ideas, new tools, new techniques, new aesthetics and new audiences amount to a new practice with a breadth of responsibility that perhaps highlights why musicians are mostly absent from a field increasingly dominated by programmers, visual artists, film-makers and graphic designers. Removed from the ability to just plug and play, the digital musician must actively devise new approaches to performance to avoid the trap of lazy visualisations that fail to inspire an audience and reinforce novelty stereotypes.

4. CONCLUSION

The emerging practice of the performing audiovisualist has numerous heritages, yet is currently lacking an established framework that does more than merely use convenient technology to repeat and remix other artistic developments. Mark Coniglio points out that "… because this technology is still relatively new, our problem is how can we include it in a piece and not make it about the technology?"[60] An understanding of the interlinking frameworks that outline an audiovisual performance work and the divergent approaches that define audiovisual performance can help us towards an appropriate mode and setting to express our integrated art in a progressive fashion. Writing about the cinematic form Greenaway notes that "the absolute strength of the medium is in its aesthetic, its relationship of language to content, its relevance to now, the ability to stimulate and entrance, provide stimulus to dream, legitimize imagination, set fire to possibilities, indicate what happens next... and I would say encourage wholehearted participation to the point of the panic of overexcitement."[29] With a clear understanding of the strengths of the medium, approaches to these integrative technologies and the nature of performer / audience interaction within this form, the sustainability of our art is no longer tied to assets from a previous era and the possibility that the performing audiovisualist can become a true expression of our time becomes a reality.

5. ACKNOWLEDGEMENTS

The work reported in this paper has been supported by the Australasian CRC for Interaction (ACID) through the Cooperative Research Centre Programme of the Australian Government's Department of Education, Science and Training.

6. REFERENCES

[1] Botborg "A Brief History of PhotoSonicNeuroKinaesthography" in Filter, Issue. 66, Summer 2007, ANAT: Adelaide, Sth Australia, pp.13-15

[2] Brandrud, M. D. The Knife: Silent Shout, An Audiovisual Experience, Rabid Records, 2006.

[3] Brown, A. R. "Tools and Outcomes: computer music systems and musical directions" from Imaginary Space: The Australasian Computer Music Conference. Wellington, The Australasian Computer Music Association. 1999, p. 16-22.

[4] Brown, A. R. "SoundCipher: A music and sound library for Processing", from International Computer Music Conference. Montreal, ICMA. 2009 (Forthcoming).

[5] Burt, Warren "Picking up the threads of a history more extensive than previously known: Percy Grainger's work with music technology" from Proceedings of the ACMC, Wellington, NZ, 1994, p. 14.

[6] Byrne, David. "I <3 Powerpoint" from http://www.davidbyrne.com/art/david_byrne_artbio.php#Lectures_Presentations, accessed 15/06/09.

[7] Camart. Ltd. "VJAMM 3" from http://www.vjamm.com/, accessed 15/06/09.

[8] Cantrill, Arthur & Corinne. "Arthur & Corinne Cantrill" in Innersense: Melbourne Independent Filmmakers compiled by Bob Mousoulis, from http://www.innersense.com.au/mif/cantrill.html, accessed 07/06/09

[9] Cates, Jon. "projected performances of Projection Performance" from http://blog.furtherfield.org/?q=node/279 accessed 28/04/2009

[10] Chance, Ian. "Sounds like a good idea - art and defence", in Australian Defence Science 10.1, DSTO Defence Science Communications: Canberra, Australia, 2002, pp. 6-7.

[11] Chion, Michel. Audio-Vision: Sound On Screen, Columbia University Press: New York, 1994, p. 136.

[12] Coniglio, Mark. "The importance of being interactive", in New Visions in Performance: The Impact of Digital Technologies, eds. Carver, Gavin and Beardon, Colin, Swets & Zeitlinger B.V., Lisse: The Netherlands, 2004, pp. 5-12

[13] Cramer, Florian. "Automatisms and their Constraints" in Words Made Flesh: Code, Culture, Imagination, Piet Zwart Institute: 2005, from http://pzwart.wdka.hro.nl/mdr/research/fcramer/wordsmadeflesh/07-chapter_4/view accessed 15/04/09

[14] Create Digital Media Ltd. "Create Digital Motion" from http://createdigitalmotion.com/, accessed 14/06/09

[15] Cycling '74. Max 5 from http://www.cycling74.com/products/max5, accessed 15/06/09.

[16] Daniels, Dieter. "Frédérick Kastner" in Media Art Net, 2009 from http://www.mediaartnet.org/works/pyrophon/ accessed 14/04/2009

[17] Dawson, J. "Dziga Vertov" in Senses Of Cinema, February, 2003, from http://archive.sensesofcinema.com/contents/directors/03/vertov.html, accessed 13/04/09.

[18] Dixon, S. "The Digital Double" in New Visions in Performance: The Impact of Digital Technologies, eds. Carver, Gavin and Beardon, Colin, Swets & Zeitlinger B.V., Lisse: The Netherlands, 2004, pp. 13-30

[19] Dixon, S. Digital Performance: A History Of New Media In Theater, Dance, Performance Art, And Installation. The MIT Press: Cambridge, MA. 2007, p. x, 5.

[20] Drennan, C. “Illuminart® : Audiovisual & Illuminated Arts”, from http://www.illuminart.com.au/, accessed 13/06/09.

[21] Drennan, C., Maynard, J. “<tesseract> research laboratories: Archive Site”, from http://www.trl.com.au/, accessed 13/06/09.

[22] Eno, Brian. “Dr Pangloss: A Living Art Reborn” in Prospect Iss.158, May 2009, from http://www.prospect-magazine.co.uk/article_details.php?id=10784 accessed 15/05/09

[23] Faulkner, Michael. VJ: Audio-Visual Art and VJ Culture, Laurence King Publishing, 2006.

[24] Fox, Robin. “Backscatter”, DVD, Synaesthesia: 2007

[25] Friel, Patrick. “Bruce McClure”, from http://www.whitelightcinema.com/McClure.html accessed 28/04/2009.

[26] Fry, Ben., Reas, Casey. Processing 1.0, 2009 from http://processing.org, accessed 15/06/09.

[27] Gharavi, Lance. “Of Both Worlds: Exploiting Rave Technologies in Caridad Svich’s Iphigenia.” in Theatre Topics 18.2, 2008, pp. 223-242

[28] Grau, Oliver. "Remember the Phantasmagoria! Illusion politics of the eighteenth century and its multimedial afterlife", in Media Arts Histories, ed. Oliver Grau, The MIT Press: Cambridge, MA, 2007, pp.142-145.

[29] Greenaway, P. "Toward a re-invention of cinema", 2003 from http://petergreenaway.org.uk/essay3.htm accessed 20/04/2009.

[30] Griffith, D. Al Jazari, 2008, from http://www.pawfal.org/dave/index.cgi?Projects/Al%20Jazari, accessed 29/04/2009.

[31] Gruchy, Tim. “Visual Musician & Media Artist”, April 2008 from http://www.grup.tv/tim.html, accessed 15/06/09.

[32] Healy, Sean. “A Brief History Of VJing in Australia” in Cyclic Defrost Issue 9, November 2004 from http://www.cyclicdefrost.com/article.php?article=735, accessed 06/06/09.

[33] Healy, Sean. “Skynoise” from http://www.skynoise.net/, 2009, accessed 05/06/09.

[34] Holtzman, S. R. Digital Mantras. MIT Press: Cambridge, MA, 1994, p. 239.

[35] Hooper, Greg. "Roving Concert, Sensory Exploits", in RealTime Arts 49, 2002, from http://www.realtimearts.net/article.php?id=9304 accessed 14/04/09

[36] Ichiraku, Y. “Last FM: Doravideo”, 2008, from http://www.last.fm/music/Doravideo, accessed 15/06/09.

[37] Ihlein, Lucas. “Line Describing A Cone” in Squatspace, 2005 from http://www.squatspace.com/cone/ accessed 20/04/2009

[38] Ikeda, Ryoji. “Formula”, DVD, Cornerhouse Publications: Manchester, 2005.

[39] Jorda, S., Kaltenbrunner, M., Geiger, G., and Bencina, R. "The reacTable", in Proceedings of the International Computer Music Conference, Barcelona, Spain, 2005.

[40] Kandinsky, Wassily. Concerning the spiritual in art (1911), translated by Sadler, Michael. T. H. and Gracyk, Theodore., 2002 from http://www.mnstate.edu/gracyk/courses/phil%20of%20art/kandinskytext.htm, accessed 12/04/2009

[41] Kinder, Marsha. "Designing a Database Cinema" in Future Cinema, eds. Shaw, Jeffery. and Weibel, Peter. The MIT Press: Cambridge, MA, 2003, pp. 348-349

[42] Kokoras, P. A. "Morphopoiesis: a general procedure for structuring form", 5th International Music Theory Conference, Vilnius, Lithuania, 2005.

[43] KPX Consultancy. “Chemical Brothers: Hey Boy Hey Girl” from http://kpxvideo.com/wp/chemical-brothers-hey-boy-hey-girl/, accessed 15/06/09.

[44] Leopoldseder, Hannes., Schöpf, Christine., Stocker, Gerfried. “Cyberarts 2008 International Compendium” from Prix Ars Electronica 2008, Hatje Cantz Verlag: Ostfildern, Germany.

[45] MacDonald, Adrian., Jones, Theresa., Fogle, Leone. "Visual Jazz: Rise of the VJ" in Onscreen magazine, 2008 from http://www.onscreenmag.com/feature-articles/visual-jazz/ accessed 14/04/2009

[46] Manovich, L. The Language of New Media, The MIT Press: Cambridge, MA, 2001, pp. 233, 238

[47] Manovich,L. "New media from borges to html", in The New Media Reader, eds. Wardrip-Fruin, N. and Montfort, N., MIT Press: Cambridge, MA, 2003, p. 18.

[48] Marynowsky, Wade “Interpretative dance: Experiments in real time audio visual performance 2002-2007”, DVD, Demux: 2008.

[49] McLuhan, M. Understanding Media: The extensions of man. Sphere Books: London, 1964.

[50] Moore, Dean., Lane, Simon. CTRL ALT SHIFT, 2009 from http://vimeo.com/4676745 accessed 06/06/09.

[51] Nemeth, Peter., Papp, Gabor., Samu, Bence. Animata, 2008 from http://animata.kibu.hu/ accessed 29/04/2009

[52] Nezvanova, Netochka "Nettime email response to salon.com article", 2002 from http://amsterdam.nettime.org/Lists-Archives/nettime-bold-0203/msg00078.html accessed 30/04/2009

[53] No TV "Tulse Luper VJ Performance Tour 2008-2009" from http://www.notv.com/index.php?option=com_content&task=view&id=12&Itemid=26, accessed 12/06/09

[54] Ratsi, Olivier. "Exyzt installation ripped off by 'The Killers'", 2008, from http://blog.antivj.com/2008/exyzt-installation-ripped-off-by-the-killers accessed 25/04/2009

[55] Ratsi, Olivier. "Live Painting - Shackleton", 2009, from http://blog.antivj.com/2009/live-painting-shackleton accessed 25/04/2009.

[56] Reggio, G. "The Qatsi Trilogy: 1972-2005" from http://www.koyaanisqatsi.org, accessed 15/06/09.

[57] Residents, The. "The Commercial Album", DVD, MUTE US: 2004

[58] Schippers, Huib. "Musical futures: Towards an ecology of musical diversity", 2009, QLD Conservatorium Research Centre

[59] Sorensen, A. "Impromptu: An interactive programming environment for composition and performance." from Australasian Computer Music Conference 2005. A. R. Brown and T. Opie. (Eds.) Brisbane, ACMA, p. 149-153.

[60] Spinrad, Paul. The VJ Book: Inspirations and Practical Advice for Live Visuals Performance, Feral House: Los Angeles, 2005, pp. 86-87, 110.

[61] Spinrad, Paul. (2009). "The Video Injected Hive Mind" in Boing Boing from http://www.boingboing.net/2009/02/24/the-videoinjected-hi.html, accessed 15/04/2009

[62] Toop, David. Haunted Weather. Serpents Tail: London, 2004, p.14

[63] U & I Software Metasynth, 2009, from http://www.uisoftware.com/MetaSynth/, accessed 15/06/09.

[64] Yahoo (2008). "mulch-discuss" from http://tech.groups.yahoo.com/group/mulch-discuss/ accessed 02/04/2009

[65] Yellow Peril. (1999). "Coldcut: Digital Pioneers" from http://www.snarl.org/texts/features/coldcut.htm accessed 29/04/2009

[66] Zilczer, J. "Music For The Eyes: Abstract Painting and Light Art" in Visual Music: Synaesthesia in Art and Music Since 1900, ed. Kerry Brougher, Thames & Hudson: London, 2005, p. 49

[67] Zuvela, D. "Stirrings in the Undergrowth: some notes on Live Audiovisual Art in Brisbane" in Eyeline #68, April 2009, QUT: Brisbane.

DO ANDROIDS DREAM OF MUSICAL CHIMERA?

Toby Gifford
Queensland University of Technology,
2 George Street, Brisbane, Australia

Andrew R. Brown
Queensland University of Technology,
2 George Street, Brisbane, Australia

ABSTRACT

This paper describes the use of the Chimera Architecture as the basis for a generative rhythmic improvisation system that is intended for use in ensemble contexts. This interactive software system learns in real time based on an audio input from live performers. The paper describes the components of the Chimera Architecture including a novel analysis engine that uses prediction to robustly assess the rhythmic salience of the input stream. Analytical results are stored in a hierarchical structure that includes multiple scenarios which allow abstracted and alternate interpretations of the current metrical context. The system draws upon this Chimera Architecture when generating a musical response. The generated rhythms are intended to have a particular ambiguity in relation to the music performance by other members of the ensemble. Ambiguity is controlled through alternate interpretations of the Chimera. We describe an implementation of the Chimera Architecture that focuses on rhythmic material, and present and discuss initial experimental results of the software system playing along with recordings of a live performance.

1. INTRODUCTION

"The Chimaera was a beast in Greek Mythology with the head of a lion, the body of a goat, and the tail of a serpent. We use the word Chimera metaphorically to refer to an image derived as a composition of other images. An example of an auditory Chimera would be a heard sentence that was created by the accidental composition of the voices of two persons who just happened to be speaking at the same time. Natural hearing tries to avoid chimeric percepts, but music often tries to create them. It may want the listener to accept the simultaneous roll of the drum, clash of the cymbal, and brief pulse of noise from the woodwinds as a single coherent event with its own striking emergent properties. The sound is chimeric in the sense that it does not belong to any single environmental object. To avoid Chimeras the auditory system utilizes the correlations that normally hold between acoustic components that derive from a single source and the independence that usually exists between the sensory effects of different sources. Frequently orchestration is called upon to oppose these tendencies and force the auditory system to create Chimeras" [1].

This work is a component of a broader research program into real-time improvisational computational agents, the ultimate goal being to create a 'robot' musician that can jam with human musicians playing live music on acoustic (or electronic) instruments. Conceptually this research program has been divided into a number of stages including signal processing to convert a raw audio stream into musical notes, analysis to convert the notes into some higher order musical representation, and generative processes for utilising the higher order representation to improvise appropriate musical output. Aspects of the signal processing have been reported elsewhere [2, 3], and a process for the generation of improvised rhythmic material, the Ambidrum, was outlined in [4]. We will similarly constrain our attention in this paper to unpitched rhythmic material for the purposes of clarity, but we expect that the Chimera Architecture can be generalised to include pitched material as well and will report on a more full implementation in future publications.

This paper is conceptually related to our earlier work on the Ambidrum system but extends it in a number of ways. The Ambidrum produced generative rhythms that contained a specifiable amount of 'metric ambiguity' given assumptions about the underlying metre. Although it was designed with an ensemble context in mind, at that stage of the research we had not implemented any machine listening algorithms and so the Ambidrum system did not 'listen'. Rather it introspectively used knowledge of its own output to ensure that in-and-of itself it produced appropriate rhythms.

The generative improvisational system described in this paper utilises a novel representational architecture dubbed the Chimera Architecture, which parses the musical surface into a collection of metric scenarios with associated confidences, and is able to 'listen' in an ensemble. The Ambidrum utilised a particular measure of metric ambiguity, which essentially measured how correlated the rhythmic variables (duration, dynamics, timbre) were with the actual metre, which was assumed to be known. The Chimera Architecture does not rely on the assumption of a known metre - rather it takes into account a distribution of metric scenarios that are contained within the Chimera. Like the Ambidrum, the generative system described here utilises metric ambiguity for musical effect.

2. BACKGROUND

There are a wide range of roles for the computer to play in cybernetic musical partnerships (Brown 1999, 2000). As we consider our research into a system for collaborative improvisation it is useful to clarify the requirements for such a systems and how they differ from other computer music tasks, particularly in relation to the degree of autonomy required of the computer system. As a tool, computers are used for tasks such as recording, publishing and communicating. When using the computer as a tool the system is expected to follow directions unambiguously and show little initiative. As a technical assistant, for example in computer assisted compositional systems, the computer is often asked to complete tasks or to generate algorithmic material. Although CAC systems are generative, they less commonly achieve this based on analysis, with some notable exceptions (Cope 1996) and rarely operate in real time. Computers have played roles as real-time performers, from simple audio playback systems to score following accompanists. Another real-time application for computers in music is as instruments. This often involves the generation of synthesized sounds in response to gestural input, but can also involve symbolic manipulation and generative processes, for example during live coding performances. The computational processes here are usually highly directed, but non-linear or stochastic processes are frequently used to provide interest or surprise for the performer/user. Finally, computers can be used as improvisational collaborators where their role is to generate, at times based on ongoing evaluation, appropriate musical material as part of an ensemble performance. Early examples of these improvisational systems included those by Chadabe [5], Biles [6] and Rowe [7] with more recent examples utilising beat tracking [8] and adaptive feedback [9]. Our work reported here extends this latter tradition that combines machine listening and improvisation in collaborative performance situations.

3. ANALYSIS

In order for the improvisation system to make an informed contribution to the collaborative performance it first needs to form an impression about the musical context in which it is operating. To do this the Chimera Architecture performs a real-time analysis on the audio input it receives.

3.1. Audio Analysis

The Chimera Architecture aims to provide representational information to enable real-time percussive accompaniment in an ensemble context. To achieve this it utilises an analysis process to transform raw audio input into an abstract representation that it is used in generating appropriate musical accompaniment. In the current implementation the analysis process perceives only percussive onsets, but is designed to be able to extract percussive onsets from a complex audio stream, for example a full live band performing, or from the playback of an audio recording (so that it might be used by DJ's in performance).

A machine-improvisation system that can demonstrate some autonomy needs to represent its musical context in some way. Our approach assumes that musical understanding is codified as abstract structures that can be reorganised to form the basis for novel generative elaboration. It has been shown that these abstractions can be based on statistical abstraction (Huron, 2006) and successfully applied in improvisation systems such as as OMax, whose developers concur that "musical patterns are not stored in memory as literal chains, but rather as compressed models, that may, upon reactivation develop into similar but not identical sequences" [10:126].

In the current implementation of our analysis process we utilise a two level hierarchy of reduction to represent the musical context. The two levels are referred to as (i) salience and (ii) scenario. As the system listens only for percussive onsets, the representation of the musical context is limited to metric/rhythmic analyses (melodic and harmonic contexts are not considered). The first level of reduction is used to create a timeline of rhythmically salient moments, which involves the isolation of significant features in the signal and a measure of their relative significance. The second level, scenario, postulates possible metric abstractions, such as tempo and metre, from analysis of the salient features. We expect that similar and deeper hierarchical structures may be both possible and helpful as ways of elaborating this approach, especially given that hierarchical structures are commonplace in theories of musical analysis [11-13] and in contemporary theories of mind [14].

There are numerous benefits to be gained from the use of hierarchical abstractions for machine improvisation, including robustness and stability over time and situation, and reductions in data storage and processing requirements.

3.2. Salience

The first stage of the Chimera Architecture's perceptual hierarchy is the reduction of the raw audio signal into a timeline measuring the musical salience at each point in time. This is a low-level measure of salience - it is not intended to represent higher order musical features such as downbeats, phrase endings, and so on. Rather it aims to parse the audio stream into something similar to an event-based representation of just the percussive component of the audio signal. This is achieved using three novel percussive onset detection algorithms.

The three detection algorithms are, roughly speaking, looking for percussive onsets in the high, mid and low frequency bands; for example when listening to a drum-kit the onset detection algorithms are tuned to discriminate between and kick-drum, a snare, and a hi-hat. The onset detection algorithms being used are not simply band-pass energy spike detectors. The following approaches are employed.

(i) The hi-hat detection algorithm is our SOD technique [3] for detecting 'noisy' percussive onsets.

(ii) The kick-drum detection algorithm utilises a novel time domain approach for detecting low-pitched percussive onsets.

(iii) The snare detection algorithm utilises a combination of the SOD technique and a traditional energy tracking technique.

All detection algorithms report on the amplitude of the onsets, which is interpreted as the salience at that point in time (the salience is zero at other times).

3.3. Scenario

The second tier of the Chimera Architecture's representation parses the salience into a metrical information in four steps.

(i) Firstly, it estimates the pulse.

(ii) Having estimated the pulse, it then estimates the number of beats in a bar.

(iii) Given the length of the bar, it then estimates which pulse is the downbeat; establishing the phase of metre.

(iv) Having estimated the downbeat, it accumulates the saliences of onsets that occur on each particular beat within the bar. The relative level of the salience at each bar position are interpreted as the metric weight of that beat position.

A given value for these variables (the pulse, number of beats in a bar, downbeat, and metric weights) is referred to as a *metric scenario*. A scenario represents the system's current understanding of the musical context; that is, the tempo, meter, and rhythmic density of the music it is listening to and performing with.

3.4. Chimera

Figure 1. A depiction of the Chimaera on an ancient Greek plate.

The analysis procedure from salience to scenario does not yield a single scenario, rather it yields a collection of plausible scenarios with associated confidences. This collection of scenarios we are calling a Chimera, referencing the beast of Greek mythology that Bregman [1] used as a metaphor for musical sounds that, whilst physically produced by of a number of disparate sources, are artistically combined to produce a new perceptual whole. In our case we are using the metaphor to refer to a combination of separate scenarios that may be combined to form a new hybrid scenario to artistic effect. This approach is inspired by psychological research that indicates that human musical perception maintains parallel, but not necessarily equal, interpretations and expectations during musical experiences.

During the first three steps of creating a Chimera the analysis yields a distribution of plausible values, and quantifies their plausibility as a confidence value between 0 and 1. The three steps are:

(i) From the salience is produced a collection of plausible pulses,

(ii) for each candidate pulse period, a collection of plausible bar lengths is produced,

(iii) for each candidate bar length, a collection of plausible downbeats is produced.

The confidences of the candidates at each stage are combined to calculate the confidence of each candidate scenario.

The Chimeric Architecture simultaneously tracks a collection of scenarios. To control the exponential explosion in the number of scenarios being tracked, we have limited the number of candidates at each stage of the analysis to two, so that at most there are 2*2*2 = 8 scenarios tracked at any one time. The architecture has 8 slots, that are filled with the most plausible scenarios. As time goes on, the confidence values of scenarios are updated. If a scenario becomes implausible, it is dropped from consideration, and if a new scenario is substantially more plausible than an existing scenario, the new scenario ousts the weakling. The Chimera's structure can be visualised as a tree:

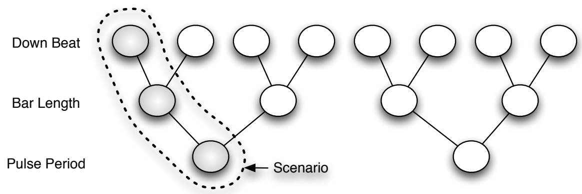


Figure 2. A Chimera data structure with one Scenario highlighted.

4. INTERPRETATION

The Chimera Architecture includes methods for reading, or interpreting, the Chimera data in a variety of ways when passing the scenario values to the generative processes that create improvisational content. The methods of interpreting data correspond to the degree of ambiguity in the generated content with respect to matching the performed input. This structure has been arrived at after careful consideration of theories relating to psychological studies of musical perception and expectation which we will now briefly outline.

4.1. Expectation

The importance of taking account of the dynamic nature of musical expectations when considering musical experience, either analysis of it (theory building) - or simulation of it (algorithmic composition/improvisation) has received attention in the music theory literature for some time [11, 15-17] but has only recently been translated into computational descriptions and rarely been the basis for algorithmic music systems. Meyer suggests that affect in music perception can be largely attributed to the formation and subsequent fulfilment or violation of expectations. His exposition is compelling but imprecise as to the exact nature of musical expectations and to the mechanisms of their formation.

A number of extensions to Meyer's theory have been proposed, which have in common the postulation of at least two separate types of expectations; structural expectations of the type considered by Meyer, and additionally dynamic expectations. Narmour's [16] theory of Implication and Realisation, an extension of Meyer's work, posits two cognitive modes; one of a schematic type, and one of a more innate expectancy type. Bharucha [17] also discriminates between schematic expectations (expectations derived from exposure to a musical culture) and veridical expectations (expectations formed on the basis of knowledge of a particular piece). Huron [18] has recently published an extensive and detailed model of musical expectations that builds further on this work. He argues that there are, in fact, a number of different types of expectations involved in music perception, and that indeed the interplay between these expectations is an important aspect of the affective power of the music. Huron extends Bharucha's categorisation of schematic and veridical expectations, and in particular makes the distinction between schematic and dynamic expectations. Dynamic expectations are constantly learned from the local context. Several authors have suggested that these dynamic expectations may be represented as statistical inferences formed from the immediate past [18-20]. Like Bharucha, Huron argues that the interplay of these expectancies is an integral part of the musical experience.

4.2. Metre as an Expectational Framework

Musical metre is frequently described as the pattern of strong and weak beats in a musical stream. From the point of view of music psychology, metre is understood as a perceptual construct, in contrast to rhythm, which is a phenomenal pattern of accents in the musical surface. Metre is inferred from the surface rhythms, and possesses a kind of perceptual inertia. In other words, once established in the mind, a metrical context tends to persist even when it conflicts with the rhythmic surface, until the conflicts become too great - "Once a clear metrical pattern has been established, the listener renounces it only in the face of strongly contradicting evidence" [11:17]. Jones [21] argues that metre should be construed as an expectational framework for predicting when salient musical events are expected to happen. This description of metre has been widely accepted within the music psychology community [18, 22, 23].

4.3. Ambiguity

Meyer [15] identifies ambiguity as a mechanism by which expectations may be exploited for artistic effect. In this context ambiguity is referring to musical surfaces that create several disparate expectations. The level of ambiguity in the music creates a cycle of tension and release, which forms an important part of the listening experience in Meyer's theory. An ambiguous situation creates tension; the resolution of which is part of the art of composition. "Ambiguity is important because it gives rise to particularly strong tensions and powerful expectations. For the human mind, ever searching for the certainty and control which comes with the ability to envisage and predict, avoids and abhors such doubtful and confused states and expects subsequent clarification" [15:27]. Temperley notes that ambiguity can arise as the result of multiple plausible analyses of the musical surface. "Some moments in music are clearly ambiguous, offering two or perhaps several analyses that all seem plausible and perceptually valid. These two aspects of music - diachronic processing and ambiguity - are essential to musical experience" [24:205].

4.4. Multiple Parallel Analyses

A number of researchers have posited systems of musical analysis that yield several plausible results as models of human musical cognition. Notably, Jackendoff [25:140] proposed the parallel multiple analysis model. This model, which was motivated by models of how humans parse speech, claims that at any one time a human listening to music will keep track of a number of plausible analyses in parallel. In a similar vein, Huron [18] describes the competing concurrent representation theory. Huron goes further to claim that, more than just a model of music cognition, "Competing concurrent representations may be the norm in mental functioning" [18:108].

4.5. Ambiguity in Multiple Parallel Representations

An analysis system that affords multiple interpretations provides a natural mechanism for the generation of ambiguity. In discussing their Generative Theory of Tonal Music (GTTM), Lerdahl & Jackendoff [11:42] observe that their "rules establish not inflexible decisions about structure, but relative preferences among a number of logically possible analyses", and that this gives rise to ambiguity. In saying this Lerdahl & Jackendoff are not explicitly referencing a cognitive model of multiple-parallel-analyses; the GTTM predates Jackendoff's construction of this model, and does not consider real-time cognition processes. Indeed it was considerations of the cognitive constraints involved in resolving the ambiguities of multiple interpretations that led Jackendoff to conclude that the mind must be processing multiple analyses in parallel [25].

Temperley has revisited the preference rule approach to musical analyses in a multiple parallel analyses model: "The preference rule approach [is] well suited to the description of ambiguity. Informally speaking, an ambiguous situation is one in which, on balance, the preference rules do not express a strong preference for one analysis over another ... At any moment, the system has a set of "best-so-far" analyses, the analysis with the higher score being the preferred one. In some cases, there may be a single analysis whose score is far above all others; in other cases, one or more analyses may be roughly equal in score. The latter situation represents synchronic ambiguity" [24:219]. In a similar spirit, Huron [18:109] argues that multiple-parallel-analyses (or competing concurrent representation, as he calls them) must all be generating expectations, and consequently must give rise to the kind of expectational ambiguity that was argued above to play a central role in producing musical affect. This is the model of ambiguity that we have adopted in the Chimera Architecture.

4.6. Interpreting the Chimera

The Chimera Architecture tracks multiple plausible scenarios. How can we use this information to create generative improvisations? The central hypothesis of this paper is that utilising all of the parallel scenarios for generative improvisation can be musically efficacious. Let us however consider other approaches. One obvious possibility is to select the most plausible scenario and generate material which is appropriate to this scenario. This approach is similar to that used by authors who have considered multiple parallel analyses models for musical analysis—the explicit assumption being that despite tracking multiple analyses, at any one time there is only one analysis which is perceived as being 'correct'. For example, Temperley (above) refers to the preferred analysis as the one with the highest score. Similarly the GTTM explicitly insists that only one analysis at a time can be 'heard', as they make clear; "Our hypothesis is that one hears a musical surface in terms of that analysis (or those analyses) that represent the highest degree of overall preference" [11].

A similar hypothesis is widely held in a number of fields of psychology and neuroscience, under a variety of names. The Gestalt Psychologists refer to it as the Figure-Ground dichotomy [26], and in neuroscience it is called the Winner-Takes-All hypothesis [27:494]. The idea is that the mind can only be conscious of a single reality at any one time.

In music perception a number of authors have similar commented on the impossibility of consciously perceiving more than one musical analysis simultaneously: "It is true that we are conscious of only one analysis at a time, or that we can attend to only one analysis at a time. But this leaves open the possibility that other analyses are present unconsciously, inaccessible to attention" [11:141]. London, while discussing music psychology experiments on attending to metre by tapping along to a single stream of a polyrhythm notes that, "These studies ... indicate that while on any given presentation we tend to hear a passage under one and only one metric framework, it is possible to re-construe the same figure or passage under a different meter on another listening occasion. A polyrhythmic pattern may be heard "in three" or "in four," just as metrically malleable patterns may be set in different metric contexts. This should not surprise anyone familiar with the basic tenets of perception, as the need to maintain a single coherent ground seems to be universal ... Thus there is no such thing as polymetre" [23:50].

However, our research is at odds with this view, and we contrastingly suggest that all of the scenarios present in the Chimera may be used to artistic effect. Support for our multiple scenario approach can be found in Huron's [18:109] discussion of his theory of competing concurrent representations. He argues that each of the concurrent representations must all be creating expectations and so must all be contributing to the musical affect. Music psychology experiments (such as those above) that suggest that only one scenario can be consciously attended to at a time, may not be relevant to the acts of listening to, or improvising with, music since these activities do not necessarily involve conscious attention to musical representations.

More precisely, we propose that by interpreting the Chimera in different ways we may achieve different musical results, particularly in regard to the level of ambiguity present in the improvised generative material. For example, utilising a single scenario interpretation in which material is generated as appropriate to only the most plausible scenario should result in a decrease in the overall ambiguity of the ensemble's playing, since highlighting the most prominent scenario is likely to have the effect of further increasing its relative plausibility. Conversely, generating material that is equally appropriate to two scenarios regardless of their relative plausibility should increase the level of ambiguity present in the ensemble. We suggest three different strategies for interpreting the Chimera for the purposes of producing generative accompaniment, revolving around controlling the level of ambiguity present.

(i) Disambiguation - utilise only the most plausible of the scenarios in the Chimera.

(ii) Ambiguation - utilise all of the scenarios with equal weight, regardless of their plausibility.

(iii) Following - utilise the scenarios in the Chimera with weight according the their plausibility.

5. IMPLEMENTATION

We have implemented the Chimera Architecture in C++ on the Mac OS X platform as an Audio Unit. The Audio Unit receives audio input, and performs the analysis into the representation described. The generative improvisational processes (described below) were implemented in the Impromptu environment [28] which also acts as a host for our Audio Unit. Parameters defining the Chimera are stored as Audio Unit parameters, allowing the generative processes to query the Chimera at any time. The software system design is shown in figure 3.

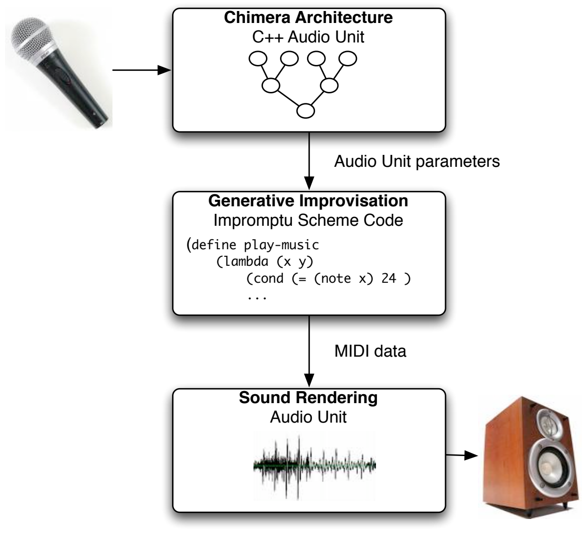


Figure 3. The schematic design of the system implementation.

5.1. Generative Improvisational Processes

At this stage of the research we have implemented only a very simple generative process, designed to clearly demonstrate the musical results of the three interpretative strategies outlined above. For any given metric scenario, we want to create rhythmic material that is appropriate to, or stereotypical of, the scenario. To achieve this we simply play a percussive attack on each beat identified in the scenario, with an accent on the downbeat. The downbeat accent is created by utilising a different timbre (i.e., a different percussive instrument), whilst the other beats have the same timbre, but varying dynamics. The dynamics are chosen to correspond to the metric weights of the beats specified by the scenario.

The dynamics of the attacks on the beats are further modulated by the desired weighting of the scenario in the generated material. So, for example, if we are using the interpretative strategy for Ambiguation discussed above, the pulse streams corresponding to the different scenarios will be equally loud (on average), whereas if we are using the Following interpretative strategy then the pulse stream corresponding to the most plausible scenario will be loudest.

5.2. Experimental Results

We have provided examples of the system improvising rhythmic accompaniment to a short recorded loop of live drums to demonstrate its operation. The audio files discussed may be downloaded from http://explodingart.com/giffordbrown2009/. The original loop (MR10.aif) is a sample rock beat played on a standard drum kit. We perceived it as being in a 4/4 time signature at 110 bpm. The rhythm has regular hi-hats played on the quavers, with snare hits on the backbeat, and a triplet-feel kick-drum pattern. The kick-drum pattern contrasted with the snare and the hi-hat gives rise to metric ambiguity, as the triplet feel is metrically dissonant with the straight four/eight feel of the snare and hi-hat.

The results of the first stage of analysis, the parsing of the musical surface into salience streams for the kick-drum, snare and hi-hat, is pictured below. The spikes in salience correspond to detected onsets.

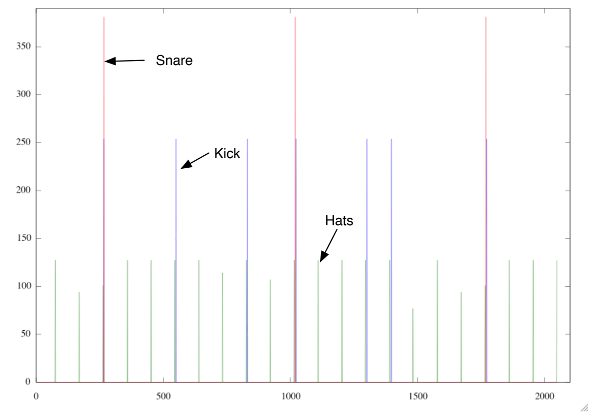


Figure 4. Salience streams for the original live music file MR10.aif

The second stage of analysis, the conversion of the salience streams into metric scenarios, yielded four plausible scenarios.

(a) The most plausible scenario has a pulse period of 0.27 seconds (~ 220 bpm), and 8 beats to the bar.

(b) The second most plausible has the same bar period as the first, but counts only 4 beats of 110 bpm.

(c) The third has 5 beats at 220 bpm

(d) The fourth has 3 beats at 220 bpm.

The Chimera (the collection of these four scenarios together with their relative plausibilities) changes through time, most notably change occurs in the confidence of scenarios with 5 and/or 3 beats to a bar. The top two scenarios (8 and 4 beats to the bar) are stable, whilst the others pop in and out of the Chimera as the system updates their relative plausibilities. An example print-out of the state of the Chimera data is given in Figure 5. In the print-out scenarios are labelled 'context dumps'. The letters S, B and H indicate the pulse period, bar period and number of beats in a bar respectively (periods are measured as a number of analysis windows, where an analysis window is 128 samples @ 44.1khz). Notice that the Chimera briefly entertains the notion of a bar of 10 before pruning it out when its plausibility becomes too low.

```
--------------------- Context Dump ----------------------
Context #0
S = 94 B = 282 H = 3 LDB = 6.79617 confidence = 0.136748
Context #1
S = 188 B = 752 H = 4 LDB = 2.25089 confidence = 0.224367
Context #2
S = 94 B = 751 H = 8 LDB = 2.62708 confidence = 0.357491
Context #3
S = 94 B = 939 H = 10 LDB = 2.50199 confidence = 0.1498
---------------------------------------------------------


*********** Pruning *********
S = 94 B = 939 H = 10 LDB = 2.5025 confidence = 0.0261448

--------------------- Context Dump ----------------------
Context #0
S = 94 B = 282 H = 3 LDB = 6.79757 confidence = 0.174113
Context #1
S = 188 B = 751 H = 4 LDB = 3.174 confidence = 0.353134
Context #2
S = 94 B = 751 H = 8 LDB = 2.62762 confidence = 0.553314
---------------------------------------------------------


--------------------- Context Dump ----------------------
Context #0
S = 94 B = 282 H = 3 LDB = 10.4628 confidence = 0.220562
Context #1
S = 188 B = 751 H = 4 LDB = 3.17417 confidence = 0.302642
Context #2
S = 94 B = 751 H = 8 LDB = 3.67601 confidence = 0.429445
---------------------------------------------------------
```

Figure 5. Data from three updates of the scenario values.

Audio examples from the system generating material utilising the three interpretative strategies of disambiguation, ambiguation, and maintenance are given in MR10_click_disambiguate.aif, MR10_click_ambiguate.aif, and MR10_click_maintain.aif respectively. These examples have been transcribed as common practice notation below. From these examples, the alternative scenarios, especially relating to likely metres, is clear to see. The improvisation generated by this simple reflection of the analysis data indicates that the richness of material (multiple parts and poly metre) that arises from considering several scenarios (the Chimera) provides opportunities for much more interesting material from which generative processes can be derived. What is not so evident in the notated examples in figures 6-8 is that the relative dynamic levels are mapped to the plausibility of each pulse such that the less likely beats are downplayed in the audio mix providing an automatic depth and subtlety otherwise unavailable.

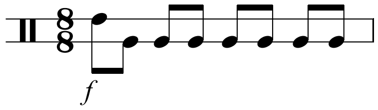


Figure 6. The generated 'disambiguated' music

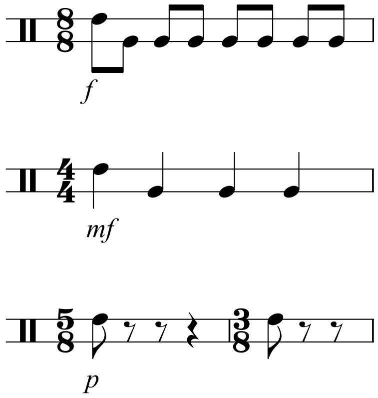


Figure 7. The generated 'following' music

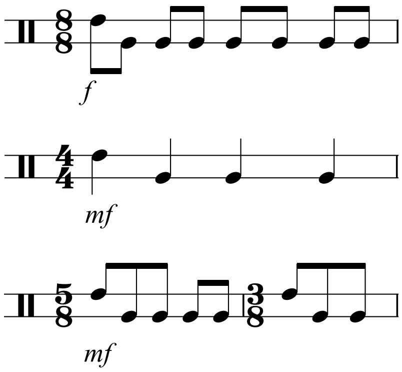


Figure 8. The generated 'ambiguated' music

6. CONCLUSION

In this paper we have reported on the Chimera Architecture and its use in generative machine improvisation that accompanies human performance. The implementation has focused on unpitched rhythmic music presented to the system as an audio signal. From this audio input the system performs a novel metric analysis that locates the pulse duration, tempo, bar length and down beat position, as well as metric weights at each beat of the bar. The system introduces an innovative approach in its maintenance of multiple analytical outcomes, described as scenarios that, with their associated plausibility weightings, we have called a Chimera and which is stored in a binary tree data structure. This approach provides flexibility for the system to interpret the analytical material in a variety of ways in order to inform its generative improvisational output. We have suggested that there is a connection between the interpretation of the Chimera and the level of rhythmic ambiguity in the generated material. When the most likely analytical scenario is used for generation, the music can reinforce the assumed metrical intent of the human musician; we labelled this a disambiguated response. When the scenarios in the Chimera are interpreted as a weighted average to inform the generative process, the musical result tends to maintain or reinforce the existing level of metrical ambiguity. When the scenarios in the Chimera are interpreted as being equally weighted the generated material tends to be more metrically ambiguous than the performed input.

Our in initial experimental results provide confidence in the ability for the Chimera Architecture to accurately track human performance via an audio stream, for the Chimera data representation to maintain several plausible interpretations of the metrical characteristics of the performance, and for different Chimera interpretations to enable a useful variety of appropriate musical responses.

In the future we plan to extend the implementation of the Chimera Architecture to incorporate pitched material that will inform the harmonic and melodic aspects of the systems generated improvisations. We also plan to test more throughly the correlation between human improvisational decisions and generative interpretations of the Chimera.

7. REFERENCES

[1] Bregman, A.S., *Auditory Scene Analysis: The perceptual organization of sound*. 1990, Cambridge, MA: The MIT Press.

[2] Gifford, T. and A.R. Brown. *Polyphonic Listening: Real-time accompaniment of polyphonic audio*. in *Trans: Boundaries / Permeability / Reification. Australasian Computer Music Conference '07*. 2007. Canberra, Australia: ACMA.

[3] Gifford, T. and A.R. Brown. *Stochastic Onset Detection: an approach to detecting percussive attacks in complex audio*. in *Sound : Space - The Australasian Computer Music Conference*. 2008. Sydney: ACMA.

[4] Gifford, T. and A.R. Brown. *The Ambidrum: Automated Rhythmic Improvisation*. in *Medi(t)ations: Australasian Computer Music Conference*. 2006. Adelaide: ACMA.

[5] Zicarelli, D., *M and Jam Factory*. Computer Music Journal, 1987. **11**(4): p. 13-29.

[6] Biles, J.A. *GenJam: A genetic algorithm for generating jazz solos*. in *International Computer Music Conference*. 1994. San Francisco: ICMA.

[7] Rowe, R., *Interactive Music Systems: Machine listening and composing*. 1993, Cambridge, MA: MIT Press.

[8] Collins, N., *Algorithmic Composition Methods for Breakbeat Science*, in *Music Without Walls*. 2001, De Montfort University: Leicester.

[9] Cont, A., S. Dubnov, and G. Assayag, *Anticipatory Model of Musical Style Imitation Using Collaborative and Competitive Reinforcement Learning*, in *Anticipatory Behavior in Adaptive Learning Systems*. 2007, Springer: Heidelberg. p. 285-306.

[10] Assayag, G., et al., *Omax Brothers: A Dynamic Topology of Agents for Improvization Learning*, in *ACM Workshop on Audio and Music Computing for Multimedia, International Multimedia Conference*. 2006, ACM: Sanata Barbara. p. 125-132.

[11] Lerdahl, F. and R. Jackendoff, *A Generative Theory of Tonal Music*. 1983, Cambridge, MA: MIT Press.

[12] Schenker, H., *Harmony*. 1980, Chicago: University Of Chicago Press.

[13] Cope, D., *Virtual Music: Computer synthesis of musical style*. 2001, Cambridge, MA: MIT Press.

[14] Hawkins, J. and S. Blakeslee, *On Intelligence*. 2004, New York: Times Books.

[15] Meyer, L.B., *Emotion and meaning in music*. 1956, Chicago: The University of Chicago Press.

[16] Narmour, E., *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. 1992, Chicago: University of Chicago Press.

[17] Bharucha, J.J., *MUSCAT: A connectionist model of musical harmony*, in *Machine Models of Music*, S. Schwanauer and D. Levitt, Editors. 1993, MIT Press: Cambridge, MA. p. 497-509.

[18] Huron, D., *Sweet Anticipation: Music and the psychology of expectation*. 2006, Cambridge, MA: The MIT Press.

[19] Temperley, D., *Music and Probability*. 2007, Cambridge, MA: The MIT Press.

[20] Pearse, M. and G. Wiggins, *Expectation in Melody: The Influence of Context and Learning.* Music Perception, 2006. **23**(5): p. 377-405.

[21] Jones, M.R., *Dynamic patterns structure in music: Recent theory and research.* Perception & Psychophysics, 1987. **41**: p. 621-634.

[22] Large, E.W., *Dynamic Representation Of Musical Structure*. 1994, Ohio State University: Ohio.

[23] London, J., *Hearing in Time: Psychological Aspects of Musical Meter*. 2004, Oxford: Oxford University Press.

[24] Temperley, D., *The Cognition of Basic Musical Structures*. 2001, Cambridge, MA: The MIT Press.

[25] Jackendoff, R., *Languages of the mind: Essays on mental representation*. 1992, Cambridge, Mass. ; London: MIT Press. ix, 200p : ill : music ; 24cm.

[26] Boring, E., *Sensation and Perception in the History of Experimental Psychology*. 1942, New York: Appelton-Century.

[27] Kandel, E.R., J.H. Schwartz, and T.M. Jessell, *Principles of neural science*. 2000, New York: McGraw-Hill Professional.

[28] Sorensen, A., *Impromptu: An interactive programming environment for composition and performance*, in *Australasian Computer Music Conference 2005*, A.R. Brown and T. Opie, Editors. 2005, ACMA: Brisbane. p. 149-153.

TRASE ALGORITHM - AUTOMATIC EVOLUTIONARY MORPHING OF ELECTRONIC DANCE MUSIC

Rene Wooller
Creative Industries
Queensland University of Technology

Andrew Brown
Creative Industries
Queensland University of Technology

ABSTRACT

The *TraSe* (Transform – Select) algorithm has been developed to investigate the morphing of electronic music through automatically applying a series of deterministic compositional transformations to the source, guided towards a target by similarity metrics. This is in contrast to other morphing techniques such as interpolation of parameters or probabilistic variation. *TraSe* allows control over stylistic elements of the music through user-defined weighting of numerous compositional transformations. The formal evaluation of *TraSe* was mostly qualitative and occurred through nine participants completing an online questionnaire. The music generated by *TraSe* was generally felt to be less coherent than a human composed benchmark but in some cases judged as more creative.

1. INTRODUCTION

As with other morphing algorithms (Mathews and Rosler, 1969; Polansky 1987), *TraSe* operates on two note sequences – *source* and *target*, and produces a *morph,* which can serve as a musical hybrid and/or transition between the two. This paper outlines some of the overall process, the transformations used and the results of a formal questionnaire.

TraSe is an evolutionary music algorithm (Biles 2008) in that the source is mutated into a pool of 'offspring' from which only one is selected for further mutation. The only other previously known significant evolutionary algorithm applied to note sequence morphing I have found is described by Horner and Goldberg (1991). However, this differed in the range of compositional transformations that were applied, the way the transformations were applied and lack any accompanying music (morph examples shall be provided on ACMC CD) or formal analysis. For those interested in other, non-evolutionary works within the area of note sequence morphing, an extensive review is available (Wooller 2006).

2. OVERVIEW OF TRANSFORMATION SELECTION PROCESS

TraSe takes two note sequences, source, S, and target, T, and produces morphed material, M. The morph, M is a list of note sequence loops called **frames**, the first and last of which is S and T respectively. If we let n be the length of the frame list, $M_1 = S$ and $M_n = T$. The frames of the middle $M_2, M_{...}, M_{n-1}$ constitute a sequential progression.

During playback, the morph index, Ω, determines which frame in the series is playing. Letting i be the index of any frame, ranging from $1 \leq i \leq n$, and noting that Ω is normalized between 0 and 1, M_i will hold the note sequence that is looped when $\Omega = i/n$. Logically, the morph will be **smooth** if each frame is somehow made similar to its neighbours. Smoothness is the quality of continuity, whereby each segment is perceived as being more similar to the points directly ahead and behind than the others which are further away.

The musical representation includes loop lengths, note onsets, durations, dynamics, scale degrees/passing notes, (rather than absolute pitch) scale, key and octave**.** There are two separate *TraSe* algorithms that operate in parallel – one for key and scale and another for note-level data (note onset, duration, dynamic and scale degree/passing note). I refer to the former as a *key/scale morph* and the latter as a *note morph*. Both the note morph and key/scale morph have a separate M which can be of different lengths. *TraSe* processes described in this paper are for the note morph.

During playback, the current frame of the note morph is overlayed with the key/scale morph to produce MIDI note output (pitch, duration, onset and dynamic). Having the key/scale and note morphs separate affords more specific control over the tonal elements of the morph.

If S and T are different lengths, each frame will have a length equal to the lowest common multiple. For example, if S was 6 beats and T was 4, the loops in each frame of M would be 12 beats long. Requiring the frames to be the same length is practical; a uniform length affords automated comparisons between sequences, due to their notes occupying an identical space. It also affords the combination of material.

Recall that i is an index to M. *TraSe* sequentially fills each frame of M, starting with $i = 2$, and finishing with $i = n$. Each frame, M_i, is created by passing the previous frame, M_{i-1}, through a chain of compositional transformation functions: let this be the list C. The functions in the chain C perform musical transformations that a composer might attempt, for example *harmonise* creates or removes harmonies at particular intervals, while *rate* speeds up or slows down the music by particular commonly used ratios, looping when necessary.

To generate the frame M_i, the first transformation, C_1, is passed the previous frame, M_{i-1}. The second transformation is passed the output of the first, and this continues until the last transformation. The last transformation is fed the output of the

second last transformation to create M_i. Letting m equal the length of C, or $m = |C|$, this means: $M_i = C_m(C_{m-1}(C_{...}(C_1(M_{i-1}))))$. *TraSe* stops generating frames when the current frame is judged as being similar enough to the target, T, by a variable for 'cut-off' that is specified by the user. Each frame is part of a sequence of frames that is the morph: $M_1, \ldots, M_i, \ldots, M_n$.
The evolutionary component of *TraSe* is in the fact that each transformation produces a range of different sequences from which only one is selected, through a comparison with the target sequence. The transformations are 'mutations' and the comparison to the target is a 'fitness function'. Allowing specific compositional transformations and dissimilarity measures to be designed and plugged into the algorithm enables elements of compositional style to be specified.

3. TRANSFORMING AND SELECTING

Each transformation has a set of possible parameter configurations. A pool of candidate note sequences are created using each possible parameter configuration. A single candidate is then selected, using a fitness function that compares the candidate note sequences with the target note sequence.

Recall that the number of transformations is m, that is $m = |C|$. Let j be an index into the list of transformations, C, ranging over $1 \leq j \leq m$. For C_j, there are a related set of parameter configurations, P_j. The parameter configurations within P_j are used by C_j to create transformed note sequences within a 'selection pool'; let it be O_j. The 0^{th} item in each parameter set is "bypass", $P_{j,0} = bypass$, thus enabling the unmodified input to also be an item in the pool.

For example, the *rate* transformation multiplies the start-time of each note by a ratio from the set $\frac{1}{4}, \frac{1}{2}, \frac{2}{3}, \frac{3}{2}, 2$ and 4. If *rate* is the 1^{st} transformation, that is, $C_1 = rate$, the corresponding parameter set is: $P_1 = \{1, \frac{1}{4}, \frac{1}{2}, \frac{2}{3}, \frac{3}{2}, 2, 4\}$. While *rate* has a single parameter, more complex transformations include particular combinations of multiple parameters. Having a fixed set of values may seem like an unnecessary limitation when dealing with parameters that are otherwise continuous; however, it also affords precise control over parameter values and thus musical style. As well as this, the parameter sets can be any size and requiring discrete specification encourages careful consideration of the musical ramifications of each parameter value.

Each transformation C_j has an input note sequence; let it be I_j. Let k be an index to parameters in the set P_j and the corresponding output in O_j, that is, $0 \leq k \leq |P_j|$. Thus, the creation of the k^{th} note sequence in the pool for the j^{th} transformation is: $O_{j,k} = C_j(I_j, P_{j,k})$. Continuing the *rate* example, where $j = 1$, creating the 1^{st} note sequence in the selection pool ($k = 1$) would be: $O_{1,1} = C_1(I_1, P_{1,1}) = C_1(I_1, \frac{1}{4})$. In this example, the start time for each note would be multiplied by $\frac{1}{4}$ and the result looped four times to make it the same length as the input. The diagram below provides an example of the complete selection pool for this *rate* example:

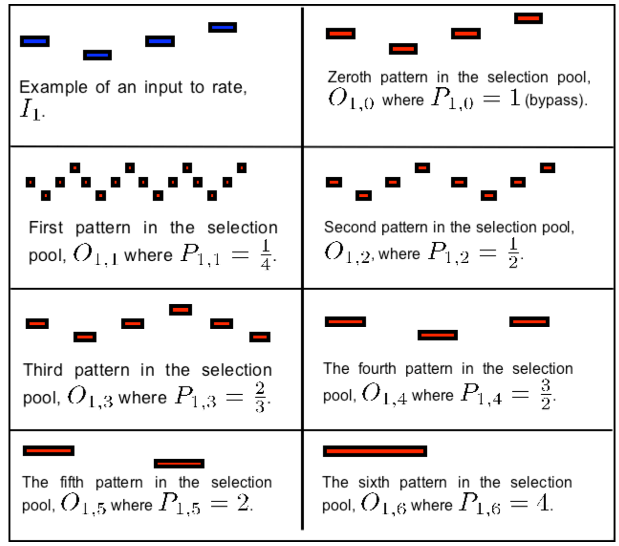


Figure 1 Each of the seven note sequences that would be in a selection pool for ***rate*** (j=1), given the original (top-left).

A fitness function determines which note sequence in the selection pool, O_j, is selected for output. Let the index of this selected note sequence in O_j be denoted by k^*. The input to the first transformation, C_1, is the previous frame (in the first cycle, this will be the source). Recalling that n is the number of frames and that $1 \leq i \leq n$, this means: $I_1 = M_{i-1}$. For transformations other than the first in the chain, the input is the selected output of the previous transformation in the chain, that is, for $2 \leq j \leq m$, $I_j = O_{j-1,k^*}$. Each frame other than the first is derived from the selected output of the last compositional transformation in the chain, that is, for $2 \leq i \leq n$, $C_m(C_{m-1}(C_{...}(C_1(M_{i-1})))) = O_{m,k^*} = M_i$. The first frame is the source music: $M_1 = S$.

4. SELECTION THROUGH DISSIMILARITY MEASURES

Each element in the selection pool is compared with T using a dissimilarity measure[1]. The selected note sequence, O_{j,k^*}, has a dissimilarity rating that is the closest to a particular target value. The process for determining the target value of dissimilarity with T is described further below. The application of user-defined weightings and "convergence" settings are also involved.

The dissimilarity measure for a particular transformation C_j,

[1] Discussing the measures in terms of 'dissimilarity' rather than similarity is an arbitrary choice, however, it is useful as 'dissimilarity' is analogous to 'distance', which is a natural method of comparing two values. The dissimilarity measure is not a distance metric in the formal mathematical sense, because triangle inequality is not necessarily upheld.

denoted by D_j, is designed to complement that transformation. For example, if $C_1 = rate$, D_1 calculates dissimilarities between the inter-onset envelopes, which are directly affected by the *rate* transformation. Each dissimilarity measure takes two note sequences and returns a value between 0 (similar) and 1 (dissimilar).

Each dissimilarity measurement between elements in the selection pool and the target is factored by a user controlled weight. The weights allow various parameter configurations to be unnaturally favoured which means the style of compositional transformations used throughout the morph can be influenced by the user.

Let w hold the user-defined weights. Recall that k is an index to the items in the parameter set P_j and the corresponding selection pool O_j. Let each item in the selection pool $O_{j,k}$ have a related weight, $w_{j,k}$. Let $R_{j,k}$ hold the result of the dissimilarity between $O_{j,k}$ and T, scaled by $w_{j,k}$. This means: $R_{j,k} = w_{j,k}\ D_j(O_{j,k}, T)$. Each measure in the set of weighted results R_j can be sorted from lowest to highest. Let the result of this sorting be $\overrightarrow{R}_j$. Thus, within this list, $k = 0$ refers to the smallest value (least dissimilar), and $k = |R_j|$ refers to the largest (most dissimilar). The selection pool can be arranged in the same order, so that $\overrightarrow{O}_{j,0}$ refers to the $O_{j,k}$ that minimises $w_{j,k}\ D_j(O_{j,k}, T)$.

5. SPEEDING THROUGH THE SCENIC ROUTE

If the goal of *TraSe* was to converge with T in as few frames as possible, then it would make sense to not use weights, and always choose $\overrightarrow{O}_{j,0}$, however, this is not the case. The aim instead is to find a musically interesting progression and this involves taking the "scenic" route. The **transform speed** is a user-defined parameter, s_j, which influences the extent to which the number of frames, n, is minimised, for each of the transformations in the chain. s_j is between 0, for the maximised or "slowest" setting, and 1, for the minimised or "fastest".

The inverse of s_j represents how many frames, n, can be expected using a single **perfect transformation**. A perfect transformation is a theoretical transformation that has the capacity to produce an infinite number of patterns in the selection pool, with an even spread of dissimilarity to T throughout. In a perfect transformation, when $s_j = \frac{1}{3}$, there will be 3 cycles before convergence. In practice, most transformations are imperfect and with some combinations of transformations and settings, convergence is never reached.

5.1. Target dissimilarity and transformation speed

From the user-defined transformation speed, s_j, a target value of dissimilarity, let it be t_j, needs to be established. This involves estimating how much the dissimilarity should be reduced by each step to ensure the specified number of steps, $1/s_j$, and multiplying it by the number of steps that have occurred already, i, to find the appropriate target, t_j, for this step. The estimate of the change in dissimilarity required for a single step is obtained by normalising s_j between the difference in least dissimilar, $\overrightarrow{R}_{j,0}$, which would be 0 with a **perfect** transformation, and the rating of the current note sequence that is the input, $R_{j,0}$ (recall that $P_{j,0} = bypass$). Considering that it would be unfeasible to pick a target that is below the lowest dissimilarity, $\overrightarrow{R}_{j,0}$ and recalling that i is the current frame, t_j is defined thus:

$$t_j = \max\left(\overrightarrow{R}_{j,0}, R_{j,0} - i\big(R_{j,0} - (1 - s_j)(R_{j,0} - \overrightarrow{R}_{j,0})\big)\right) \qquad (1)$$

The equation above defines the target value of dissimilarity t_j. s_j is the user-defined 'speed' (0 is slowest). $\overrightarrow{R}_{j,0}$ is the lowest dissimilarity rating. $R_{j,0}$ is the dissimilarity rating of the unmodified input, that is, the previous frame.

This definition of t_j can be understood more intuitively thus: when the speed is set to slow, $s_j = 0$, it becomes $t_j = \max\left(\overrightarrow{R}_{j,0}, R_{j,0} - i\overrightarrow{R}_{j,0}\right)$. In the **perfect** case, where $\overrightarrow{R}_{j,0} = 0$ and $R_{j,0} = 1$, this becomes $t_j = \max(0, 1) = 1$. Note also that convergence does not occur in a single frame.

With this target (the most dissimilar), the algorithm will never converge, which is appropriate considering that as s_j approaches 0, $1/s_j$ (the number of steps) will approach ∞. If the speed is set to fast, $s_j = 1$ we obtain $t_j = \max\left(\overrightarrow{R}_{j,0}, R_{j,0} - iR_{j,0}\right)$, which, in the **perfect** case, becomes $t_j = \max(0, 1 - i)$.

In this case, for $1 \le i \le \infty$, the target similarity will be 0. This means that, in the **perfect** case, the algorithm would converge in a single frame, confirming that $1/s_j = n$.

5.2. Aiming for target dissimilarity while maintaining consistency

An appropriate k^* (the index of the selected sequence) will minimise the difference of measured dissimilarity, R_{j,k^*}, to the target dissimilarity, t_j, while at the same time minimising the dissimilarity of the selected sequence O_{j,k^*} to the source, S. This ensures that in situations where the candidates are rated equally close to t_j, the one that is the most similar to S will be selected, providing a greater sense of musical continuity. This will involve interpolating between the consistency with source and target for the currently desired dissimilarity. The difference between $R_{j,k}$ and the target dissimilarity t_j is: $|t_j - R_{j,k}|$. The distance from the source is $D(S, O_{j,k})$. Let v_j be a user-defined weight, called 'tracking VS consistency', which controls the balance between tracking t_j, when $v_j = 0$, and being consistent with S, when $v_j = 1$. This effectively controls the influence of $|t_j - R_{j,k}|$ or $D(S, O_{j,k})$. Usually, $v_j < 0.5$ so the difference to the target dissimilarity is the principal component. All together, k^* is determined thus:

$$\min_{k^*} v_j \left|t_j - R_{j,k^*}\right| + (1 - v_j) D(S, O_{j,k^*}) \tag{2}$$

In the equation above, the index to the selected note sequence, k^*, is determined by minimising the difference of the dissimilarity of that note sequence with the target dissimilarity (first term above), while minimising the dissimilarity between that note sequence and the source (second term). The user can control the influence of each of these terms using v_j - the 'tracking VS consistency' variable.

6. COMPOSITIONAL TRANSFORMATIONS

The transformations that have been implemented are described here in order of their position in the transformation chain: *divide/merge*, *rate*, *phase*, *harmonise*, *scale pitch*, *inversion*, *octave*, *add/remove* and *key/scale*. All of these perform large scale transformations to the note sequences, except for *add/remove*, which deals with individual notes.

6.1. Divide/merge

The *divide/merge* transformation was envisaged as a technique for affecting the note density of the input without dramatically altering the music. *Divide/merge* has five parameter configurations, apart from 'bypass', and uses a 'Nearest Neighbour' dissimilarity measure to compare the transformed items in the selection pool with T. The five configurations are: merge forwards, merge backwards, split $\frac{1}{4}$, split $\frac{1}{2}$ and split $\frac{3}{4}$.

'Merge forwards' iterates through the note sequence, merging notes that overlap into one, retaining the pitch and dynamic of the earlier note. The end-time of the earlier note will increase to become the end-time of the later note and the later note will be removed. 'Merge backward' is similar, except that it proceeds from the last note to the first and the scale degree and dynamic are copied from the later note, deleting the earlier note.

'Split' finds the longest note and splits it in two. The three different parameter settings, $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{3}{4}$ refer to the point at which the note is split. For example, splitting a whole note (four beats) with $\frac{1}{4}$ would turn it into one crotchet followed by a dotted minim. The scale degree and dynamic for both of these notes would be the same as the original.

To compare each note sequence generated by the difference parameter settings, *Divide/merge* uses the NN dissimilarity measure (below).

6.1.1. The Nearest Neighbour (NN) dissimilarity measure

The Nearest Neighbour (NN) dissimilarity measure is the most thorough of all dissimilarity measures within *TraSe*. For each note in the two note sequences being compared, the NN measure finds the distance of the closest (neighbouring) note in the opposite sequence. The final distance returned is the average of all such distances. This is currently an inefficient process at $O(n^3)$ although some ideas for optimisation have been explored which could bring it down to $O(n\log(n))$. This may be discussed in future research.

The process for NN is bidirectional, comparing the target sequence of notes, T, to the input, I, as well as I to T. These two calculations will be referred to as **backward** and **forward** respectively. The bi-directionality is necessary because the NN of one is not necessarily the NN of the other. For an example using note onset: if T has a note on beat 1 and 3 while I has a single note on beat 1.5, the note on beat 3 in T will be closest to the note in I and the note in I will be closest to the note on beat 1 in T.

Let A and B be two note sequences and let $av(A,B)$ be the average distance between each note in A and its NN B. Let i be an index to A that ranges over $1 < i \leq |A|$ ($|\ |$ is cardinality/length) and let j be an index to B that ranges over $1 < j \leq |B|$. Let $dist$ be a function that finds the distance between two different notes primarily in terms of pitch and note onset. The av function is thus:

$$av(A,B) = \frac{1}{|A|} \sum_{i=1}^{|A|} \min_{j=1}^{|B|} \left(dist(A_i, B_j)\right) \tag{3}$$

Equation three shows the average distance between each note in A and its NNs in B.

The *forward* calculation will be $av(I,T)$ and the *backward* calculation will be $av(T,I)$. Letting nnd be the NN dissimilarity measure, we have:

$$nnd(I,T) = \frac{1}{2}\left(av(I,T) + av(T,I)\right) \tag{4}$$

The note distance function for the NN measure is a combination of distance between note onsets within the loop and distance between "octavised" scale degrees. That is, the scale degree, d, plus the octave for that pitch, o, multiplied by the number of scale degree steps per octave, s: $d + os$. This scheme favours the use of source and target note sequences within the same octave. In future work it would be simple to overcome this by calculating the octave and scale degree distance separately and combining them afterward, weighted on a user-defined variable. Other measures that compare the similarity of scale degrees and passing notes could be applied. Inclusion of duration and dynamic and the Circle of Fifths (CF) distance, as explained in the previous chapter, would also be simple.

6.2. Rate

Rate multiplies the onset of each note by a ratio, removing notes that exceed the loop length, or looping the sequence as many times as needed to fill the loop. The ratios are based on patterns from Elctronic Dance Music genres: $\frac{1}{4}$, $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{2}$, 2 and 4. For example, a straight beat transformed by $\frac{3}{2}$ or $\frac{2}{3}$ yields a common three against four rhythm. Transformations by $\frac{1}{2}$, and $\frac{1}{4}$ are notorious in breakdowns, while a rate change of 2 or 4 often increases the intensity of build-ups.

The dissimilarity measurement used for *rate* is the difference in area between the onset envelopes, combined with the difference in area between the pitch envelopes (for an example, see Figure 2, below). The difference for inter-onset is normalised by the maximum possible difference area, which is the length of the loop squared. The difference for pitch is normalised by an arbitrary maximum of 30 multiplied by the loop length, which is equivalent to one note in each pattern, 30 semitones apart. If the difference happens to be greater than this, the maximum dissimilarity of 1 is returned. These measures are sufficient, but could be improved by normalising by the maximum (for pitch only) and magnifying the lower end of the spectrum through a logarithmic function.

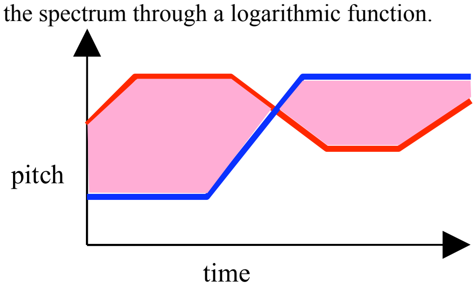


Figure 2 Two pitch envelopes and the difference in area between them. The difference in area between the onset and pitch envelopes are used as the dissimilarity measure for rate.

The result is probably similar to a NN measure. A comparison of the two measures is yet to be thoroughly investigated, however it is imagined that because *rate* preserves the contour of the envelope, the dissimilarity function that measures envelope differences is more relevant than the NN distances.

6.3. Phase

The *phase* transformation shifts the start time of the input loop by a certain amount, either ahead or behind. Onsets that exceed the loop boundary are wrapped around. The dissimilarity measure for *phase* is the NN dissimilarity measure, because of the high-level of accuracy in absolute difference between notes. The effect of *phase* is that the input pattern is "nudged" ahead or behind a certain amount, so as to maximise the absolute similarities between the patterns. This can have the musical effect of highlighting the similarities of notes relative to one and other (rather than relative to the metre) in the input and target patterns. The parameter space for *phase* is the series of quarter-intervals from -4 through to $+4$, that is, -4, -3.75, -3.5, $\cdots$, 3.5, 3.75, 4. This makes 32 different parameter settings, not including 0 or "bypass".

6.4. Harmonise

Harmonise works by either adding or removing parallel harmony at each of the tonal intervals from the 3^{rd} through to the octave. This means 12 different parameter settings in all, excluding "bypass": remove 8^{ve}, 7^{th}, 6^{th}, 5^{th}, 4^{th}, 3^{rd}; or add 8^{ve}, 7^{th}, 6^{th}, 5^{th}, 4^{th} or 3^{rd}. While not all of these intervals are particularly common, they are included to provide flexibility.

To remove a harmonic interval, the note data is analysed to find the notes that occur on the same start time but with a different pitch – "harmonic clumps". Within each clump, notes that are the specified interval above the lowest pitch are removed. For the addition of harmonic intervals, monophonic notes are doubled with a parallel harmony at the specified interval.

The dissimilarity measure used by *harmonise* is a weighted combination of the difference in average harmonic interval and the difference in average clump size. The average harmonic interval is the sum of the average harmonic interval of each clump, normalized by the number of harmonic clumps (including clumps with a single note).

The weighting between average harmonic interval and average clump size is 0.8 and 0.2 respectively. This is because the most important effects rendered by *harmonise* are the changes in the harmonic interval, but clumps of various sizes can have the same average interval and so the average clump size is included so as to help differentiate between cases such as this.

6.5. Scale pitch

The *scale pitch* transformation expands or reduces pitch range, while maintaining contour. The pitch of each note is scaled by a certain ratio, relative to a Central Tonic (CT) pitch. There are fourteen ratio values, excluding "bypass", in the set of possible parameter configurations. These are all the intervals of $\frac{1}{7}$ from 0 to 2: $\frac{0}{7}$, $\frac{1}{7}$, $\cdots$, $\frac{13}{7}$, $\frac{14}{7}$. This was chosen so that it would be possible for an octave to be shifted to any of the other scale degrees above or below it.

The CT pitch is the tonic in the most central octave covered by the pitches of the input sequence. That is, the root of the octave in which the pitch centroid is found. Firstly, the average pitch of all the notes is calculated. This is then rounded down to the nearest tonic. For example, if in the key of C and the average pitch is B6, the tonic will be C6.

To scale the pitch, firstly the interval to the CT is found for each note pitch in the pattern by subtracting the CT from it. This interval is multiplied by the scaling ratio and the fundamental is added back to it. The scaled pitch is then adjusted to be in-scale or out-of-scale, to be consistent with the original value. For example, if it was a passing note (out of scale) to begin with, it will remain a passing note after the transformation.

An alternative *scale pitch* technique that avoids the need to bound the pitches is to determine the maximum possible ratio without exceeding the bounds, and weight this ratio with a parameter to determine the final scaling ratio that is used. This approach was trialled, however, the current approach was favoured as it was more predictable and the exceeding of bounds was rarely a problem.

The dissimilarity measure used for *scale pitch* is the difference in average interval from the CT, $avc(I)$. Recalling that I is a sequence of note pitches, n is the length of I, i is an index to I with the range $0 \leq i < n$ and c is the central tonic pitch, we have:

$$avc(I) = \frac{1}{n}\sum_{i}|I_i - c| \quad (5)$$

Equation five shows the average interval from the central tonic. I is the sequence of note pitches, n is the number of note pitches in I (the cardinal). i is an index to I and c is the central tonic.

This measure is used because the average interval from the central tonic is directly affected by the *scale pitch* function. One improvement to this measure could be to store and use the CT of the original sequence, rather than recomputing it for each of the scaled sequences. This is because in some situations the CT itself may be shifted, resulting in overly dramatic scaling ratios.

6.6. Inversion

Inversion operates in a similar way to "chord inversion", but inverts the pitches of the input sequence, I, rather than a chord. The degree of inversion is controlled by a parameter, let it be p, that ranges between $-1 \leq p \leq 1$. The number of octaves, q, by which to shift the pitches that are to be inverted must be greater than the octave range of the sequence, so that inverted notes do not clash with existing notes. Letting the number of steps per octave be denoted by s, we have $q = \frac{1}{s}(\max(I) - \min(I)) + 1$, rounded down[2].

The parameter p controls which pitches will be shifted and the direction. $|p|$ is the fraction of the range that will be shifted, and $sign(p)$ determines whether the percentage is taken from the top or bottom of the range and which direction it will shift. When $p > 0$ the selected pitches are taken from the bottom and shifted up and when $p < 0$ they are taken from the top and shifted down. For example, when $p = \frac{1}{2}$, all of the pitches that are below half the pitch range of the input pattern are shifted up by q octaves and when $p = -\frac{1}{2}$, all of the pitches that are above half the pitch range are shifted down by q octaves.

To select an inversion, the dissimilarity measure used is the difference in pitch envelopes, as described above in Rate (6.2). This was chosen because the inversion process has a substantial effect on the contour of the pitch envelope.

6.7. Octave

The *octave* transformation shifts the whole note sequence by a number of octaves, either: -3, -2, -1, 1, 2 or 3. The dissimilarity measure used for octave is simply the difference in average pitch.

6.8. Add/remove

The *add/remove* transformation guarantees that the output will be closer to the target, T, than the input, I. Each note in I is considered for removal and each note in T is considered for addition[3] and because of this, the parameter space for *add/remove* is not fixed like the other transformations. The NN dissimilarity measure (described above) is used to compare each result to T, so as to ensure accurate results. *Add/remove* is always the last transformation in the chain, enabling sequences that are not brought closer to T by any other transformation to eventually converge. Add/remove can be cycled a number of times specified by the user, feeding the output immediately back into the input. This is the 'add/remove cycles' parameter.

Add/remove operates in either 'polyphonic' or 'monophonic' mode. In monophonic mode, if a note is added at an onset that is already occupied, the existing note is replaced by the new note. Specifically, the pitch of the old note is replaced by that of the new note whereas the values of duration and dynamic for the new note are derived through combination between the old and new notes. The weighting of the combination is determined by the user. This feature was added after I had trialled source and target examples with very different durations and dynamics. It was found to add a degree of smoothness to the transition.

In polyphonic mode, the new note is overlaid, regardless of whether a note exists on the same onset. If a note exists on the same pitch (and onset), the attributes of the existing note are merged with the new note, using the same weighted combination parameter as monophonic mode.

Monophonic mode has greater transformational impact than polyphonic mode because, through replacement, an addition and removal can occur in a single step. Musically, especially in contexts that are mostly monophonic, the harmonies generated by polyphonic *add/remove* can sound unrealistic. A possible improvement could be for polyphonic mode to perform complete vertical replacements. That is, add or remove vertical note groups rather than single notes.

7. FORMAL QUALITATIVE EVALUATION: WEB QUESTIONNAIRE

A web questionnaire was developed as an empirical qualitative musicological investigation into morphing between MEM loops of various styles. The two primary aims were:

1) Collect knowledge that can be used to improve future morphing algorithms.
2) Establish if *LEMorpheus* is likely to be successfully applied to real-world contexts.

These were broken down further into more specific objectives:

1) Discover elements of *LEMorpheus* generated morphs that are perceived to have a positive or negative impact, benchmarking against human composed morphs.
2) Obtain techniques for morphing.
3) Determine if there is a correlation between smoothness and effectiveness. Smoothness is defined as similarity between successive sections, whereas effectiveness is defined as the ability the music to affect the listener in a way that is perceived to be the intention of the composer.
4) Obtain opinions as to whether the morphing techniques would be applicable to games or live

[2] Adding 1 and rounding down is chosen over simply rounding up, as it includes the octaves, which would not normally be rounded up.

[3] Adding notes from the target is a kind of 'cheat'. Ultimately, *TraSe* aims to function without deriving material directly from the target.

electronic music performance.

The approach was to research, compose and recreate sets of MEM loops, create *LEMorpheus* morphs between them and pay for a professional music producer to compose and produce morphs manually.

A randomly generated generated benchmark was also considered, however, human composed morphs were eventually chosen as they provided much more realistic and challenging material than randomly generated morphs and the time constraints of the questionnaire allowed for only the most effective benchmark. As well as this, random generation is only another human compositional decision and, as such, may have provided a false sense of objectivity to the study. Ultimately, comparing against cleverly composed music provides the kind of insights which may eventually be used to make *LEMorpheus* compose in a similarly clever way.

The questionnaire was developed according to the objectives, widely publicised and completed by nine participants with musical backgrounds. Australian respondents were paid and feedback was collated and analysed.

7.1. Results of questionnaire

In terms of results, *LEMorpheus* morphs were characterised by progressive musical variations, while the human composed morphs utilised layering and larger structural changes. While the majority of respondents favoured the human composed music, there was a great deal of controversy in opinion, probably due to participants attending more or less to different elements in the music and differences in musical background. As a taste of the controversy, two very different responses are included here:

"Excellent - because of the complexity of the tracks, the highly skilful melding of the source and target was a joy to hear. It was a very deliberate transition in moods, and almost evoked a story in itself as the listener was transported from a relaxed environment with the potential for disaster directly into a schizophrenic sound-scape. Very worthwhile."

- Participant five (15 years of avid listening to popular and alternative music), responding to the *LEMorpheus* morph for the second example.

"Blechh!! Intense negative. Not effective, an apparently random substitution of elements."

- Participant four (40 years of teaching electronic, art and popular music) responding to the *LEMorpheus* morph for the second example.

A number of suggestions for additional morphing techniques were gathered, including the blending of similar elements, removal of dissimilar elements, trading of loops (as with 'beat juggling'), incremental substitution of phrases, gauging of source and target dissimilarity to determine whether to attempt a continuous blend or switch directly, generation of a separate bridge section or breakdown and synchronisation of rhythm and evolution of other elements.

There was found to be very little correlation between smoothness and effectiveness. For example, many human composed morphs were found to be coherent but not smooth, while many *LEMorpheus* morphs were smooth but not coherent. Some terminological ambiguity with 'smoothness' was apparent in some responses, however, there was sufficient additional explanation for this not to impact significantly on the validity of the data.

Overall, *LEMorpheus* was rated, near-unanimously, as being applicable to both computer games and live electronic music contexts. The instances of negative results were somewhat balanced by similar reactions to the human composed morphs, although the balance was in favour of the human composed morphs. This was mostly due to the more deliberate and granular nature of the changes composed by the human, rather than the forced "smooth" transitions generated by *LEMorpheus*. The data from the questionnaire can be readily obtained through correspondence with the authors.

8. CONCLUSION

Most aspects of the evolutionary note-sequence morphing algorithm *TraSe* have been explained, along with an overview of the formal evaluation process. The core process involves iteratively feeding the source into a chain of compositional transformations and selecting from a pool of transformed candidates after each transformation. The result after each iteration of the chain is used as a single frame in a list that constitutes the morph. The method of selection for each transformation is to compare each candidate to the target to derive a measure of dissimilarity. The dissimilarity value for each candidate is weighted by user defined parameters and the candidate which is closest to a target level of dissimilarity is selected. The target level of dissimilarity is determined from the user-defined number of steps it would take to reach the target given an ideal transformation. The level of dissimilarity of the mutated note sequence to the source is also minimised where possible.

The following compositional transformations have been explained: divide/merge, rate, phase, harmonise, scale pitch, inversion, octave and add/remove. Divide/merge splits notes into separate notes while maintaining pitch, or merges notes with the same pitch. Rate transforms the onsets within the loop by certain ratios that are common to electronic dance music. Phase shifts onsets ahead or behind by quarter beat intervals. Harmonise either removes or adds harmonic intervals. Scale pitch either stretches or shrinks the pitch. Inversion shifts pitches at the upper or lower bounds to the lower or upper octaves respectively. Octave shifts the octave up or down. Add/remove individually adds or removes notes, ensuring eventual convergence with the target.

In the formal web-questionnaire evaluation, human composed source and target material was used, human adjustment of *TraSe* parameters allowed and the evaluation involved empirical qualitative feedback from a group of nine people with musical listening skills. Particular morphs were judged by some participants to be extremely novel and musically innovative. As well as this, the music generated by *TraSe* was considered overall to be applicable to real-world applications such as computer games and electronic dance music. Despite this, the human composed morphs that were used as a benchmark were regarded as more appealing overall. Participants often differed in opinion because of different foci of attention at particular points in time and differing musical expectations based on their backgrounds.

Researching evolutionary approaches to compositional morphing of MEM has only just begun and there are many possibilities for future work that would yield substantial gain. This includes note thinning, to avoid muddiness and clashes; note clustering, to allow musical phrasing; new transformations, to extend the range of compositional capabilities; automated layering, to allow higher level structural features such as break-downs; and automatic adjustment of parameters, which would uncover less obvious – but workable – settings and reduce the time spent by the user. Load tests of the algorithm involving randomly generated material were performed and the results of these test may also be discussed in the future. As well as these, some ideas for optimisation of the algorithm may be implemented, which could allow truer realtime operation.

9. BIBLIOGRAPHY

Biles, A. 2008. “Evolutionary Music Bibliography”. accessed on: 15th of February, 2008. site: www.it.rit.edu/~jab/EvoMusic/EvoMusBib.html

Horner, A. and Goldberg, D. E. 1991. “Genetic Algorithms and Computer Assisted Composition”. *Proceedings of the International Computer Music Conference*. pp. 479-482.

Mathews, M. V. and Rosler, L. 1969. "Graphical Language for the Scores of Computer-generated Sounds". in *Music by Computers*, H. V. Foerster and J. W. Beauchamp. New York, John Wiley and Sons, Inc.**:** pp. 84-114.

Polansky, L. 1987. "Distance Music I-VI for any number of programmer/performers and live, programmable computer music systems". *Perspectives of New Music*. **25** (1)**:** pp. 537-544.

Wooller, R. 2006. "Review of compositional morphing: works, techniques, applications and possibilities". *Australasian Computer Music Conference ACMC2006:Trans*. Adelaide, ACMA.

NEURAL NETWORKS AND EVOLUTIONARY ALGORITHMS IN MUSIC COMPOSITION

ABSTRACT

In this paper we describe a system developed to generate short musical phrases in the same style as a set of training melodies. The system uses an ensemble of neural networks to rate the similarity of a generated musical phrase to a set of human composed phrases. Given this rating a genetic algorithm is used to effectively 'search' for other similar phrases. Preliminary evaluations indicate that the proposed approach shows promise. A limitation of the system is the relatively simple representation of musical phrases employed and the processing time required to use this technique on longer phrases.

1. INTRODUCTION

There has been much research into automatic computer music composition, and several techniques and approaches proposed, including the application of musical theories [1], deconstruction and intelligent reconstruction of compositions [2], transition tables [3] and cellular automaton [4] for example. Neural networks and genetic algorithms are also widely used and the application of these techniques is the focus of this paper.

The program GenJam [5] improvises jazz music using genetic algorithms using a human judge as it's fitness function. For the representation scheme music solos are broken into phrases which are then made up of four measures. A human judge listens to the generated solo and presses one or more 'g's for good or, one or more'b's for bad. From these human judgements the genetic algorithm selects measures and phrases to survive and reproduce. The phrases are then modified by musically meaningful rules which aim to create better offspring. Such modification rules for measures include transposition, inversion and reversion. For phrases modification rules include reversal, rotation and "lick thinner" which substitutes a random phrase in the measure for the most common phrase in the entire phrase population. After many generation of selection, reproduction and modification, solos are output.

Neurogen [6] also uses a genetic algorithm, however it divides its music compositions into building blocks of rhythm, melody and harmony to be judged by an independent neural networks rather than a human. Those musical phrases selected by the neural network may reproduce, but rather than the musical phrase being modified by rules informed by existing music theories a random modification rule is used.

Other projects have used a different approach in which the system attempts to complete a given musical phrase by predicting the next notes using a neural network. This prediction is the same concept as used for transition tables by which the use of neural network is an extension of [7]. The project "Learning the Long-Term Structure of the Blues" [8] (LTSM) uses this idea: it learns to predict the next notes from the previous notes using a neural network. Then given a seed of a few notes it then composes the rest of a piece using its predictions. This step by step composition might seem incapable of capturing capture long term features within music. In order to resolve this a Long Short-Term Memory [9] (LSTM) neural network architecture which is long term memory of past input allowing it to take in to consideration long term structure.

The project CONCERT [10] has a similar approach to LTSM as it predicts the next note given the previous using a neural network. However it does not use LSTM to help capture long term structure but rather a reduced description of the compositions. The individual pitches are "smoothed and compressed" so that events at a course time scale are more explicit. As the representation is also more compact it allows for a longer timescale to be considered at a time helping again in learning long term structure.

GenJam using human guided genetic algorithms with LTSM and CONCERT using predicting neural networks represent two common approaches. The less common approach of Neurogen using neural network guided genetic algorithms is most similar to the project we describe here.

1.1. Aim

The aim of this project is to develop a simple proof-of-concept system using neural networks in a novel way, that can be extended if desired. The aim of the system described in this paper is to compose single-voiced short phrases. These phrases should sound similar, to human ears, to provided training compositions. This includes emulating structure or context sensitive rules, if any, from the provided training compositions. For example if the provided compositions have rising arpeggios or call and response, then those should be present in generated compositions. The system should learn how to generate similar compositions, removing the need for an expert. The system should also be able to generate its phrases with minimal assumptions on the nature of music so that it will be flexible enough to be used in a wide range of musical contexts. In other words, given examples of a new style of composition the system will be able to learn and generate similar compositions.

Initially, we have limited our system to single voice phrases to avoid the complexity of multiple voices and longer composition. The task of emulating any structure has been explicitly noted as this is a common shortcoming of computer generated music. In the case of predictive neural network projects this shortcoming is often due to the neural network looking at a small number of prior notes or events in order to decide what should come next. Notes further back are ignored and hence global structure is lacking. This is somewhat remedied by using LSTM recurrent neural networks which have a degree of memory such as the LTSM project, or by a reduced description representation scheme as CONCERT.

This paper's project hopes to avoid the problem completely by having all notes within a composition, rather that a few local notes, as input to a neural network. In this way the context considered by the neural network is of the whole piece. This is a similar approach as NeuroGen. Whilst Neurogen treats phrases as building blocks of rhythm, melody and harmony, this project deals with a melody as a whole. Neurogen deals with a smaller palette in terms of pitch. Where Neurogen is constrained to eight consecutive notes on the C major scale, this project is constrained to twenty five notes on the chromatic scale. The most significant change is the method of evaluation of musical phrases where this project introduces an iterative process which builds a growing ensemble of neural networks.

2. OVERVIEW

The system will be emulating a set of single voice human compositions, which will be used for training the system. These compositions are made up of successive notes of particular pitch and duration. The compositions are split into phrases of sixteen beats to be emulated, or four bars of common time. The phrases are then split again into slices of quaver length. Each slice represents a portion of time, whose value represents either a new note's pitch, a hold of the previous note or a rest. Pitches themselves are represented using a numerical value where middle C is 0, C sharp 1, D 2 and so on.

To generate phrases a genetic algorithm is used. A population of random phrases is first created. The phrases are then judged to be more or less similar to the given compositions. Those more similar are deemed to be "fitter". Within the population the fittest phrase can then reproduce with crossover and mutations forming the next generation, while "unfit" phrases do not. Each successive generation then should have "fitter", of more similar, musical phrases.

In order to judge the fittest phrases a neural network is used. The neural network is trained using the prepared compositions along with generated random phrases. The random generated phrases provide a counter example and consists of randomly selected note values. The neural network's task is then to give the probability of a given phrase of belonging to the class of training compositions. This probability can be each phrase's fitness rating.

However this neural network generation might not lead an accurate judge of musical phrases. This is as a neural network trained with random and training phrases may differentiate between each class by limited or even non-musical features. For example it might be possible to differentiate between random and human phrases simply by the frequency of a note, the mean values of all notes or mode of the starting interval. While these could be features of the human phrases they do not encompass enough features to direct generation of musical phrases. In turn such a judge used as fitness function in a genetic algorithm is unlikely to produce human phrase similar compositions, as was found when tried.

To overcome this issue an iterative process is used, illustrated in figure 1. At the first iteration, a single neural network is trained with the random and human phrases. The genetic algorithm then generates new phrases. For the second iteration two neural networks are used, each with its own purpose.

The first neural network's aim is to find features that have not yet been found, features of human phrases not found in the generated and random phrases. To do this neural network is trained with the human phrases as one class and, generated and random phrases as the counter-class. The second neural network aim is to find features already found, shared features of human and generated phrases not found in the random phrases. To do this the neural network is trained with human and generated phrases as one class and random phrases as the counter-class.

This results of the neural network are again used to guide the genetic algorithm, and generate the next new set of generated phrases.

The following iterations continue to use an ensemble of neural networks to find features found in each generated set and the final neural network to find features not yet found. Features found within each generated phrases are learnt by using human phrases, all later generated sets and the generated set itself as examples, and all other phrases, being earlier generated sets and the random sets as counter-examples. Features yet to be found are learnt by using human phrases as an example and all other phrases as counter-examples. In this way the ensemble can remember what features it has learnt before while looking for new characteristics.

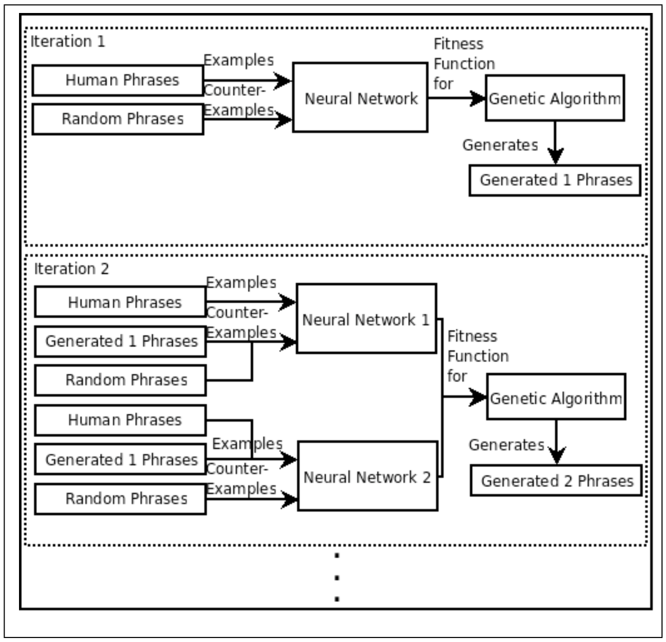


Figure 1. The iterative process as described, with each xth iteration training x neural networks.

3. IMPLEMENTATION DETAILS

3.1. Data Preparation and Representation

The human compositions are provided to the system as abc notation files. Abc notation [11] is a language for notating music which is easily readable by human and machine, with which many compositions have been written. For the system's human compositions a group of 831 Irish reels [12] were used.

The last sixteen beats of the human compositions, a beat being of crotchet duration, is used as the training phrases. This last part is used so that each training phrase has the same position in the composition. The phrases are then split in to thirty two positions where each position represents a slice of time being half a beat or a quaver in duration. Each position then has a value of either a new sounding note's pitch, a hold of the previous value or a rest. Both held notes and rests are represented as both are present in the training phrases. The pitch of a note is represented by numbering pitches on the chromatic scale from C_4 (middle C) to C_6 (two octaves above) inclusive. C_4 is zero, C_4 sharp is one, D_4 is two and so on. Notes in the human composition that fall outside this range are transposed an octave up or down until they fit. While this does limit representation to a range on the chromatic scale it can make learning and generation simpler. This particular range of just over two octaves was chosen as most human examples fell in this range. An absolute pitch representation was chosen over an intervallic representation as it can represent different keys. Holds and rests are represented by twenty five and twenty six respectively. Rhythm is represented implicitly by which position and thus when a new note is to be sounded, a note is held or sustained, and when to rest. However as each position represents a quaver, only notes which are a multiple of a quaver can be represented. Triplets and semiquavers for example would not fit the scheme, and thus phrases containing such were not used. A phrase and its representation below demonstrates the scheme.

Figure 2. Last four bars from The Primrose Lass by Matt Molloy and Stony Steps.

The phrase of figure 2 is represented as:

11-14-14-25-16-14-19-14-16-14-19-14-16-14-19-14-11-
14-14-25-16-14-19-16-19-16-14-11-9-11-7-25

3.2. Neural Networks

The neural network model used is the multilayer perceptron [13]. This neural network is able to find mappings between input data and expected output data. For this project the input data are the prepared music phrases and the output is the phrases class. The multilayer perceptron is also able to generalise. That is given training input/out pairs, the network is able to accurately predict the desired output for an unseen input.

As explained in 3.1, the input data is composed of a set of variables each representing a position of time, the value being either a hold or rest, or the pitch of a new note. Each of these variables are treated as nominal variable. This is as the variable values, pitches, holds and rests, are not assumed to be ordered in a meaningful way. In other words a position having the value of E is not seen as higher or lower than the value of the adjacent F nor the value of a rest or hold. This may seem counter-intuitive, as music theory has ordered scales and intervals which suggest meaningful order between pitches. However treating input variables as nominal resulted in much higher accuracy by the neural networks.

For nominal variables 1-of-N input encoding is suitable [14]. For each variable's possible value or category, there is a separate input neuron which acts as a flag, being either on or off. For each variable there are twenty seven values. Thus for thirty two variables or positions within a phrase, there are then twenty seven times thirty two input neurons, or eight hundred and sixty four input neurons.

The neural network uses only a single layer of neurons. This is as a single layer is all that is needed to capture any

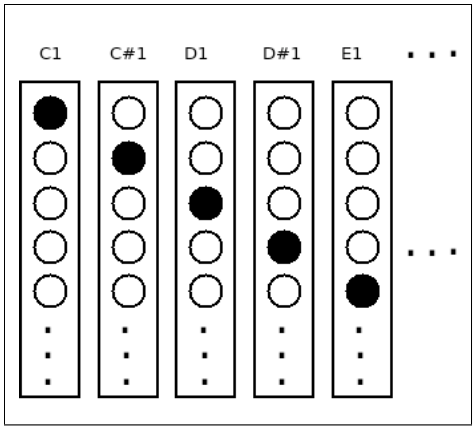


Figure 3. A 1-of-N encoding for single variable. Each circle represents an input neuron, being filled when on.

continuous function to a desired level of accuracy, given enough neurons [15], and in practice additional layers did not improve accuracy. The number of neurons to be used in the hidden layer is to be minimised as less neurons can result in better generalisation. In practice four hidden neurons resulted in accuracy that would not be significantly improved upon by simply adding more hidden neurons. Less hidden neurons performed poorly.

The final output layer has neurons that represent each class, being the examples class and the counter examples class. Training is accomplished by the back-propagation algorithm. The learning and momentum rates are 0.2 and 0.3 respectively, chosen by experimentation for higher accuracy. The learning also decays after each training epoch. This helps avoid the risk of over-fitting the data and thus be unable to generalise well.

3.3. Genetic Algorithm

The genetic algorithm treats phrases as individuals within a population of fifty. From experimentation this level of population was enough diversity to avoid stagnation. To initialise the population phrases are created with random values. An alternative would have been to initialise the phrases as pre-constructed musical phrases [16] or simply from the provided human phrases. However this was not necessary as the genetic algorithm performed well in its search with random initialisation.

The genetic algorithm finds the fittest phrases amongst the population, using the neural networks as a guide. The fittest phrases are those which when inputted to the neural network return the highest confidence for belonging to the set of human phrases. The highest phrases are selected for reproduction and then randomly put into parent pairs. Offspring are created by taking a part from one parent and the rest from the other. The resulting offspring are then mutated by randomly selecting notes, if any, to change to a random value. A dumb, or musically ignorant, mutator was used as opposed to musically aware mutator such as used in GenJam, as it does not require human understanding of a music style nor human preparation of a mutator. The offspring and the parents then form the next generation. This process is repeated for many generations until the highest rating of the population does not increase significantly, an increase of 0.001 over three generations when the algorithm is stopped. This is to avoid spending time when significant improvements are unlikely to occur. Otherwise the algorithm stops after one hundred generations.

The prior described process is used to output the single fittest phrase in the population. While there may be other highly fit phrases they will likely be very similar to the fittest phrase and thus of not much use. For each required phrase the entire process is repeated.

4. SYSTEM RESULTS

The system was run to create the generated phrases. It was found that the processor and memory requirements of the system are quite high, where the time to go through several iterations can be quite lengthy taking approximately a full day using above 3gb of memory. Each iteration add significantly to the time taken. This limited the number of iterations that could be performed, and for the purpose of this paper the system went through six iterations.

For the first part an objective test was conducted to see how well the neural network, or fitness function, would judge varying musical phrases. Secondly, in line with the aim of this project, we need to judge how similar the system-generated phrases are to the human phrases used as training data. We intend to ask human judges to identify which unlabelled phrases are human composed and which are computer generated in order to somewhat objectively measure how well the system performs. Qualitative feedback will also be obtained by asking judges the similarities and differences. However, at this early stage of the project we have conducted only a limited evaluation in which the generated phrases are examined by the authors and the effectiveness of the trained neural networks are tested in some simple experiments. These are described in the following sections.

4.1. Fitness Function Evaluation

Ideally the fitness function should be sensitive to features derived from music theory or readily apparent to most human listeners. These characteristics can include positioning of rests, syncopation, tonality, pitch direction, etc. Those phrase that have the correct characteristics should rate higher than those that don't. Phrase of similar style to the provided human composition should also rate higher than those of different styles.

4.1.1. Test Design

Specific characteristics can be tested for by modifying the human phrases to differ on the given characteristics. These modified phrases should then have lower ratings than the unmodified human phrases themselves. Figure 4 shows the last four bars of The Ewe Reel along with example modifications. In addition the neural networks will also judge sets of 12 bar blues, Swedish polska and Irish jigs to compare different styles, along with random note phrases to provide a base low rating.

Figure 4. The original phrase and it's modifications

4.1.2. Results

Phrase Set	Average Rating
Unmodified Reels	0.56039
Notes' octave randomly changed	0.25385
Transposed to an unused key	0.10806
Shifted left a quaver	0.49461
Shifted right a quaver	0.50592
First two bars doubled in duration	0.27368
Last two bars doubled in duration	0.26279
Single note changed	0.52469
Twelve bar blues	0.18607
Swedish polska	0.248957
Irish jig	0.39618
Random note phrases	0.09514

Table 1. Results of fitness function evaluation

Through the experiments the unmodified reel phrases received the highest average rating. The ratings show that the system is sensitive to key, octave, note duration and single note changes. However a shift to the left or right resulted in only a small reduction in rating. This suggests that the system is not sensitive to a note's absolute position. It should be noted too that as the phrases are single voiced shifting the phrases does not change the melody position against a backing beat. This means that shifting a melody is less noticable to human ears.

The polska, jig and blues phrases received significantly lower rating than the Irish reel. Particularly interesting is that the Irish jig came the closest to the Irish reels in ratings, perhaps due to the two genres' stylistic similarities. These results suggest that the system is sensitive to musical style.

4.2. Authors' Evaluation

While the authors were not experienced in Irish reels they are experienced musicians, the first being a pianist for many years, the second having a music degree and who performs proffesionally. In their view the system was able produce composition that resemble the given human composed reels, yet the output is still quite recognisably different. Some of the compositions are displayed in figure 5. The following was noted:

- The generated phrases were able to keep within a key. However while the training phrases had many different keys, the generated phrases used a D major or equivalent key. At times the generated phrases would use a note out of key at an unsuitable time.

- Rhythm in both generated and human phrases were similar, with the majority of notes being quavers punctuated by crotchets. Usually the crotchets are on the beat. Rests are uncommon in both.

- The generated phrases at times have more consecutive repeated notes.

- Both human and generated phrases used smaller intervals, with the occasional run. However the generated phrases would at times have a large interval jump caused by a note being an octave too high or too low. Transposing the note up or down an octave is an easy remedy as the out of place notes usually have a suitable pitch class.

- Distinct structure was hard to discern in both human and generated phrases. This is likely as the phrases are four bars long. However this lack of structure was more apparent in the generated phrases.

Figure 5. Generated phrases.

5. CONCLUSION

In this paper we have described a system which uses neural networks and evolutionary algorithms to compose music in a style similar to a given set of melodies. This project is in its initial stages but the system shows promise. While the phrases created have musical characteristics and similarities they are clearly distinct from the training phrases, as demonstrated by the evaluation results.

The system generated these phrases without requiring an expert, the author being unfamiliar with Irish reels. The system was also designed with minimal explicit incorporation of musical knowledge. As for flexibility, while only Irish reels were generated for review the system was not designed with a genre in mind. Generated Irish reels whilst resembling the human compositions in some characteristics are distinct.

5.1. Possible Improvements and Future Direction

The representation is currently very limited. Only particular pitches and durations are represented. This shortcoming was most high- lighted by the lack of ornamentation. Elements such as timbre, velocity, articulation, ornamentation, multiple voices and chords are also unrepresented along with pitches outside of the acceptable range. Note durations which aren't multiples of a quaver, such as triplets, are not represented either as they do not perfectly fit the thirty two position representation. The representation could be extended to include and express many of these elements.

The representation is also limited in the generated phrases' length. This length can be extended however this increases the complexity of the task and thus processing time. An alternative would be to treat phrases as elements to be combined into larger compositions.

The system has only been trained on Irish reel. It will be interesting to see it's results given other styles of music. This may require changes to the representation for example for many styles this might require extending the representation to include other elements particularly multiple voices, or adjusting the representation size to fit different meters.

6. REFERENCES

[1] A. C. Sorensen and A. R. Brown, "A computational model for the generation of orchestral music in the germanic symphonic tradition: A progress report," in *Sound : Space - The Australasian Computer Music Conference*, 2008.

[2] D. Cope, *Experiments in Musical Intelligence*. A-R Editions, 1996.

[3] D. Lorrain, "A panoply of stochastic 'cannons," *Computer Music Journal*, 1980.

[4] P. Reiners, "Automatous monk: A collection of generative music and art projects." [Online]. Available: http://www.automatous-monk.com

[5] J. A. Biles, *Evolutionary Computer Music*. Springer, 2007, ch. 7, pp. 137–169.

[6] P. M. Gibson and J. A. Byrne, "Neurogen, musical composition uing genetic algorithms and cooperating neural networks," in *Second International Conference on Artificial Neural Networks*, 1991.

[7] M. Dolson, "Machine tongues xii: Neural networks," *Computer Music Journal*, vol. 13, no. 3, pp. 28–40, 1989.

[8] D. Eck and J. Schmidhuber, "Learning the long-term structure of the blues," in *ICANN*, 2002.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[10] M. C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scal processing," *Connection Scienc*, vol. 6, pp. 247–280, 1994.

[11] C. Walshaw, "abc music notation :: introduction." [Online]. Available: http://www.walshaw.plus.com/abc/

[12] H. Norbeck, "Henrik norbeck's abc tunes." [Online]. Available: http://www.norbeck.nu/abc/

[13] S. Haykin, *Neural Networks A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1999.

[14] P. L. Thomas Hill, "Statistics methods and applications." [Online]. Available: http://www.statsoft.com/textbook/stneunet.html#index

[15] G. V. Cybenko, "Approximation by superposition of a sigmoidal function," in *Mathematics of Control, Signal and Systems*, 1989.

[16] A. R. Brown, "An aesthetic comparison of rule-based and genetic algorithms for generating melodies," *Organised Sound*, vol. 9, pp. 191–197, 2004.

SPATIAL-GRAINS: IMBUING GRANULAR PARTICLES WITH SPATIAL-DOMAIN INFORMATION

Etienne Deleflie
Sonic Arts Research Network
Faculty of Creative Arts
University of Wollongong
School of Creative Arts

Greg Schiemer
Sonic Arts Research Network
Faculty of Creative Arts
University of Wollongong
School of Creative Arts

ABSTRACT

Granular synthesis techniques have been appropriated for 3D sound spatialisation in a number of ways, such as the spatial encoding of individual grains. This paper proposes a new technique that aims to use the spatial information already encoded in ambisonic signals, the principle hypothesis being that this encoding is actually retained at the granular level. This opens up exciting new possibilities for spatial sound. The paper outlines some of these possibilities but focuses primarily on the synthesis of non-point sources of sound which forms the basis for a second hypothesis, involving functions that relocate spatially encoded grains in time.

1. INTRODUCTION

Granular synthesis techniques have been used by a number of composers working with different spatialisation techniques from diffusion to ambisonics.

Truax, composing for diffusion environments, has used granular synthesis for the decorrelation of sounds projected over multiple speakers, thereby imbuing a sense of aural volume [1]. Barrett, working in ambisonics, has explored allocating individual spatial positions to grains [2].

Roads summarises the spatialisation of sound particles into two broad approaches [3]:

- the spatial encoding of individual particles and,
- the use of processing techniques to use particles as spatialisers for other sounds.

This paper proposes a different approach that retains both the original spatial encoding and the original sound content of an ambisonic source file; this is the first hypothesis.

A natural extension of the technique proposed in this paper, in combination with specific ambisonic source files, involves the synthesis of non-point sources of sound; this forms the second hypothesis.

2. PROPOSED TECHNIQUE

A new technique is proposed that uses the spatial encoding already inherent in the component channels of an ambisonic signal as a palette of pre-spatialised grains. The technique involves no spatial encoding other than that which is inherent in the source file.

In the granulation of source files, a grain is defined by the following micro-control properties [3, 4]; position (in the source file), duration, envelope and pitch (or speed of playback). In the context of the technique proposed, spatial encoding is retained by using the same micro-control parameters for each component (or channel) of an ambisonic signal. The resultant set of grains are the component parts of what is referred to hereafter as the 'spatial-grain'.

For example, a first order ambisonic signal typically contains 4 components (or audio channels) named W, X, Y and Z (for periphonic B-format). Since it is the combined components that define the spatial encoding, maintaining the same grain micro-control over W, X, Y and Z should maintain the spatial encoding for each spatial-grain.

> *" The grain is an apt representation of musical sound because it captures two perceptual dimensions: time-domain information (starting time, duration, envelope shape) and frequency-domain information (the pitch of the waveform within the grain and the spectrum of the grain)"* [3]

As an extension to this description, the spatial-grain also contains spatial-domain information.

2.1. Importance of the source content

Since both the sound content and the spatial encoding of the source material is retained in a spatial-grain, it is therefore the contents of that source material that will define the nature of the spatial opportunities that we intend to explore.

Consider the example of an ambisonic recording of a car moving left to right. The recording is granulated, or broken up into tens of thousands of spatial-grains each lasting 10-50 msecs. Since each spatial-grain retains the original recording's spatial-domain information, it is now possible to mix spatial-grains of the car in the left position with spatial-grains of the car in the right. The result would be granulated car sound coming from both the left and the right.

In other words, the temporal dimension of the recording has been abstracted so that individual spatially-encoded grains may be selected and played in any order. In effect, each spatial-grain is accessed by specifying a time-position in the source ambisonic file. Essentially, time is used as an index to the library of spatial information contained within the source.

3. SYNTHESISING NON-POINT SOURCES OF SOUND

While much has been written about synthesising point sources of sound, relatively little has been published about synthesising non-point sources of sound [5]. Perhaps due to complexity, the methods described [5, 6, 7, 8] have not yet found widespread use amongst composers.

The technique we have proposed could be used in combination with specific source files to synthesise non-point sources of sound, by choosing a collection of spatial-grains whose locations model a line, plane, or any volumetric shape. The term "volumetric" is used in this instance to refer to the dimensions of the space and is not to be confused with the term "volume" which is commonly used to refer to sound level.

Reconsidering the above example of a car moving left to right, imagine now that all grains -- all tens of thousands of them -- are played back simultaneously. This would encapsulate every position in which the car was recorded and would effectively model (instantaneously) the sound's spatial trajectory from left to right independent of its temporal trajectory. The result would effectively be a line source of sound.

Ambisonic recordings often contain sounds that emanate from many directions other than the principle subject (i.e. the car). It will be useful, therefore, to create an ambisonic source file which contains a point-source of sound that is completely isolated from ambient sound. This can be done by ambisonically encoding a point source of sound such as a synthetically generated sine tone.

Now consider how such a synthetically created point source might zigzag throughout a cubic volumetric space. Because the source file is ambisonic, the recording would contain spatial-grains that essentially occupy every possible position within the space. Removing time domain information reduces an ambisonic recording to a granular cloud spread throughout the entire cube -- and played in a single instant.

Lastly, instead of spreading the granular cloud evenly over the entire recording, imagine using an algorithm such as a statistical function to select which spatial-grains to play. If that algorithm includes a time component, then it becomes a sequence. This gives the potential to create dynamic, moving, morphing volumetric shapes projected in sound.

The success of this second hypothesis depends on how the clouds of spatial-grains are perceived psycho-acoustically. It is one thing to model a non-point source of sound, it is another for it to be heard that way.

4. PROCESSING ENVIRONMENT

Many audio processing platforms contain some form of support for granular synthesis techniques. These include independent software applications such as Road's Cloud Generator [3] and plugins for platforms such as MaxMSP. The implementation of spatial-grains has some extra requirements that need to be satisfied.

4.1.0. Control

To retain spatial information, the same micro-control parameters must be applied to every component grain of a spatial-grain.

Some granulators implemented exclusively in stereo may not be appropriate if the random number generators are encapsulated within the granulator; and can not be overridden.

4.1.1. Non real-time processing.

The ability to record a CPU intensive process to file. There are two ways in which processing with spatial-grains can scale up CPU usage beyond the capabilities of current processing power; the first is in an increase in the number of spatial-grains; the second is to offer more accurate localisation by increasing the ambisonic order.

The first scaling factor involves the number of component grains. When extending spatial-grains to higher orders, CPU usage will increase dramatically. For a periphonic B-format ambisonic source, each spatial-grain will contain 4 component grains. When this technique is extended to periphonic second order ambisonic material which has 9 component channels, each spatial-grain will contain 9 component grains. Periphonic fourth order is likewise modelled using 16 channels, thereby requiring 16 component grains.

The second scaling factor involves the number of grains used per second – or the granular density. As an indication, Truax in 1988 achieved a granular density of approximately 2000 grains per second for real-time processing [9]. Computer power has come a long way since 1988, but considerably more will be required now to produce granular synthesis where each individual spatial-grain is ambisonically encoded. For example, in a second order periphonic ambisonic signal where there are 9 component channels, 2000 spatial-grains per second will equate to 18000 component grains per second.

4.1.2. Score-like environment.

An ability to score the orchestration of spatial-grains in such a way that this orchestration can be repeated in non real-time.

Tests were conducted on a Core2Duo Macbook Pro. To leverage the power of both CPUs, it was decided that Jackdmp [10] (a multi-processor version of Jack [11]) be used such that the granulation could be done on one CPU, and the ambisonic decoding could be done on the other CPU. Jackdmp enables routing audio signals between applications running on different CPUs. Ambisonic decoding was done with AmbDec [12], a recommended ambisonic decoder [13].

The first implementation was done with PureData.

4.2. Exploration in PureData

Whilst PureData (Pd) does not, strictly speaking, implement a non-real-time mode it can execute expensive processes at 100% CPU and still successfully write audio signals to file. Pd also does not have a purpose built score feature but can easily communicate with other score-like applications (such as Csound).

Initial explorations in Pd exposed an important distinction between control data and signal data. When control messages were used to organize the macro structure of grains (which involves triggering grains in time) audio artifacts were created. These artifacts were caused because control data is not processed as often as signal data. In a vanilla Pd install, control data are only processed at the start of a block of 64 samples of a signal. If a control message does not land squarely -- in time -- at the start of a block of samples then it is processed in the next block. The software architecture of separated signal data and control data is common to much audio software.

Since granular synthesis involves triggering grains of sound which are typically 10-50 msec in length (grains whose size is not aurally distinguishable as an event), many grain triggers will not be time-accurate, since many triggers will not fall squarely at the start of a block size.

There are a number of solutions, within Pd, that can be used to work around this issue. One is to use a sub-patched defined block size of 1 (using [block~]), such that control messages are processed at every sample (restricted to a chosen sub-patch). Another is to use a set of sample-accurate trigger externals (known as [t3~]) bundled in the 3rd party lib IEMLIB [14]. Yet another would be to write C externals that behaved in the desired manner.

However, all of these methods would introduce a performance degradation which can be circumvented by working within Pd's control data rate. Whilst this limits the control one has over certain granular parameters, it allows for very efficient processing. Triggering 20 msec grains at 4 msec intervals, within a sampling rate of 48 kHz allows perfect synchronization of control messages with a block size of 64 samples. Using this method 20,000 1st order spatial-grains per second could be generated. This is equivalent to 80,000 grains per second.

The Pd patch shown in Figure 1 was used to confirm the first hypothesis, that spatial information can be retained in spatial-grains.

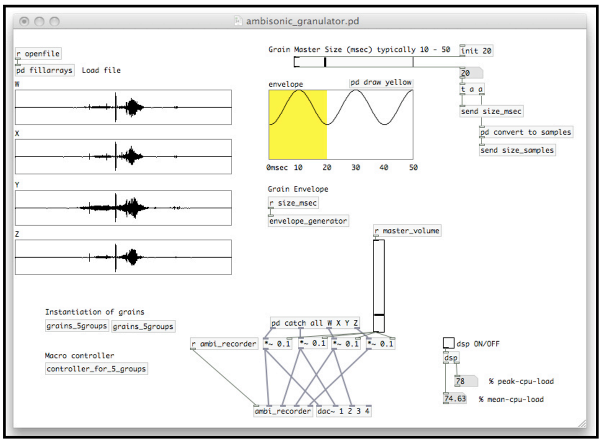


Figure 1. Screenshot of Pd patch shows the 4 channel B-format source file and some micro-control parameters. The patch also abstracts objects used for macro-control.

5. INITIAL RESULTS

A recording of a steam train moving right to left was initially used. When incrementally spreading the grain cloud from a time width of 4 msec to a time width of 20 sec (covering the entirety of the train's movement), a spatial gesture somewhat like a controlled audio 'explosion' (exploding right to left) was created. At the end of the 'explosion' granulated steam train sounds could be identified from the right to the left of the sound field.

This is not to say anything about the apparent width of the resultant sound.

Initial explorations involving the adjustment of granulation parameters and temporally changing spatial-grain cloud spread confirmed that rich and complex spatial textures could be created.

It became quickly apparent that the 'cleanliness' of the source file has a strong impact on the result. In this context, 'cleanliness' is used to refer to the extent of isolation of source sounds from ambient sounds. When engaged in open-ended exploration of spatial sound textures, a 'dirty' source file may have more interest. When attempting to create a designed spatial result, 'clean' source files may offer better results.

Lastly, it was evident that an enormous number of spatial-grains are required to granulate the entirety of a source file (depending on the length of that file).

6. A MORE APPROPRIATE AUDIO PROCESSING PLATFORM

Once the initial results were confirmed, the Pd patch described above quickly revealed its limitations. The support of a strong score and non-real-time processing environment became a priority. Sample-accurate triggering of grains became necessary.

Research returned to audio processing platforms and granulators. Searching for sample-accurate grain triggering revealed other solutions such as GMEM's bufGranul~, an external for Max/MSP [15].

A quick look at SuperCollider, however, revealed that it was better suited to the task.

6.1. Exploration in SuperCollider

SuperCollider (SC) caters well for non real-time processing. It provides an OSC message based score syntax which can be executed in non-real-time, and the results saved to disk.

SC includes a range of granular synthesis unit generators (uGens) which provide sample-accurate triggering.

SC's architecture also supports a powerful multichannel expansion feature. Passing an array of signals to the input of a single uGen causes that uGen to automatically copy itself for each signal in that array. This made short work of converting GrainBuf (a granulation uGen) for processing multi-channel source files. The following code triggers 4 sample-accurate grains (all with identical micro-control parameters) from 4 input buffers:

```
GrainBuf.ar(
        // number of channels
        1,
        // trigger (sample accurate)
        Impulse.ar( triggerFrequency ),
        // duration of grain in seconds
        duration,
        // the audio buffer
        [w, x, y, z],
        // playback rate
        1,
        // position in file 0 to 1
        pos,
        // pitch shifting interpolation
        // 1 is none, 2 linear, 4 cubic
        1,
        // pan left or right (none)
        0,
        // window envelope
        windowbuf
)
```

Converting GrainBuf for the creation of spatial-grains -- where each component grain used identical micro-control parameters -- only required passing an array of buffers (instead of a single buffer) into its audio buffer argument.

7. SYNTHESIS OF NON-POINT SOURCES OF SOUND

Potard [16] identifies the following approaches for synthesizing apparent source size:

1. Boosting the zeroth order (known as W for B-format) in ambisonic signals.
2. Turning an ambisonic signal inside out, to define a point source radiating outwards. This is known as O-format [6].
3. VBAP spread. Spatialising multiple point sources around the main source. A technique proposed for use with a simple method of spatial encoding called vector based amplitude panning (VBAP).

Potard also proposes and perceptually evaluates a fourth approach involving the decorrelation of multiple point sources.

Much research has been conducted into the psycho-acoustic perception of apparent source size [7]. The principle factors indentified as affecting the perception of apparent source size are [16]:

- The inter-aural cross correlation coefficient,
- sound loudness,
- pitch,
- signal duration.

It is interesting to note that no sophisticated spatial modeling is required to deliver three of the above factors: sound loudness, pitch and signal duration. This explains, perhaps, how composers using sound diffusion techniques (where typically stereo music is diffused over multiple speakers placed amongst the audience [1]) have been able to intuitively create apparent size in sound [17].

The second hypothesis proposes a technique for generating non-point sources of sound. The success of this technique will not be perceptually evaluated in this paper.

Generating non-point sources of sound essentially involves the design of the content in the source ambisonic file. It should be noted that this approach may not be limited to ambisonic source files. A similar approach could perhaps be applied to the channels of a 5.1 file, or to the speaker feeds derived from an ambisonic signal.

7.1. Generating a line of sound

A point source of sound, spatialised ambisonically, is recorded moving from A to B. This recording will contain spatial information describing the sound at every position from A to B.

If the entire recoding is turned into grains, all of which are played back simultaneously, then a line source of sound will be modeled.

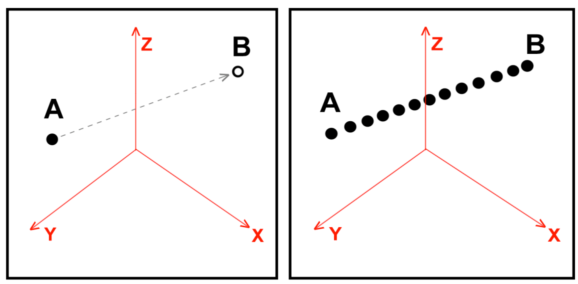


Figure 2. shows a spatial trajectory (on the left) which, when recorded, essentially contains the spatial encoding of that sound object at every point between A and B (illustrated by dots on the right). The dots can be viewed as a crude representation of spatial-grains. If all spatial-

grains are played back simultaneously, a line is modeled. In reality each spatial-grain can be as small as 10 msec in length.

7.2. Generating planar and volumetric sources of sound

Generating planar and volumetric sources of sound is similar to lines. This time, the point source of sound is moved on a spatial trajectory that covers either a plane or a volume.

A surface can not be covered in its entirety by a point source in the same way that a line is covered by a moving point. Generating a plane is ideally done from a moving line, but a well positioned moving point should achieve a similar result.

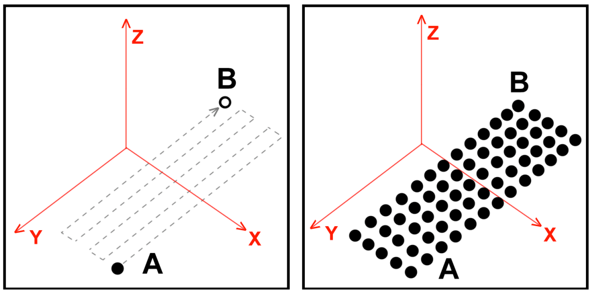


Figure 3. shows an example of a path (on the left) used to cover a planar area, and the resultant planar shape (on the right) when spatio-granularised.

By extension, creating a volumetric shape involves only defining a different more complex spatial trajectory.

7.3. Application of algorithms for variation

Algorithms can be implemented to alter the parameters of certain spatial-grains (like loudness), thereby creating non-uniform linear/planar/volumetric shapes.

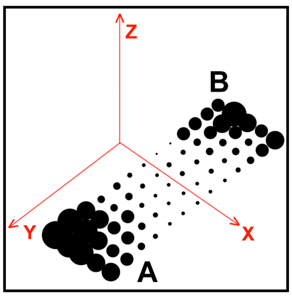


Figure 4. shows the potential effect of applying an algorithm which varies the loudness of spatial-grains. The size of each dot represents the loudness of a spatial-grain.

Consider a long spatial trajectory where a point source is moved to effectively cover every inch of space for a 100 x 100 x 100 m volumetric space. This would effectively create a spatial palette of every possible position within that space. Once that palette is created, algorithms can be designed to choose which of the palette's spatial-grains will be played. These algorithms could model any shape or form possible within that space, and affect any of the micro-control parameters used to create the spatial-grains.

7.4. Introducing time component to the algorithms

If the algorithms include a time component then the shapes created can begin to change in time. This could be used to create spatial gestures with movement or morphing of volumetric shapes.

7.5. Advantages of this approach

Granular synthesis has the secondary advantage of being an effective technique used to perceptually 'blend' – or decorrelate -- multiple sources of sounds into one larger source.

> *"The volume, or perceived magnitude, of a sound depends on its spectral richness, duration, and the presence of unsynchronized temporal components ... Electroacoustic techniques expand the range of methods by which the volume of a sound may be shaped. Granular time-stretching is perhaps the single most effective approach, as it contributes to all three of the variables just described... It should be noted that delays of only a few milliseconds are sufficient to decorrelate the different grains streams and thus increase their sense of volume."* [1]

Another advantage of the proposed technique is that there is no processing overhead involved in spatial encoding, since all spatial encoding is already present in the source file. Further, many (tens of thousands) of separate point sources can be used, with relatively little impact on CPU.

7.6. Limitations and possible workarounds

7.6.0. Sounds limited to contents of source file.

The sounds generated, and the spatial encoding generated, are both sourced from the ambisonic source file. Resultant spatial sound designs are therefore limited (in both audio content and spatial character) to what is present in the source file.

A potential technique to circumvent this limitation might involve convolution to substitute the source audio content (e.g. a 440Hz sine tone, or an impulse) with a more temporally complex sound.

7.6.1. Restricted spectral character.

When dealing with spatial-grains which have a time dimension of 10-50 msec, there is a limit to the sound's spectromorphological [18] evolution (once converted to a non-point source). In effect, the temporal character of a source sound is limited, since the differences between the source sound at time t and time t_1 are (essentially) used to distinguish two separate spatial positions, and not two separate spectral profiles. In other words, when the source sound is moved from A to B, it will typically remain the same sound. Of course, one could explore using sound sources that change in time thereby associating specific and different spectral characters for each spatial position.

Again, using convolution to replace a known source sound may offer a way to include rich spectromorphological evolution.

7.6.2. Reverberation in the source file.

Time based effects such as reverberations (both early and diffuse) included in the source file could easily be lost. Reverberations may occur in a window of time larger than the spatial-grain size.

One technique to retain reverberation information would be to make sure that the spatial-grains containing the reverberations are included in the cloud. Another simpler workaround would be to increase the grain size to include reverberation effects. This may require extending the grain size to greater than 50 msec – where a grain may start to be recognized as a single event – which may require greater care in the granulation so as to avoid a pointillistic sound character.

7.6.3. Doppler shift in the source file.

Any Doppler shift applied to the movement of the point source (in the original ambisonic file) will not help localise the resultant line/plane/volumetric shape. Rather, it may obfuscate the cohesion of the resultant shape due to the variation in pitch which is no longer relevant to the re-mixed content.

A designed or synthetically created sourced file can easily omit simulation of the Doppler shift.

It should be possible, however, to simulate the Doppler effect in a cloud of spatial-grains. Since 'playback speed' is one of the granular micro-control parameters, it should be possible to define a per-spatial-grain playback speed, that mimics a Doppler shift, simply by knowing where the spatial-grain is, and in what direction it is moving.

7.6.4. Two step process.

The contents of the source file may be designed (synthesized) or recorded. In either case, this file must be generated ahead of time, before being processed into spatial-grains.

A source file could be generated in real time, with the granulation following closely behind it. If an entirely new spatial trajectory is to be modeled, its complete granulation can only occur after the movement has been completed.

8. FUTURE WORK

Future work is needed in the application of spatial-grains using time as an index. This involves some fine tuning in two areas:

8.1.0. Micro-control parameters

Experimentation is required to understand how micro-control parameters affect the spatial image of the spatial-grain cloud. This will involve tweaking parameters that control grain size, grain envelope and speed of playback.

In addition to the modification of the source sound as it accessed using time as an index we expect that increasing the spatial-grain size will produce interesting spatial sound design possibilities given that time based spatial cues may be embedded in the source file.

8.1.1. Statistical functions

Statistical functions will be useful for varying the volumetric shape and character of spatial-grain clouds affording new opportunities for the exploration of the spatial composition. Statistical functions involving a time parameter can introduce movement.

8.1.2. Convolution

Convolution may allow a method for substituting the character of the sound in the source ambisonic file, with a more spectro-morphologically interesting one. This would free the sound designer from being restricted to the sound content of the original source file, whilst still being able to explore its wealth of spatial information.

Initial explorations will focus on simple spatial gestures characterized by a 'dualism' [19] between two spatial states.

All three areas will be explored further using various ambisonic speaker configurations possible in the newly configured speaker array called CHESS [20].

9. CONCLUSION

The proposed technique involves granulating ambisonically encoded audio files in a way that makes use of the spatial encoding embedded in the file.

Two audio processing environments have been investigated. An initial proof-of-concept implementation was done with PureData. SuperCollider has shown promise as an appropriate platform for multi-channel processing.

Spatio-granulation of a B-format recording confirmed that spatial information could be retained, confirming the first hypothesis. Some simple manipulations of spatial-grain clouds also confirmed that this technique could offer exciting possibilities for spatial sound design.

A second hypothesis, in which non-point sources of sound can be synthesized, has been outlined and discussed. The exploration and implementation of the second hypothesis is ongoing.

10. REFERENCES

[1] Truax, B. "Composition and diffusion: space in sound in space". in *Organised Sound*, 1998, pp. 141-146.

[2] Barrett, N. "Spatio-musical composition strategies". in *Organised Sound*, 2003, pp. 313-323.

[3] Roads, C. *Microsound*. Cambridge, Mass. London: MIT Press, 2001.

[4] Opie, T. "Sound in a Nutshell: Granular Synthesis". *Music*. Melbourne: La Trobe, 1999.

[5] Menzies, D. "W-panning and O-format, tools for object spatialisation". *AES 22nd International Conference*. Espoo, Finland, 2002.

[6] Malham, D. "Spherical Harmonic Coding of Sound Objects - the Ambisonic 'O' Format". *AES 19th International Conference*. Schloss Elmau, Germany, 2001, pp. 54-57.

[7] Potard, G. and Burnett, I. "Decorrelation Techniques for the Rendering of Apparent Sound Source Width in 3D Audio Displays". *Conference on Digital Audio Effects (DAFx'04)*. Naples, Italy, 2004.

[8] Menzies, D. "Ambisonic Synthesis of Complex Sources". in *Audio Engineering Society*, 2007, pp. 864-876.

[9] Truax, B. "Real-time granular synthesis with a digital signal processor". in *Computer Music Journal*, 1988, pp. 14-26.

[10] Letz, S. *Jackdmp. A multiprocessor version of Jack.* 1.9.0 Edition., [cited 29 April 2009]. Available from: http://www.grame.fr/~letz/jackdmp.html

[11] *Jack. An audio connection kit.* Edition., [cited 29 April 2009] Available from: http://jackaudio.org/

[12] Adriaensen, F. *AmbDec*. 0.2.0 Edition., [cited 29 April 2009]. Available from: http://www.kokkinizita.net/linuxaudio/

[13] Heller, A. and Benjamin. "Is My Decoder Ambisonic?". *AES 125th International Conference*. San Francisco, CA, USA, 2008, pp. 7553.

[14] Musil, T. *iemlib for PureData*. 1.17 Edition. Graz [cited 29 April 2009]. Available from: http://pd.iem.at/iemlib/

[15] Bascou, B. "A flexible granular synthesis environment in Max/MSP". *Sound and Music Computing*. Salerno, Italy, 2005, pp.

[16] Potard, G. *3D-audio object oriented coding [manuscript] / by Guillaume Potard*, 2006.

[17] Austin, L. "Sound diffusion in composition and performance: an interview with Denis Smalley". in *Computer Music Journal*, 2000, pp. 10-12.

[18] Smalley, D. "Spectromorphology: explaining sound-shapes". in *Organised Sound*, 1997, pp. 107-126.

[19] Doornbusch, P. and McIlwain, P. Integrating spatial parameters in composition practice. In: Vickery L, (ed) *Converging Technologies: Aust Computer Music Assoc Conf 2003*. Edith Cowan Uni, Perth, Australia: Australasian Computer Music Association, 2003.

[20] C. Ritz, G. Schiemer, et al "An Anechoic Configurable Hemispheric Environment for Spatialised Sound," in Proc. *Australasian Computer Music Conference (ACMC 2008)*, Sydney, Australia, 10-12 July 2008.

IMPROVISATIONAL ASPECTS OF IMAGE AND GESTURE SONIFICATION

Christopher Keyes
Department of Music
Hong Kong Baptist University

ABSTRACT

This paper/demonstration presents various approaches to gesture and image sonification and details those taken by the author in the creation of *SIHyperI* (Sonification and Intermedia Hyper-Instrument), a tool for interactive installations and real-time audio and visual performance. It discusses the aesthetic and functional issues that arise from the desire to use sonification for artistic and compositional purposes rather than for its more common application in auditory display. Interactivity, improvisation and intuitive parameter mapping are addressed along side their practical application.

1. INTRODUCTION AND MOTIVATION

The mere existence of the genres of opera, theater, and film attest to the long held desire for multimodal artistic presentations. They allow the exploration of relationships between artistic idioms that may reinforce each other or contrast in meaningful ways. Sonification, the conversion of data to abstract sounds, affords many novel relationships between images and music that can be exploited for this purpose. In the field of auditory display, which deals with the more practical aspects of representing large amounts of data as sound, these relationships are governed more by issues of perceptibility and clarity than aesthetic concerns, though they are not by any means mutually exclusive. However, the resulting audio is often more deterministic, utilitarian, and emotionless than what concertgoers are accustomed to. This paper details strategies for making sonifications more musical and expressive, as well as strategies for bridging the gap between still (timeless) images and the temporal nature of music.

2. SONIFICATION FOR COMPOSITION VS. AUDITORY DISPLAY

In their article *Listening to the Mind Listening: An Analysis of Sonification Reviews, Designs and Correspondences* [1] the authors discuss the results of a sonification competition with the criteria that submissions be both “data driven” and “musically satisfying”. They note that many of the composers/submitters viewed these as somewhat contradictory objectives, and that one seemed to constrain the other. Indeed it seems hardly likely that a systematic parsing and mapping of data to sound will have the ebb, flow, and intuitive nuances that we are accustomed to in music. One solution to this dilemma is the introduction of interactivity. Even within the realm of auditory display, many authors have stressed the need for interactivity in the meaningful exploration of data [2,3]. Interactivity becomes even more important in musical/compositional applications where one is charged with the task of creating a musical composition, which usually implies a design and structure to the flow of stimuli over the duration of a piece such that it becomes musically satisfying. Kirsty Beilharz has documented her approach to using biologically inspired generative algorithms towards this aim [4].

2.1. SIHyperI

In *SIHyperI* interactivity is implemented by the use of a mouse to control the precise point or points (rows and columns) of the image to be sonified. Depending on the exact mappings chosen, directing the mouse towards parts of the image that are lighter or darker will impart various kinds of contrasts; contrast of pitch (in the case of ‘notes’), frequency range (in the case of noise filtration or granular synthesis), pitch density, duration, timbre, timbral density, amplitude, or scale (diatonic, octatonic etc.). Such contrast, if used deliberately to form patterns, are of course the basic building blocks of musical structure, and thus given an image with a reasonable amount of contrast moving the mouse from one place to another on that image (in some perceptible pattern) will impart structure. Doing so in a manner rehearsed and refined can in turn impart nuance. Although *SIHyperI* allows one to select a chaotic algorithm for moving around the image, or to scan it left to right or top to bottom, experience shows that these are often perceived as compositional mistakes. By interacting/improvising with the mouse and feeling where in the image the mouse ‘needs to go’, a sense of drama can be created that mere scanning or relying on algorithms ignore. This is easily demonstrated using the software and becomes especially apparent when mapping brightness to loudness and/or density (notes/events per second) where the operator has an especially good control over the dramatic curve of the sonification over time. For interactive installations *SIHyperI* allows one to use the xy motion tracking from a web cam instead of a mouse such that a participant merely points to the part of the image he/she wishes to sonify.

Other means to achieve a more traditionally musical sonification involve attention to texture. In sonification for purely auditory display, the texture often remains static and is quite often either strictly monophonic or pointalistic. A Geiger counter is a simple example of this. However, mapping the data to multiple sound generators, computationally affordable on contemporary computers, can result in more complex textures as granular synthesis algorithms may create a background while MIDI and samplers may provide a foreground, even if driven by the exact same data. Pitch/frequency choice also becomes an issue here, especially if one uses MIDI synthesizers or other equal tempered sound generators. Mapping the data coming from an image of a distant galaxy may lend itself to chromatic pitch mappings where as images of flowers may be complimented more by diatonic or pentatonic mappings, though choices will vary with taste. Again, this is where composition and auditory display impart different demands.

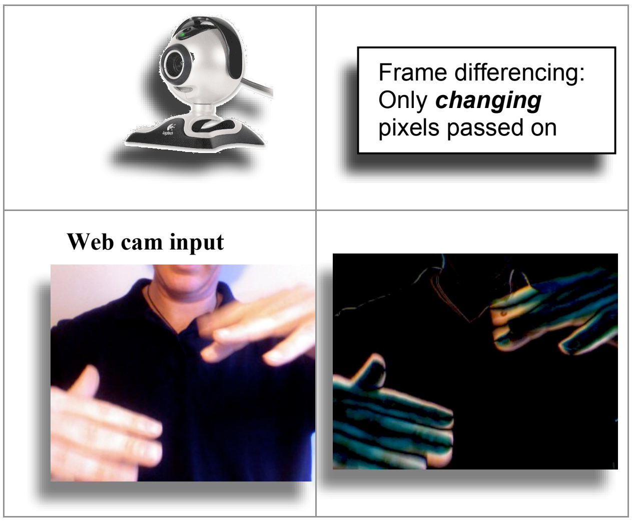


Figure 1. Frame differencing

3. IMAGE ANALYSIS OF GESTURES

Fortunately, for musical purposes, gestures exist in time and are thus relatively easy to capture and map when compared to still images. In *SIHyperI* the primary process used in gesture capture is frame differencing via a web cam; an absolute difference is calculated between the current and the last frame resulting in only delta pixels (pixels whose values have changed) passing through. Pixel values that have not changed from one frame to the next are sent as '0' (black) values (see figure 1). *SIHyperI* then tracks motion along the x and y-axes. Although there are many approaches to motion tracking, the method we use is somewhat unorthodox. 3 equally spaced columns from both the left and right side of the incoming matrix are extracted and their values averaged together to form a list of 127 values ranging from 0-255 (8 bit processing). To map these to MIDI notes the pixels with the highest values are found (post frame differencing, thus corresponding to the pixel that has changed the most) and it's position in the list calculated. Numbers at the start of the list correspond to pixels from the bottom of the matrix, which are mapped to lower notes, and numbers at the end of the list correspond to pixels at the top and are mapped to higher notes. The entire list is also used as an equalization curve for filtered noise (see figure 2).

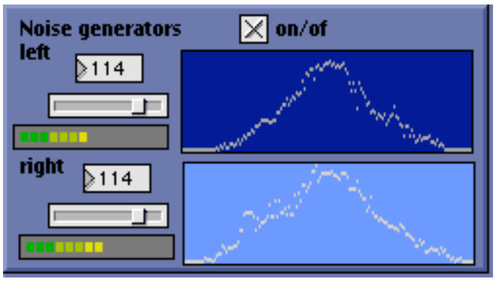


Figure 2. Equalization curves

The use of frame differencing has an important side effect for x/y axis gesture tracking in that the brightest pixel will still be lighter or darker depending on the amount of motion that generated it. Thus aside from its position it also has a *quality*, another dimension of information that can be used for expressive purposes. Similarly one very intuitive mapping is the total amount of motion generated to the amplitude of audio processing. This mapping is easily accomplished by summing (post frame differencing) the entire matrix. Thus high rates of movement correspond to high amplitudes and if there is no movement at all, there is no sound. This creates a very intuitive improvisational environment in which to create music with a web cam.

4. STILL IMAGE ANALYSIS

Still image analysis for use in the time domain is somewhat more challenging. One must first ask exactly what part of the image is to be analyzed and when. One approach is to analyze the entire image at once, allowing the mapping each of the pixel values to the frequency and amplitude of a tone generator, the transposition level of a granular synthesis instrument etc. Although this would enter the time domain for moving images and gestures, its application to still images simply creates a drone, which can be potentially annoying after relatively short periods. As stated above, *SIHyperI* uses the interactivity of the mouse to move around the image via crosshairs. One may select either the horizontal rows and/or vertical columns as data to sonify, or the precise point/pixel where the crosshairs meet. The latter then creates a 4-element set of data for the ARGB values of that pixel or, as is often the most intuitive, brightness, which is the weighted sum of the RGB values: $Y = (0.299 * R) + (0.587 * G) + (0.114 * B)$. The image analysis outputs, for the whole image, a column, row, or an individual pixel, its redness, greenness, blueness, maximum, minimum or overall brightness, saturation, contrast, dominant color and xyz axis position. Although analysis of other image characteristics are possible, their effectiveness is directly proportional to their perceptibility, and perceptibility is a key issue noted by Weinberg [5] and others.

5. MAPPING DATA TO DIFFERENT LAYERS/TEXTURES

Attention to texture is what often delineates a musically satisfying sonification (or composition for that matter) from a more one-dimensional aural display. As with an orchestra, this usually involves using more than one instrument, even if they share the same "data" as in a harmonic progression or scale. *SIHyperI* currently uses 9 sound generators of different kinds. Sound generators include the following with their detailed mappings.

5.1. MIDI

Finding individual MIDI notes is especially intuitive when tracking motion on the x, y, or z-axis. *SIHyperI* further splits the y-axis into two halves such that a performer or participant (in the case of an installation) can use their right and left hands to create counterpoint. In order not to overwhelm the instrument with data when sonifying an entire row or column of pixels, the values are placed in a table and accessed according to the number of notes per second chosen. This value seems most intuitive when mapped to brightness, accomplished, as are most mappings, by pull-down menus (see example 3). It is in this module that the use of scales, mentioned above, comes into play. In the case of the chromatic incoming values from 0-127 are simply matched directly to MIDI values. For other scales the incoming values are used as indexes for look-up tables containing the various scales.

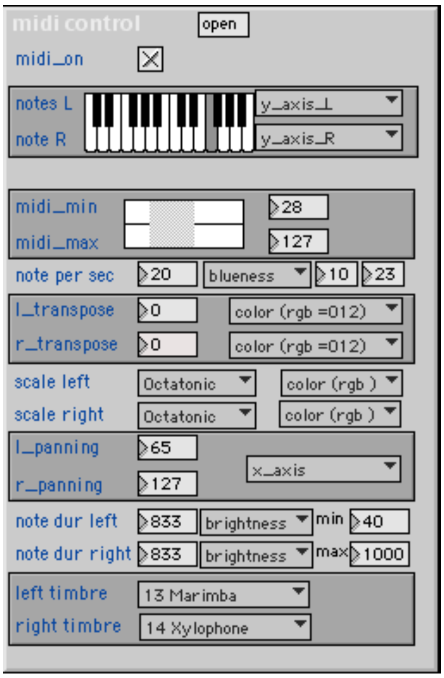


Figure 3. MIDI settings

5.2. Samplers

The same MIDI notes, along with their duration, panning and 'velocity' (amplitude) values are sent to two samplers. These feature, amongst other things, a user selectable number of notes per octave; with 24 notes/octave and a chromatic scale selected MIDI note values 60-84 would result in a one-octave quarter-tone scale. With 12 notes per octave they will simply double the MIDI notes (or substitute if MIDI is turned off) creating additional timbral richness. However, with more notes per octave the resultant pitches will move in similar (but not parallel) motion, creating counterpoint, and adding depth.

5.3. Noise generators

The noise generators (figure 2) use an algorithm by Tom Mays which takes the value for each pixel in a column, and uses that as a value for an equalization curve to filter a band of pink noise. Thus if the image is bright on top and dark below, the resultant audio with consist of mainly high frequencies and fewer low frequencies.

5.4. Granular synthesis

The nature of granular synthesis [6], with all of its various parameters, makes it a natural for utilizing various data coming from an image. In particular grain frequency, amplitude, density, and size are all parameters that respond intuitively to image analysis data, especially brightness and xyz position. *SIHyperI* employs two granular synthesis generators. The first being a cloud generator in which each of the grains are heard more or less discretely (see figure 4).

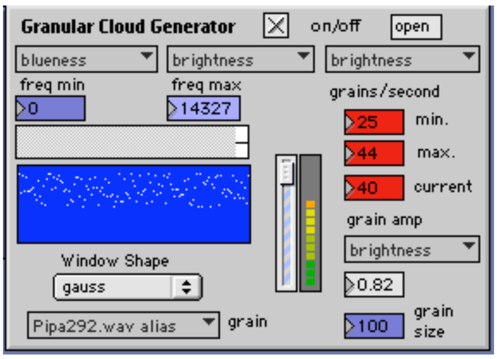


Figure 4. Granular Cloud Generator

The second is a sample granulizer which produces a constant sound as the grains transverse different parts of a sampled sound. In the case of the cloud generator the data is used to indicate the minimum and maximum frequency the cloud will use, where in the granulizer a downsized image produces 32 transposition values for each of the grains used (see figure 5).

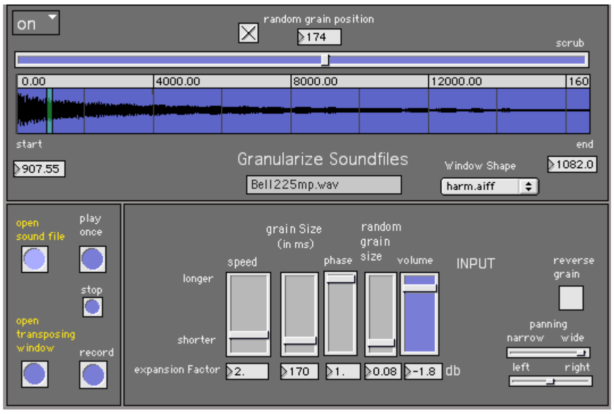


Figure 5. Sample granulator

5.5. Sound scrubber

SIHyperI uses a sound scrubber which is particularly useful when following the xyz motion tracking.

5.6. Sine wave modules

Lastly the program uses two-sine tone generating sound modules. The first uses a bank of 80 oscillators (additive synthesis), each tuned to frequencies determined by the relative brightness values of pixels (in a downsized matrix) of the entire image, or a column/row of a full image. The second, a phase synthesizer, uses values from the image analysis for frequency and amplitude values of two modulating oscillators, that then drive the phase values of two more oscillators in typical phase modulation synthesis.

6. CONCLUSION

Combinations of these 9 sound generators can be combined at will to form complex textures and timbres that sound far more musically interesting than any particular module on its own. This is key, as the motivation is to move beyond mere auditory display towards something more musically satisfying, which usually entails (amongst other things) complexities of timbre, attention to texture, and structure. These latter qualities can be achieved by real-time control, via the mouse, to allow the user to make expressive gestures and to plan structural movements through various contrasting areas of the image being sonified, thus leading to composition, as apposed to the often uninteresting parsing of data. *SIHyperI* can be downloaded (free) from: http://www.hkbu.edu.hk/~lamer/download.htm

7. REFERENCES

[1] S. Barrass, M. Whitelaw and F. Bailes, “Listening to the Mind Listening: An Analysis of Sonification Reviews, Designs and Correspondences,” in *Leonardo Music Journal*, Vol. 16, pp. 13-19, 2006.

[2] A. Hunt and T. Hermann, “The Importance of Interaction in Sonification,” *Proceedings of the 2004 Australian Computer Music Conference*, Sydney, Australia, 6–9 July 2004.

[3] M. Fernstrom, C. McNamara, “After Direct Manipulation-Direct Sonification,” in *Proceedings of the 1998 International Conference on Auditory Display,*

[4] Beilharz, K., and Jakovich, J. Hyper-shaku “Border-crossing: Towards the Multi-modal Gesture-controlled Hyper-Instrument”, in *Proceedings of New Interfaces for Musical Expression* (NIME) Paris, June 2006.

[5] G.Weinberg and T. Thatcher: “Interactive Sonfication: Aesthetics, Functionality and Performance” in *Leonardo Music Journal*, Vol. 16, pp.9-12, 2006.

[6] Roads, C. *Microsound*. Cambridge MA: MIT Press, 2001.

"LISTENING AND REMEMBERING": NETWORKED OFF-LINE IMPROVISATION FOR FOUR COMMUTERS

Ximena Alarcón
Institute of Creative Technologies, IOCT, De Montfort University
XAlarcon@dmu.ac.uk
The Gateway, Leicester LE1 9BH, UK

ABSTRACT

This paper analyses the experience of the networked off-line improvisation "Listening and Remembering", a performance for four commuters using voices and sounds from the México DF and Paris metros. It addresses the question: how can an act of collective remembering [20], inspired by listening to metro soundscapes, lead to the creation of networked voice- and sound-based narratives about the urban commuting experience?. This hybrid sonic performance extends characteristics of the Interconnected Music Network art-form [16] by adding the emotional preparation of participants (through an ethnographic process) and spatialisation. When commuters follow the structure of a journey in the metro through this networked environment, the result is a soundscape involving a counterpoint of voices, recorded voices, and machinery, all together eliciting diverse narratives. Participants re-enact their routine experience through a dialogical relationship with the sounds, themselves and the experience of sharing: a collective memory.

1. BACKGROUND

In 2003, I undertook ethnographic research about memories of soundscape with commuters of the London Underground [1], while reflecting on the intersection between the theory and concepts of soundscape [14] and of collective memory [20]. Through interviews, recordings of journeys, and listening[1] practices, I investigated commuters' feelings triggered by sound within their traveling routine. It led to the creation of an Internet-based *Interactive Sonic Environment* [2], where commuters interact with the soundscape, mediated by a process of remembering[2]. Users experience a non-linear journey in the underground by interacting with commuter-selected sounds of commuters' journeys, which overlap in a multimedia score composed of spaces, categories and subspaces. I proposed "Listening and Remembering" as the main interactive practice and it was initially fulfilled by on-line text responses to the listening experience [1,2].

Within the sonic environment and during the different research stages participants identified 'voice' as one of the most attractive features of this contemporary underground soundscape. They often miss it as a dialogical and expressive instrument during their commuting routine. Thus, voice became an important element to re-enact the human experience in this environment, and it provided a opportunity, in this context, for making Wertsch's idea of "voices of collective remembering"[3] *alive* and *heard*. Wertsch has called "textual community", to a group using a set of cultural tools, language, and objects to produce narratives or "a specific type of community, - namely, one grounded in the use of a shared set of texts."[22] This concept of textual community could be linked with Truax's concept of acoustic community. However, these concepts differ in the sense that Wertsch's community is created by sharing texts and producing narratives while Truax's community shares the acoustic environment, no matter how "this commonality is understood"[15]. Wertsch leaves "voice" as translated usually as "text" in the practice of remembering. Truax, leaves the existence of a community without specific artistic practice for exchanging sounds or feelings attached to them, thus to create community.

My current research project, called *"Sounding Underground"*[4], continues the exploration of sound stories of the commuting experience, with a particular interest in creating a space for commuters' expression and interweaving of memories with their voice. It investigates the links among the soundscapes of the México DF Metro, the Paris Metro, and the London Underground, while emphasizing the uniqueness of each underground transport system. The research consolidates my previous ethnographic methodology of interviewing, recording, listening, and selecting

[1] Listening is understood here in its wider sense as "a complex, multi-layered activity of which hearing is but a part…References, memories, associations, symbols, - all contribute to our understanding of sonic meaning." [8]

[2] According to James Werstch the process of remembering is related to the "textual resources" utilised in talking or writing about the past, to how those texts could show an image of the view of others who have utilised them, and, eventually, to the addition of the voice of whoever is recalling. [21]

[3] "Voices of Collective Remembering" is the title of James Werstch's book where concept of collective memory, textual communities and the process of remembering, are explained. [20]

[4] The project originally was called "Linking soundscapes via commuters' memories". It has evolved during the course of the project, and the new working title is "Sounding Underground".

sounds, and introduces improvisation as a performance activity.

2. CONTEXT

The soundscape of London Underground (LU) is a rich mixture between sonic, symbolic, and social contexts. During the journey, repetitive sounds of machinery, reverberant spaces, contrasts between confined and open spaces, and the everyday interaction of people with a purposeful activity, have created sophisticated sound textures and rhythms, full of micro-events, from which to approach both unconscious and conscious states of commuters' bodies and minds [1]. During the journey, a powerful connection with the individual's own life is provoked by this magical and sublime space. Williams refers to the aesthetic concepts of sublimity and fantasy as "invented to express the emotional power of subterranean environments, a power not encompassed by the traditional aesthetic terminology of beauty and ugliness"[24]. Sublimity in underground environments "celebrates ambivalence" depending on "the delicate equipoise of conflicting emotions" [25]. This experience has inspired humans and their fantasy since the first attempts of exploitation and industrialization of underground environments [23]. Symbolically the underground journey can represent death [6, 26], the disconnection from our known environment and natural light; also the journey can be thought of as re-birth, the connection to ourselves in a space of detachment, in a "womb-like" space. In the *isolation* of a public space, guided by the rhythm and movement of machines and people, conditions are created for *digging* into profound human emotions. Comparing the LU's commuters' experience with the one of commuters in México DF and Paris, it is noticeable that socially, each city's underground unveils tensions, as regards class, gender, social status, political control, identity and belonging to a particular community. These relationships are immersed in the cultural appropriation of a technological environment [23].

In the ethnographic process, seventeen volunteers in México DF and seventeen in Paris participated in an individual process of remembering that started with an interview about their commuting experience, and in particular about the remembered sounds and feelings associated with it. Secondly, each participant recorded a journey in the metro, using binaural microphones, experiencing an awareness of details of the soundscape not previously perceived, as well as of their own role within the soundscape.

During the recording, the researcher offered the participants small papers containing messages. These invited them to interact, think, or perceive in particular ways, sonic and memorable aspects of their journey, creating a playful intervention. Some of the participants became engaged with this interaction, commenting, or reflecting in silence, intrigued by the chance situations that occurred when the message was read. Others continued with their routine, as was also their option, while recording their journey.

In a subsequent session, participants listened to their journeys and selected and edited[5] the sounds that they considered most meaningful. The only constraints imposed by the project were the duration of the sounds: 15" at the shortest, and 1'30" at the longest.

The improvisation was the last stage in this series of workshops during the fieldwork. It functioned as a compilation of the entire experience of remembering. The previous fieldwork activities were considered essential emotional preparation for the improvisation.

3. NETWORKED IMPROVISATION

Networked music performances are mostly performed by trained musicians and performers, as such performances traditionally demand musical and technical skills [19]. Barbosa, in his classification of "computer-supported collaborative music", points out that "a spontaneous [free] improvisational approach" [3] is suitable in an Internet context, because of the characteristics of this medium, and introduced the term *Shared Sonic Environments* to describe a kind of performance where people can "participate in a public event by manipulating or transforming sounds and musical structures or by simply listening to music created collectively."[3] This "public event" implies the participation of non-musicians and suggests the creation of acoustic communities via the Internet, going "beyond the enhancements of existing acoustic communication paradigms"[4]. On the other hand, listening in an electroacoustic musical space to environmental sounds implies the expansion of the listening experience, "transporting the listener beyond the listening space or creating a large space for the listener to inhabit"[12].

Thus, the "Listening and Remembering" improvisation is described here as a *Shared Sonic Environment* that extends the perception of space by the use of spatialisation of soundscape sounds and the voices of commuters expressing memories. At the same time, it uses an on-line web 2.0 model of sharing, introducing tags as a resource to identify, track and trigger participants' recordings. Most of the participants had no performance experience. Their knowledge of the soundscape and what this means for their lives were the most important elements brought into the performance space. In terms of geographical location, the improvisation is off-line, meaning a co-located system[6] where participants' computers are linked by a local area network, and share the same

[5] I gave a tutorial in Audacity that taught how to excerpt sounds and make fades.

[6] A co-located system is defined by Barbosa [3] as a networked performance in which participants share the same physical space. He derives this definition from Rodden's Computer Support for Cooperative Working (CSCW) geographical nature dimension [10]. The term helps to define the "off-line" character of this networked improvisation, meaning also that it is not connected to the Internet.

room. It has been implemented first off-line (co-located), in order to establish a model for what could be created on-line (remote), thus including commuters from the other metros.

In order to describe different aspects of the improvisation, I employ Weinberg's theoretical framework for Interconnected Musical Networks [16]. The four main aspects that she describes—goals and motivation, musical content and control, social organization and perspectives, and architectures and topologies—are useful for understanding this performance experience as "an interdependent art form", while they simultaneously present a foil for the ways in which it represents a new approach to networked musical experiences.

In terms of 'goals and motivation', it can be understood as an *exploratory* and *process*-centered network. For Weinberg, exploratory networks "do not impose specific directions or goals for the participants" [17]. This improvisation maintains the narrative structure of an underground journey. I proposed to groups of four participants, whose computers were networked, that they listen to a journey based on sonic excerpts of their own journeys, organized and triggered according to such a narrative, by means of *environments* (i.e. Street, Entrance, Tickets, Corridors, Platform, Carriage), and sonic intervals called *events* (i.e. steps, amplified voices, trains arriving, opening and closing doors). The sounds were diffused without any transformation via five loudspeakers that surrounded the participants (see Figures 1 and 2): participants were each allocated an individual speaker through which their memories would be heard, while environments and events were deployed to multiple speakers, presenting a wider acoustic space. When they wanted to express a memory[7], they were able to record it via the microphone, and each memory was given a name, or tagged (see Figure 3), to be visible and available for playback on the other three participants' screens. The responses of the participants were unpredictable and the space was free for them to express as they wished.

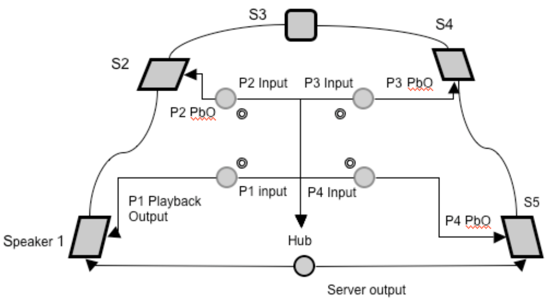


Figure 1. Topology.

[7] Memory, within this context, is understood as any form of expression made with the voice, with or without words, that has been triggered by the soundscape.

Figure 2. Participants improvising in Paris

Figure 3. Server patch in improvisation in Paris.

Thus, the 'musical content' we hear during the improvisation is a mixture of commuters' journeys, including counterpoint between human and machinery sounds, and the responses that these provoke. I, through the server software, control the amplitude and triggering of pre-recorded sound excerpts. The commuters, through the client software, control the recording and playback of the voices. As there is no transformation of sounds, all sonic events and their combination depend on human input. All these aspects define two actors in terms of 'social organization and perspectives': the leader (i.e. the researcher) and the participants (i.e. the commuters).

The technical setting, created by the composer and Max/MSP programmer Peter Batchelor, consisted of five computers connected via Ethernet. Two Max/MSP patches were designed: one for the leader of the improvisation - the server - (see Figures 4, and 5), and one for the participants - the clients - (see Figure 6). Their interfaces were designed to be colourful and engaging while being as intuitive and easy to use as possible, in order to enable fluency within the improvisation. The server acted as a hub for all audio activity, receiving controller data (record/playback triggers) from the clients.

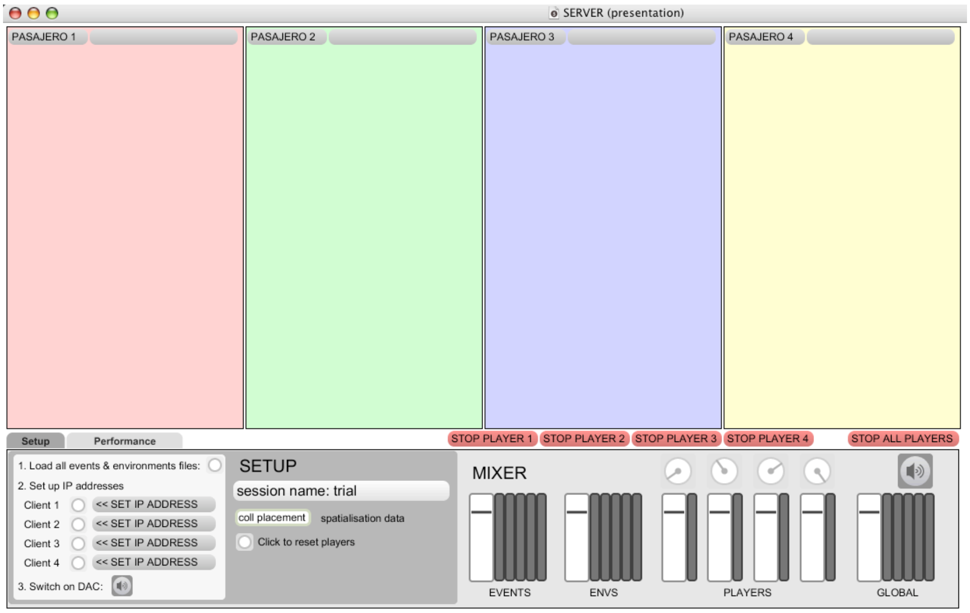


Figure 4. Server patch with features of the Setup.

Figure 5. Detail of features of the “Performance” tab.

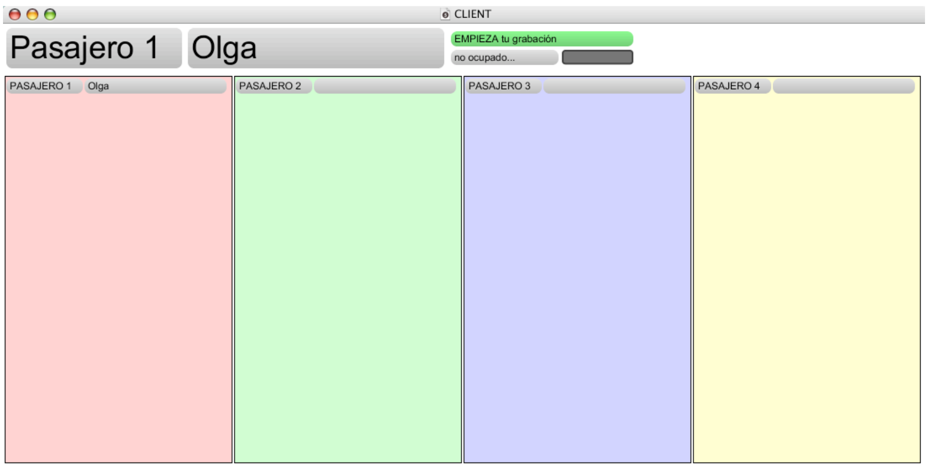


Figure 6. Client patch.

These aspects configure the network with an 'architecture' of a *synchronous* centralized network. Even if participants are stimulated by the sounds triggered by the leader, each one is free to record and playback their memories at any time, regardless the actions of the other participants and the leader. However they can influence rather than "modify [technically speaking] the music of their peers while it is being played"[18]. Spatialisation of soundscape and the use of participants' memories, are other important elements to add in its 'topology'.

Thus, in this hybrid sonic Interconnected Networked performance, interdependency, for musical and social achievements, relies not only on Weinberg's concepts, but also on the actions driven by listening and remembering, and by the connections established between the participants about a common sound structure: the soundscape. Sonic engagement is intrinsic to the process of remembering, in this context. It is strongly based on the ethnographic process, and on the emotional attitude of commuters toward this particular soundscape.

4. RESULTS - TYPES OF INTERACTION

In México DF, four groups of commuters improvised[8] in a large, non-isolated space. Their voices were amplified while recording. In this first improvisation, the amplitude of the playback of their voices differed from the amplitude of recording, which was a technical issue that needed improvement. In Paris, three groups of participants improvised[9] in a smaller space, with a more controlled acoustic. In order to improve the mentioned technical problem that occurred in México DF, participants in the performance in Paris - recorded without live amplification (i.e. 'secretly'), and played it back over the loudspeakers, thus discovering only later what the other participant had recorded. These two approaches defined an important difference between these two performances. Between México and Paris there were also certain changes in the server interface. Access to all the sounds recorded in the journey was possible in Paris, in contrast to the experience in México DF, where only the sounds previously chosen by the four participants were available. The change of this feature facilitated the leader's role in triggering the environmental sounds, enabling the mixing of all possible sounds provided by participants.

In both cities, the mixture of metro soundscapes and the intensity of multiple voices remembering created diverse narratives and stimulated the sharing of the commuting experience. These narratives varied according to cultural references and individual expressive and communicative interests. However, there were similarities in their narrative expectations and in the connection with the space. Participants switched between the "real space" (as if they were in the metro), their personal "memory space" (the memories that they share with their voices), and the "performance space", where the metro is perceived as a background space. These spaces are *Shared Sonic Environments* that bridge the experience of this acoustic community.

Performance forms that participants experienced can be divided and explained through the following typology:

5.1. Reflections about self and the space:

Participants made comments about the sounds that they were listening to, which also provoked responses from other participants. For example, in México, they expressed feelings related to the future of the city and the role of the metro: a shelter, a hub of popular culture, a chaotic city and its informal economy, and the role of the vendors. There were opinions about likes and dislikes of certain sounds and the role of the music in the journey. Some childhood memories were expressed, and also a memory of water. In Paris, comments included the weariness of the routine and the questioning and affirmations of Paris as a city. "Paris the beautiful city? Why Paris? It could be another city." (BN, my translation) "*This* is Paris and not the Eiffel Tower. This is Paris." (MM, my translation)

5.2. Sonic play:

Participants found musicality through abstraction, transforming everyday verbalization, through poetic sounds and words going beyond of the meaning of the words: "Avanzo, avanzo, avanzo, hacia abajo, hacia abajo, hacia abajo" (AI), "Des-cendre, des-cendre"(LW). Onomatopoeias were also common in the expression of the participants. They often imitated the sounds of machinery, which also lent rhythm and sonic textures to the experience, such as the case of airy sounds: "Shhhhhh, silence, silence, shhhhhhh"(MM), "Pfffff"(AS), or a screeching sound. In México, participants took advantage of one commuter's foreign language to imitate or play with the situation. The possibility of the immediate repetition of their own recordings tended to dominate the performance as a way of finding mixtures and rhythms. In some cases, this followed the rhythm of the metro, and in others it created a completely different soundscape, in a DJ fashion, which left the metro environment in the background. Within the journey, the sonic space of the carriage as it travels through a tunnel was a special place of engagement and intervention from participants playing with rhythms, and repetitions.

5.3. Incorporating everyday life:

Snippets of conversations between known or unknown people are common. Stories range between typical salutations: *"Doña Rosa dígame…"* ("Mrs Rosa tells me…", MR), "Hello Syl-vie" (GC) to non-sense interventions, which create interesting and humorous stories: *"esa mano es tuya?"* ("is this hand yours?", OC). In México a participant took advantage of a mobile

[8] At the Centro Multimedia, CENART, on the 26th of July, 2008. In the Sala Manuel Felguerez.

[9] At the Plate-forme Technologique, Maison des Sciences de l'Homme, MSH, Paris Nord, on the 24th of January, 2009.

phone call that he received during the performance, including it spontaneously in the improvisation. The other participants followed the conversation by making short comments that made the situation more humorous.

5.4. Story-telling / enacting:

Dramatic story-telling could take a conversational manner *"Oh la-la, suicide?"(GC), "No, ce n'est pas un suicide. C'est un chien, un chien sur la voie" (SD). (*"Oh dear, was it a suicide?" "No, it was not a suicide, it was a dog, a dog on the railway").

5.5. Imaginary spaces:

Participants created imaginary spaces into which they invited each other, for example, to dance "a waltz". Here time and space diverge from the routine, and this is prompted by the music that they are listening to. Music plays the role of linking with other spaces and times, and participants take advantage of this situation.

5.6. Communication Media:

Participants sent political messages, using the improvisation as a diffusion medium: *"Espero que la gente vaya a votar este domingo. No podemos dejar que PEMEX se quede en manos extrañas"* (EV). ("I hope people go to vote this Sunday. We cannot permit that PEMEX [Mexican Petroleum Company] is managed by foreign hands").

5.7. Just listening/intimate experience:

The feeling of not being obliged to record or play back was an unexpected option here. The setting, with a microphone and a big screen, can be intimidating for many people and particularly for non-performers. In the improvisation, some of them decided to simply listen. In Paris, one group was concerned with intimacy, and their memories were not played back often. A post-performance reflection by them was about respecting others' voices, or being self-conscious. This touched the boundaries of intimacy in the performance experience. They talked to themselves about dreams, the particular acoustic of the metro, and the anger felt about a robbery in the metro.

This typology reveals the performative strength of the improvisation, with its interweaving of narratives as a collective memory of an underground space. It is possible to see in these stories elements suggested by Williams, such as the aesthetics of underground environments moving between the sublime and the magical, derived from the realism given by the contemporary experience. Also, rhythmic relationships between machinery and the body are present in the performance, resembling experiences felt by London Underground commuters in previous research [1].

There are also perceived transformative possibilities for participants and their stories: transforming tragedy in daily life misunderstandings, exchange between unknown languages, and imagining a ballroom in a metro, amongst others. It's the interweaving of these stories, thanks to the networking structure and the switching between spaces, that makes it a special networked performance with extra-musical elements. An interconnection follows multiple purposes and causes, from the most individual to the most collective one. In the improvisation, we listen to "the unpredictable turns of chance permutation, the meatiness, the warmth, the simple, profound humanity of beings that bring presence and wonder to music."[9]

5. DISCUSSION WITH PARTICIPANTS

The variety of possible expressions combined with participants' diverse expectations, are considered part of the processes of remembering. The general feeling of the participants, who were listening and making sounds, was that it was an enjoyable yet strange experience. The experience of sharing in oneway or another was successful, and the making of linear or non-linear stories is part of the richness of this soundscape environment, transcending time and space. Conscious decisions of sounding and playing voices back create powerful moments of connection with the other participants in the exercise of playing together, creating tensions, harmonies and also disconnections.

Technical issues, such as the level of different voices, were discussed as potential obstacles in the performance. This issue defines whether participants are in the space of the metro or in a performance space. Some participants suggested it would be great if they were able to trigger their own (previously recorded and selected) sounds and not just their voices. Also, they would like to have a recording of the full improvisation for their records.

Acoustic conditions, such as the size of the room, were important and influenced the performances. The main obstacle described for participants in Paris was the fact they needed to write a "title" before recording. That interrupted the fluidity of remembering, recording, and playing back.

Expectations for people who are musicians[10] were different than for people who are not. Being able to control the sound amplitude and spatialisation were important issues for the former group. For other participants, narrative, in terms of story telling, imitation and repetitions, were some of the important, enjoyable issues.

Although it is important to highlight that participants in these two cities were already engaged with their own sounds and the process of remembering, some guests did join the improvisation. In México, this type of performance also took place in Morelia, a city four hours

[10] Some of the participants were musicians. In the call for volunteers the only requirement to participate was to be a commuter, regardless of their sonic or musical background.

away from the capital[11]. Volunteers improvisers who were not metro commuters (children, young people and adults with instruments) interacted in the setting, proving that the metro soundscape could engage people who are not familiar with this acoustic environment. Although improvising with instruments was not the purpose of this research, this kind of experience alludes to jazz forms. In the same vein, voices that play with repetitions and machinery sounds, suggest the influence of train and other mechanical and digital sounds as inspirations for the creation of musical and poetical forms, as has been the case in other contexts described by music scholars from soundscape and jazz history [7, 11, 13].

6. CONCLUSIONS AND FUTURE PLANS

The researcher believes that the improvisation "Listening and Remembering" is a catalyst for commuters from diverse backgrounds to perform an act of collective remembering about a shared soundscape: the underground public transport system. They use their voice as a source to *counter-point* the underground transport system's soundscape, in a broad sonic and musical experience. This opens scenarios to experiment with narratives and aesthetics related to the underground environment and to the rhythmical relationships between machinery and commuters' bodies. For example, the tunnel is certainly an important place (sonic and symbolic) that stimulates people's feelings and the sharing of experiences and sounds. It is also a space to link participants in the act of remembering. Here, poetic moments are powerful and take place within repetition, the rhythms of voices and machines, and the imitation of machines' sounds.

In light of Weinberg's proposed characteristics of Interconnected Musical Networks, this improvisation offers additional elements to the genre, such as spatialisation controlled by two sources (the leader, and the participants), and an emotional preparation of the commuters that is provided by the ethnographic process. These elements offer strength and singularity to it, and interdependency goes further than traditional musical expectations, expanding the notion of *Shared Sonic Environments* when participants locate themselves in either the "performance space", the "memory space" or the "real metro".

In terms of location, the future of the improvisation is two-fold: as an off-line performance, and as an on-line improvisation that integrates the elements developed in this experience. The off-line performance needs to improve conditions of spatialisation (soundscape and voices) to define the space where each participant would like to be. This can be done by including controls for amplitude, as previous improvisation experiences, such as Dilon's *jam2jam* project [5] with children and Weinberg's *Voice Networks* [19] with novices, have implemented. However, including more controls could also make this interface less simple for amateurs.

I do not intend to narrow the experience by establishing fixed characteristics of the space; it has been proved that the setting works for people with different expectations and skills. It also motivates the exploration of creativity to structure the performance in different ways. For this exploration, participants need time to get used to the setting, to be involved within the journey, and to relax, in order to start the creation of sonic narratives. This improvisation could also take the form of an installation environment off-line, where time and space are flexible for the participants during the sharing of their memories.

The option for participants to trigger their own metro sounds is important, as is the recording of the improvised sequence. The first, in order to decentralize the triggering of sounds of the soundscape; the second, as a publishing act and signature of authorship.

The online version of the improvisation requires technological development from scratch. Probably already existing packages and jamming environments[12] could be used to experiment with it. However the ideal is that the technology will be developed within the *Sounding Underground* environment.

While on-line environments can offer the feeling of being in one's own world, allowing uninhibited sounding expressions, the offline improvisation could also offer to participants a clearer statement about what to expect, particularly for the ones who are concerned with intimacy and the enjoyment of their own sonic space.

A combination of on-line and off-line improvisation might also be suitable for this project, having performance spaces in each city while interconnecting between cities, e.g. using an access grid. As noted previously, text entry is an obstacle in the performance space. However it is an important feature when working on-line, because it is integrated within the language of the Internet. Research in this area is needed to improve the performance space and to clarify for participants the role of the text. Future performances are planned using the sounds of the three metros involved (i.e. including London). It could ultimately be performed in any city.

The mixture of soundscapes, and feelings expressed in different languages and other voice expressions could integrate the three cities' soundscapes as a place symbolic of contemporary urban culture. And this could be offered to and performed by a variety of audiences, engaging them in the process of "Listening and Remembering". It is an improvisation that invites commuters to engage in a transformative process of sharing memories with their voices, widening for them the musical and poetic potential of this singular urban experience: the underground commuting routine.

[11] With the support of CMMAS (Mexican Centre for Music and Sonic Arts).

[12] Such as nin-jam http://ninjam.com/, or e-jamming http://ejamming.com/ .

7. ACKNOWLEDGEMENTS

These improvisations would not be possible without the rich collaboration of Peter Batchelor, and the technical support from: the Institute of Creative Technologies (IOCT) and the Music, Technology and Innovation Research Centre (MTIRC), from De Montfort University, in Leicester UK; the Centro Multimedia del Centro Nacional de las Artes in México City; the Centro Mexicano para la Música y las Artes Sonoras (CMMAS) in Morelia; the Plate-form Technologique, Maison des Sciences de l'Homme, Paris Nord, in Saint Denis. Special thanks to Ron Herrema from the Music Technology and Innovation Research Centre (MTIRC), De Montfort University, for his technical and production support, and to all commuter participants in México DF, Morelia and Paris, for their commitment. This activity is part of an ongoing research project *Sounding Underground – Linking urban soundscapes via commuters' memories*, an Early Career Fellowship 2007-2009 funded by The Leverhulme Trust.

8. REFERENCES

[1] Alarcón, Ximena (2007). "An Interactive Sonic Environment derived from commuters' memories of soundscape: a case study of the London Underground". PhD Thesis. De Montfort University, Leicester, UK

[2] Alarcón, Ximena (2007). "Interactive Sonic Environment: London Underground" http://bartleby.ioct.dmu.ac.uk/~xalarcon/project/interactive.html

[3] Barbosa, Álvaro (2003). "Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation" in *Leonardo Music Journal*, Vol. 13, pp. 53-59. Cambridge, MA: MIT Press. p. 57

[4] Barbosa [3] p.58

[5] Dillon, Steven C (2007). "Examining meaningful engagement: Musicology and virtual music making environments". In Proceedings ISLANDS – Conference of the Australian and New Zealand Musicological Societies, Brisbane.

[6] Eliade, M. (1960). *Myths, Dreams & Mysteries*. Great Britain: Harvill Press.

[7] Murray, Albert (2000). *Stomping the Blues*. United States of America: Second Da Capo Press Edition 2000, (see pages 118-126)

[8] Norman, Katharine (1996). "Real-world music as composed listening" in *Contemporary Music Review* 15(1):1. p 2.

[9] Oliveros, Pauline (1999). "QUANTUM Improvisation. The Cybernetic Presence" in *Musicworks*: No. 75. Toronto: Musicworks Society of Ontario: 14-20. p. 17

[10] Rodden, Tom and Blair, Gordon (1991). "CSCW and Distributed Systems: The Problem of Control" in *Proceedings of the Second European Conference on Computer-Supported Cooperative Work*. Amsterdam, The Netherlands: Bannon, L., Robinson, M. & Schmidt, K (Editors).

[11] Schafer, R. Murray (1994). *The Soundscape: our sonic environment and the tuning of the world.* Rochester, Vermont, Destiny Books, (see pag. 63)

[12] Smalley, Denis (1996). "The listening imagination: Listening in the electroacoustic era" in *Contemporary Music Review*, 13:2, 77-107. London: Routledge. p. 92

[13] Thompson, Emily (2002). *The soundscape of modernity: architectural acoustics and the culture of listening in America, 1900-1933*. London, Cambridge, MA:MIT Press, (see pages 130 – 144).

[14] Truax B. (2001). *Acoustic Communication*. Westport, CT: Ablex.

[15] Truax [14] p. 66

[16] Weinberg, Gil (2005). "Interconnected Musical Networks: Toward a Theoretical Framework" in *Computer Music Journal*, 29:2, pp 23-39. Massachusetts Institute of Technology.

[17] Weinberg [16] p.32

[18] Weinberg [16] p. 34

[19] Weinberg, Gil (2005). "Voice Networks: The Human Voice as a Creative Medium for Musical Collaboration" in *Leonardo Music Journal*, Vol. 15, pp. 23-26. Cambridge, MA: MIT Press.

[20] Wertsch, James (2002). *Voices of Collective Remembering*. Cambridge: Cambridge University Press.

[21] Werstch [20] p.17

[22] Wertsch [20] p.62

[23] Williams, Rosalind H. (2008). *Notes on the Underground: an essay on technology, society and the imagination.* Massachusetts Institute of Technology Press.

[24] Williams [23] p.17

[25] Williams [23] p.85

[26] Wolkstein, D. and Kramer, S. (1983). *Innana: Queen of Heaven and Earth*. New York: Harper & Row Publishers Inc.

Approaches to Notating Electroacoustic Music with Live Instruments: Preliminary Results
ACMC 2009, BRISBANE, QLD, AUSTRALIA

***Leah Barclay**, PhD Candidate*

Queensland Conservatorium of Music, Griffith University

info@leahbarclay.com | www.leahbarclay.com

ABSTRACT

This paper provides preliminary results from a broad study on the diverse approaches to notating the infinite possibilities of environmental electroacoustic music. This broad study includes a series of case studies on works that integrate environmental soundscapes with electroacoustic music and aims to explore the diversity of methods composers are using today. This paper is primarily reporting on a survey that aims to outline the challenges and possibilities from the perspectives of both the composer and performer and identify methods to engage a wider interest and accessibility in performing mixed music today.

INTRODUCTION

The term soundscape composition defines a genre within electroacoustic music that predominantly uses environmental sounds as source material. As this is one of the fundamental features of electroacoustic music, further ascetical factors are applied to soundscape compositions; such as the relationship to the sonic environment. This particular aspect highlights the fact that sound materials will have a relationship to their source and rarely be perceived in a completely abstract manner. This is a key factor in differentiating soundscape composition, as a large quantity of electroacoustic music prides itself on the Pythagorean "Acousmatic" aesthetic; the apprehension of sound without relation to its source.

Despite this clear difference, soundscape composition still shares many synergies with the field of electroacoustic music, a genre with a particular theoretical strategy developed from the Music Concrete and Electronic Music traditions. The clearest definition of Electroacoustic Music states that the genre is "music in which electronic technology, now primarily computer-based, is used to access, generate, explore and configure sound materials, and in which loudspeakers are the prime medium of transmission" [1]. The use of natural sounds, enriched by technology, allows us to claim that this is the first musical genre ever to place under the composer's control "an acoustic palette as wide as that of the environment itself" [2].

This liberating facet is one of the greatest attractions for the composers as it releases a musical language of infinite possibilities. Despite this liberation, it is the underlying cause into why perception, valuation and particularly analysis of electroacoustic music are extremely difficult tasks. The existing literature has a strong tendency to be extremely complex with demanding terminology obscuring valid theories and the genre has faced considerable negligence predominantly due to contentious issues associated with absent visual referents and infinite sonic possibilities. Western music theory is yet to develop the means to confront such a wealth of sound materials and despite many attempts, a common language or method of analysis does not exist.

THE ACOUSMATIC SCORE

After publishing several papers on the issues of language in electroacoustic music, and an Honours dissertation on Electroacoustic Analysis the author subsequently developed a teaching tool called the 'Acousmatic Score'. This resource employs theories from Albert Bregman, Denis Smalley, Trevor Whishart and Simon Emmerson and enables young composition students to engage with electroacoustic music in an accessible way. The intention of this project was to create a tool to make the genre more accessible and engage a wider audience, and although this addresses some of the obvious problems at a fundamental level, it fails to make a significant impact on truly engaging a wider audience.

This key issue is still driven by exposure and education and in the context of soundscape composition with live instruments it is imperative that it is accessible for performers. This forms the key motivation for a study on the diverse approaches of notating the infinite possibilities of environmental electroacoustic music, with the aim to outline the challenges and possibilities from the perspectives of both the composer and performer and ultimately identify methods to engage a wider interest and accessibility in performing mixed music.

PERFORMING ELECTROACOUSTIC MUSIC

Performing electroacoustic music comes with a range of opportunities and challenges, with most of the challenges revolving around the notion of visual representation; the score and documentation of the compositions. The existing literature frequently documents debates on the strategies concerning the fusion of live instruments and electroacoustic sounds and the way performers relate to electroacoustic music has become a key concern for composers and researchers. In the context of this study we are specifically looking at electroacoustic music featuring soundscapes with live instruments, within this genre a further division is necessary to discuss fixed and interactive works.

'Fixed' is used to define compositions where the soundscape exists in a fixed form on a CD, DVD or sound file for the performer to accompany. 'Interactive' describes compositions with live electronics, potentially affecting the sound in real time or providing the performer with the ability and control to trigger sound files and different elements of the piece.

The 'fixed' and 'interactive' subgenres come with a range of contentious issues that have contributed towards the genres neglect in contemporary music. Many of these again revolve around issues of accessibility and visual representation. The ultimate outcome is a new model that allows a balance between the extended use of instrumental sources and the infinite possibilities of electroacoustic music in an accessible and sustainable framework.

The role of the score is a critical issue in this context, as early as the 1940 composers were calling for scores to be represented in timbre and texture. The issue of documentation is also a prevailing factor, prior to electroacoustic music the score was essential to guarantee that a composition would survive, and that it could be performed without the composer's presence. Although research has suggested technological innovations provide similar frameworks to the 'score', the fact is a MAX/Msp patch or an Acousmographe diagram is simply not as sustainable, detailed or accurate as traditional music notation. This forms a debatable subject in itself, but unfortunately electroacoustic music is being lost daily purely due to the advancement of technology, ironically the very tool that brought it too life.

LIBERATING THE STAVE

The proposed concept of "Liberating the Stave" is ultimately about developing a unified, accessible and sustainable method for notating electroacoustic music with live instruments. There are multiple research intentions embedded in this concept, the first being to develop an accessible method for the composer, with a particular emphasis on emerging composers engaging with electroacoustic music for the first time. The second intention is developing an accessible method for the performer, this is a key factor, if it is not accessible and engaging it simply won't live past the first performance. The final intention is to develop a sustainable method; specifically not dependent upon technology.

In recent years, new hardware has allowed us pioneering ways to integrate instruments and technology, but due to the complexity it's often a requirement the composer is present to bring the piece to life. When developing models for interactive composition, it's essential the balance between the use of computer-based tools and the relationship between performers and the electroacoustic sounds is thoroughly considered. The technology should be a key to expand the expressiveness of musical language not an obstacle in the performance. So in order to facilitate the intended sustainability the 'art' must always take priority over the technology.

METHODOLOGY

The methodologies for this study are very reliant on qualitative research methods. The first is compiling case studies on significant works in the field and documenting the methods that they use to create visuals representation of their unique sonic languages. The second is a series of comparative analyses between different performances of key works and the third is a survey of performers and composers. Each methodology has produced positive results and while the first will be briefly discussed, this paper is focusing on the preliminary results from the survey.

The case studies have been a key in establishing the most prominent methods and approaches from the perspectives of the composer. Graphic scores are undeniably the most popular notational methods for composers working in this domain; this is not surprising considering western music theory is yet to make any significant contribution to the field. Works such as Ros Bandt's 'Blue Gold' and Leah Barclay's 'Wolf Rock' use a combination of western notation, graphic notation and text information to convey the message to the performer. In this context, the score has been created purely for the purpose of the performer, but in the vein of traditional music notation, now serves as a form of documentation and a tangible tool for analysis. Many of the most prominent mixed works take a similar approach to notation, but the validity of the notation is not necessary a direct result of a successful performance.

Composers may use a graphic notation style open to infinite interpretation, resulting in a work that will be ephemeral and reliant on the performer's improvisation abilities. In soundscape composition, this is often a very liberating and desirable factor, but it means the score is serving more as inspiration as opposed to specific directions.

The comparative analysis methodology is therefore used to test the validity of various notational methods and explore the key issues of interpretation with graphic notation. Though this aspect of the study is revealing some encouraging results, the survey is the most imperative at this stage to begin collating the fundamental data. This survey specifically focuses on the challenges and possibilities of performing electroacoustic music with live instruments.

The survey brings together an array of responses from Australia, New Zealand, Canada, Britain, Scotland, France, Korea, Japan and India with a variety of professional performers and composers, some who regular perform and commission electroacoustic music and some who refuse to perform electroacoustic music. This study aims to document this area in an honest and candid approach, therefore the identity of the performers is concealed, revealing only their nationality and instrument.

The intention of the survey is to establish why musicians do or don't perform electroacoustic music, identify the perceived challenges and opportunities in performing electroacoustic music, establish a perception on the fixed and interactive debate, identify the most desirable methods for scoring electroacoustic music, and finally to establish an ideal scenario for the score from the perspective of the performer.

RESULTS

The majority of the motivations for performing electroacoustic music are obvious and bear no surprises. "It's invigorating, it's liberating, it's exciting", is a common strand and naturally revolving around the fact that it offers new challenges and opportunities. The explanations for avoiding electroacoustic music have the same predictable results; "it's too challenging, technology is too unreliable, and it's not accessible to tour" being the most prominent.

A Canadian 'cellist who has commissioned a lot of electroacoustic music said "interactive electroacoustic music can continuously vary its response and I can't predict how the computer will react, which is very stressful, when I have a new piece I often have a very organised score for my part and no visual representation of the electroacoustic part so it's this whole other world I have to navigate my way through and it would make it a lot easier to see what is going to happen." This statement from an experienced performer and advocate for electroacoustic music is a clear justification for notating electroacoustic parts. In this context traditional western notation would not be sufficient; the composer could use a combination of graphic notation, sonograms, spectrograms or waveform.

Many performers rate the intimidation of technology as a key deterrent in electroacoustic music, with approximately 87% stating they would be more confident performing electroacoustic music had they had more exposure to this technology in their undergraduate music studies. A New Zealand flautist remarked that she finds "working with the computer very reliable and intuitive, I can do anything with my instrument and the computer won't play a wrong note, it won't go too fast or too slow, it won't lose its place in the score, it won't miss an articulation, and it won't have trouble breathing – it's much better than most performers I've worked with!" She suggested the idea of the composer being more accommodating to the performer is vital, stating "we're here to translate your vision into sound so the more information on the score the better, I find a mix of graphic notation and western notation works well and I like as much information as possible about the electronic part on my score, which is rarely the case." The same performer suggested graphic notation was the most effective method for electroacoustic music, but it's vital that the part is extremely accurate in relation to timing.

The survey questions in relation to the interactive and fixed debate collected a diversity of responses, some quite unexpected for our current music climate. When we consider interactive music, the compositional strategies which include any kind of software for interaction should avoid dependency on the software. It's important to consider 'fixed' music has been active since the conception of electroacoustic music. In the typical performance for an electroacoustic work with instruments from the 1960s onwards performers followed a strict framework with the 'fixed' or prerecorded part. It is quite obvious that this prevents a sense of freedom and a natural interaction between the performer and the electronic part, hence the progression to interactive electroacoustic music.

In the context of fixed music, even highly innovative and complex parts will never be affected or modified by the performers' decisions. The technological developments in the progression to interactive music now create infinite possibilities of what we can do in real time, this also facilitates a sonic dialogue and interactivity between the performer and electronic part. Interactive electroacoustic music is much more flexible and obviously allows the performer a greater degree of freedom, but as the diverse survey results reveal, this does not necessarily mean it is the preferred method.

Although 'interactive' electroacoustic music is more flexible, it is important to take into consideration the requirements of the composition, and certainly not employ interactivity for the sake of it. The intention must always come from within the piece. Performers also suggest when using software such as MAX/msp, an understanding of the patch and exactly how it works is a highly desirable factor.

Writing a MAX/msp patch is a complex notational language in itself, and many composers take an extremely different approach. The notion of creating a graphic score and a full written documentation of the patch creates accessibility for the performers and it also creates the sustainability for the composition, if a full written documentation of the patch exists it can ultimately be recreated in other platforms when the next software advancements materialize.

The responses from the surveyed performers all offer valid and diverse opinions on this subject. An Australian percussionist said she loves interactive music, responding with; "knowing I have the Freedom to work with the part I feel like I'm so much more involved in the piece and engaged with it – playing with a fixed track is like playing with a backing track I feel like I'm just playing along and feel quite restricted and trapped in the part – even just a couple of triggers are enough to give the performer that sense of control."

This comment is a clear indication that the control factor in interactive music is highly appealing.

A prominent British electroacoustic composer provided some strong responses to interactive music in the survey, stating that he believes “MAX/msp and interactivity is not at all sustainable, the works disappear, we have no documentation and it’s a frightening position for the composer when we leave our selves so vulnerable to technology, I am not interested in MAX and can explore all I need with my fixed resources in the studio with much better quality.” An acclaimed Australian performer and composer said she will not play interactive music commenting “you have no control, the quality is often terrible and things always go wrong”. She would much rather play with a professionally mastered CD, or ‘fixed’ part and know that the piece will work without the unpredictability and stress of interactivity.

A Chinese violinist said he only plays interactive works as the spontaneity and excitement is what drew him to the genre. An Australian Clarinetist said he would rather play fixed works as they are so much easier to tour and come with less technical problems. The responses to the interactive and fixed debate continue with inclusive results, overall 38% of the surveyed performers prefer working with fixed composition, 33% preferred interactive compositions and 29% were happy to work in both areas. The overall responses show that the key is to give the performers a sense of control, create a balance between the live instruments and electronics and facilitate that process in a framework that is accessible to tour without too much technical assistance. In this context, it essentially becomes irrelevant if the composition is fixed or interactive, it is completely dependent on what the piece requires. Many composers are now exploring the possibility of creating multiple versions of the same work to facilitate this accessibility. ‘Wolf Rock’ for example by Leah Barclay exists as an accessible fixed version that is easy to tour and an interactive version for when the time and resources are available for a full performance.

The concept of establishing an ideal score from the perspective of the performer shows that the majority of artists participating in the survey simply require more information and a visual representation of the electronic part. Out of the performers surveyed 93% find graphic notation the most effective method, and 83% say they would be interested in exploring new interactive methods that use tools such as animation.

It is evident that to ‘liberate the stave’ a combination of graphic notation, western notation, sonograms, spectrograms, and tools such as GRM’s ‘Acousmographe’ is required to create ‘perceptual’ scores. There are many other examples of digital tools available on the market to create graphic representation of sound. Yet, many performers surveyed suggest a composer’s own graphic notation tells them so much about the piece.

An Australian guitarist says “Graphic notation speaks to me in a way western notation doesn’t, I feel more connected to the gesture and energy the composer is creating and I feel a sense of freedom working with the electronics in this context”. A number of performers share a similar perspective, and emphasize the fact that notation is essentially a fixed physical form of symbols that represent a musical action. The great revelation of the notational experiments of John Cage, Earle Brown, and Morton Feldman challenges the very meaning of these symbols. Their intentions were to point to the infinite possibilities of any interpretation of a notation and to engage the performer in the creative process of the piece. This form of graphic notation can still be created in digital software but the results of the survey suggest it does appear to be more effective in the composers own hand.

CONCLUSION

This paper has provided some preliminary results in the methods of notating live instruments with electroacoustic music, and more specifically environmental soundscapes. It is evident a full documentation of the work, in both visual and text formats is essential for sustainability and accessibility and a fusion of existing notational tools and innovative methods is essential in creating this framework.

The final outcome of this research is combining the results from the case studies, comparative analyses and survey and developing a unique graphic notation for environmental electroacoustic music. This model will place a strong focus on accessibility and sustainability, and facilitate interdisciplinary collaborations fueling the possibilities for environmental electroacoustic music to have implications across a diversity of fields.

REFERENCES

[1] Emmerson, S., & Smally, D. (2001). Electro-acoustic music. In The new Grove dictionary of music and musicians (Vol. 8, pp. 59-67). London: Macmillan.

[2] Emmerson, S. (1986). The relation of language to materials. In S. Emmerson (Ed.), The language of electroacoustic music (pp. 17-39). Basingstoke, UK: The Macmillan Press.

IMPROVISING WITH GRID MUSIC SYSTEMS

Roland Adeney *Andrew R. Brown*
Queensland University of Technology
Victoria Park Road, Kelvin Grove

ABSTRACT

Grid music systems provide discrete geometric methods for simplified music-making by providing spatialised input to construct patterned music on a 2D matrix layout. While they are conceptually simple, grid systems may be layered to enable complex and satisfying musical results. Grid music systems have been applied to a range of systems from small portable devices up to larger systems. In this paper we will discuss the use of grid music systems in general and present an overview of the *HarmonyGrid* system we have developed as a new interactive performance system. We discuss a range of issues related to the design and use of larger-scale grid-based interactive performance systems such as the *HarmonyGrid*.

1. INTRODUCING GRID MUSIC SYSTEMS

We define Grid Music Systems as musical systems that provide a visual grid or matrix layout on a screen or physical interface as a method for the temporal structure of musical content. Moving around on the grid, like a game board, or pressing buttons of a grid-like array, provides the means to operate on, or 'play' the grid. A system may be a hardware console, software application, or a combination of the two. A typical grid system has a checkerboard-style layout with an active marker or cursor that shows movement on it. At times physical grid interfaces comprise a device covered with a square array of 'buttons' that are activated by being pressed. Typically, grid systems allow the user to perform or program a sequence of steps around the grid that produce cyclic musical patterns. Sound output from electronic grid systems is usually provided by virtual synthesizers and heard through a sound system.

Because grid layouts are so ubiquitous, we need to also describe some grid-related musical interfaces that are outside our current definition. There is a 'music grid' on *Facebook*[1], where one can create a grid of pictures of favourite album covers, listen to samples, and discuss them with others. There are various other 'grids' which are collections or databases of music files, such as the *In-Grid* MP3 download site. In addition, a chord chart may be referred to as a harmony grid. Such systems are outside the scope of this article.

2. OVERVIEW OF THE *HARMONYGRID*

We have developed an interactive music system called the *HarmonyGrid*. While "MusicGrid" might have been a more suitable name, given that it allows control over several musical parameters, it was not used because it is already widely used to mean other things. The only remaining direct source of confusion may be Levitt's *HarmonyGrid* program for Macintosh [14]. Levitt's program was a mouse-driven music application using a grid, where the speed of mouse movement over the grid affected note duration and position on the grid selected pitches, and cell steps could be adjustable to cover a variety of interval ranges. However chords need be selected for each (different) chord on each pitch.

Our *HarmonyGrid* uses a simple 4x4 grid of squares which is projected vertically down onto a performance area, so that it forms an area 2 metres square. The user, a musical performer, walks on and around the grid. The user's location on the grid is detected by a webcam overhead. Software developed in the Pure Data environment with the GEM graphic library extension, drives both the grid graphics (which are somewhat animated), and the generated music. Changes to the music can be triggered by the performer's location. The grid operates in five modes, corresponding to the musical parameters of spatial position, volume, rhythm, timbre, and pitch (or harmony). Movements of the performer on the grid sets the current mode parameter to the particular cell activated. Musical output runs continually and is generated in real time and interaction is 'live', so that for example, in 'harmony' mode, arpeggios based on the currently triggered chord continue to sound until a new cell is reached. Musical output is rendered with virtual synthesizers, spatialised to a quadraphonic speaker field, covering the performance area. The performer's role is to both improvise over the grid's musical output with a portable musical instrument, whilst moving around the grid, and operating the controller (described below) when desired.

1 The site is available from one's own Facebook profile via the search box, so a reference web address is not meaningful.

A setup diagram and photo of the *HarmonyGrid* system are shown in figures 1 and 11 respectively.

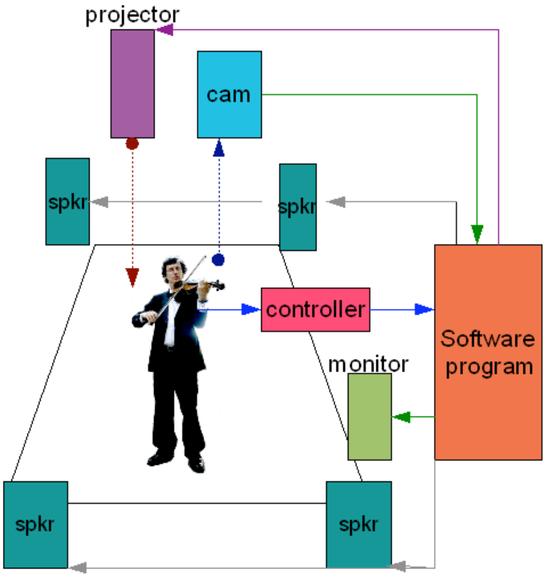


Figure 1. Information flow with the *HarmonyGrid*.

The *HarmonyGrid* is designed as a new musical instrument. It appears to be quite unique, in that it is a (relatively) large-scale grid on which the performer moves around to produce music. The *HarmonyGrid* bears some similarity to a number of previous music and movement systems, but these have often focused on movement as dance rather than movement as simple musical control. Dance games, such as *Dance Dance Revolution,* provide somewhat similar systems, in that they involve moving on a special surface (e.g. the dance-pad). However, the motivation for engaging is dancing and dance accuracy, rather than the creating and controlling of a musical score. Other similar systems where musical control is at least part of the result include interactive installations where sound samples and soundscapes are triggered by a user's location, as well as a few motion-tracking systems for installation or dance performance. Many motion-tracking systems, such as Garth Paine's *Gestation* [9], track full-body motion and allow one or many users to trigger quite complex musical systems. Tracking in Paine's systems is continuous across the space. Discrete dance-based motion tracking was used in *Dis-patch* [18], where quantized inputs – locations of limbs - was used to inform a complex music-making computer program. In these dance systems there is no obvious (visible) geometrical system constraining the input, to inform the output, and as such, is obscure to the observer.

3. REVIEWING GRID MUSIC SYSTEMS

In order to better understand the opportunities and limitations of various approaches to grid music systems we will review a range of different ones. This review is structured around a developing taxonomy of factors relevant to grid music systems. Typically, grid music systems create music from a combination of direct gestural input that triggers predefined music sequences or sound events. Playback often scans through the grid triggering selected cells, and scanning iterates, resulting in a looped output. In a more complex system like the *HarmonyGrid*, multiple synchronous scans or paths through grid space are possible. Few systems provide their own sound output, most requiring additional hardware such as a computer or MIDI instruments, to sound. The *HarmonyGrid* is no exception.

3.1 Topologies

We restrict our review to those systems using square grids as the layout for placement and movement of components, in two dimensions. Naturally, other topologies are possible, such as for the *ReacTogon* [5] that has a hexagonal array of buttons. Structures in many dimensions are also possible, but most systems use 2D grids, as these provide sufficient complexity to manage several parameters in real-time (e.g. time and pitch are common). The grids may be delineated by lines bordering the active spaces, such as a simple grid of squares, or be constructed of a square array of squares, circles, discs, buttons, or locations marked in some way. For example, the *Tenori-On* [12] model shown in figure 2, is a hardware grid providing a 16x16 array of circular buttons.

Figure 2. The Tenori-on
[Reproduced with permission from
http://www.global.yamaha.com/tenori-on/index.html]

The grids are often constrained to a small area, as with portable hardware grids, or expandable to user requirements, as with *Nodal* [16], which provides an expandable 'graph-paper' style grid to write to. Based on a rectangular 2D grid, *Nodal* [16] allows for a user-configurable network structure that provides for construction of paths of significant complexity and various topologies. To some extent the degree of flexibility of *Nodal* [16] moves it beyond the scope of our current definition of grid music systems, because it is difficult to assess whether the musical outcomes can be significantly attributed to the underlying grid.

3.2 Activation

Activation of components on grid 'squares' or cells is achieved in many ways. With hardware devices, buttons may be pushed directly as with the *Monome* [6] and *Tenori-On* [12], or objects on the grid may be recognized by video from beneath, as with the Bubble-Gum Sequencer (Hesse et al. 2007).

Figure 3. The bubblegum sequencer
[Reproduced with permission from http://www.backin.de/gumball/]

In software-based music grids, activation often occurs by clicking on a cell or an animated icon 'arriving' at a location. Many systems allow for paths to be set up on the grid, using graphical icons to control path directions as well as musical outputs. This is similar to a network of railway tracks with sets of points or switches, and one or several trains running on them. In these systems, current location is indicated by a flashing disc or the players' avatar/icon, as it moves along the paths. The Al-Jazari system [10] provides an example of automated icons moving around a software grid (see figure 4). Usually, several active icons are traversing the paths.

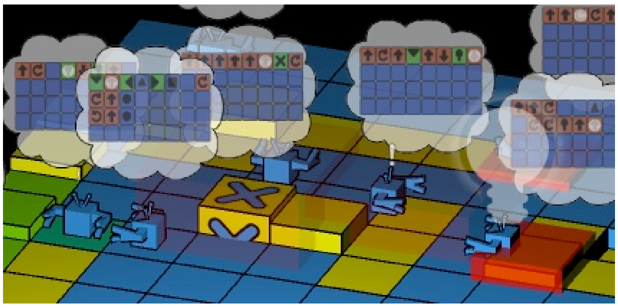


Figure 4. A screenshot from the Al-Jazari software
[Image from http://www.pawfal.org - Creative Commons License]

3.3 Input

Input for software systems, can be via a mouse, joystick, trackpad, MIDI controller or other device. Input devices typically function to trigger cells or set up and modify paths or sequences with a series of cell clicks. At times they act as tracking mechanisms where the pointer *is* the active component that is used to 'steer' through grid space. Typically, arrows and other keys on the keyboard may assist in direction controls when step-by-step navigation around the grid is desired. Input devices for hardware grid systems are typically provided by an array of buttons, as with the Monome [6] shown in figure 5, or a touchscreen.

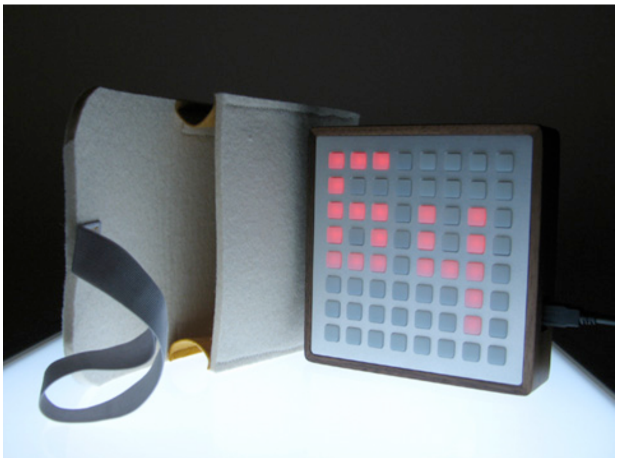


Figure 5. The Monome 64 grid interface device
[Reproduced with permission from http://monome.org/]

Input can also be via human movement or gesture that ranges from detection of digits or limbs to whole body location in space, or even extend to the detection of geographical position via GPS. The movement input can then be detected directly via hand operated input devices, or remotely by sensing or tracking from a distance. The *Buchla Lightning II* allows two wands to be played or conducted in a 2D grid space in front of the player. Detected by infra-red, the wands act as a MIDI controller to play synthesizers. *Lightning III* extends that control to 3D space.

To display the input, some systems favour a primary active icon, triggered by the user, for example with dance mats as shown in figure 6, whilst many systems have multiple active icons, only limited by musical intelligibility.

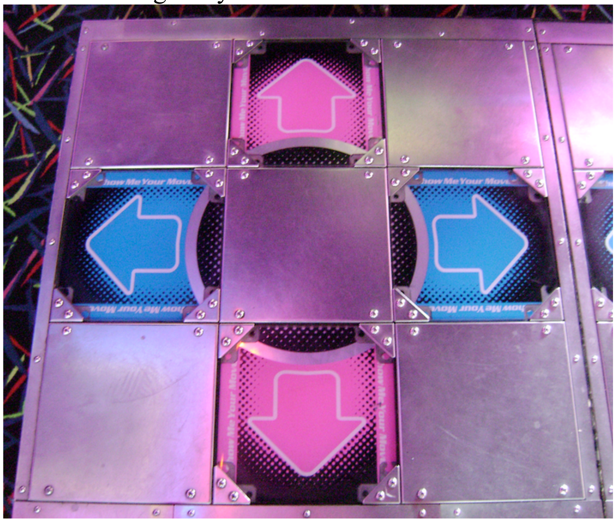


Figure 6. An example of a dance mat grid
[Images adapted from http://en.wikipedia.org/wiki/File:Dance_ Dance_Revolution under the GNU Free Documentation License]

Regardless of the input device, system design requires decisions about the mapping from inputs to various parameters, including graphics, music, spatial direction, among others. Mapping processes can range from the direct and simple to the comprehensive and detailed, and discussion of these is well covered elsewhere [8].

3.4 Outputs

Output from a grid music system comprises sound and a visual or graphical display. Some systems may provide haptic feedback. Graphical outputs usually include the grid itself along with the representation of paths, icons showing current location(s), and possibly other system state indicators. Graphical representations tend toward diagrammatic 2D designs, as with Nodal [16], rather than the 3D graphics more typical in computer games, as with Al Jazari [10].

Musical outputs typically are notes, short phrases, arpeggios or samples. These may combine and accumulate to make surprisingly complex resultant music. Tempi are generally adjustable for all systems. Sound palettes for hardware devices may be non-existent, where the device acts only as a controller, like the Monome [6], may be simple, particularly in self-contained systems such as the Tenori-on [12], or may connect to the vast array of software synthesis plug-ins. The audio spatialization capacities of systems will be directly dependent on the capabilities of the software synthesizers and audio hardware employed.

Systems may also send grid location or path position data over various protocols for coordination with auxiliary devices. Typically these protocols include MIDI and Open Sound Control (OSC), but may also include Wi Fi, BlueTooth, and so on.

3.5 Musical Geometries

There are many possible topological mappings of musical parameter space onto a grid layout. Given that a grid has a fixed number of cells - for practical purposes usually (much) less than 100 - choices need to be made about both the selection, quantization or clustering of parameter values and their arrangement on the grid. Topologies may be set up as a simple array of one parameter, such as pitch scales, or cells may relate to each other in two dimensions, as do 'neighbour relations' between the cells in 2D cellular automata. Parameter values may scan horizontally, vertically, diagonally, or in more unusual arrangements such as a spiral pattern. Either way they form a knowledge-space or matrix, relating similar items. Simple musical grids may use discrete pitches on different squares, though informationally this can work better in hexagonal arrays using a harmonic table (accordion style, e.g. *ReacTogon* [5]).

In certain circumstances, grid music systems can operate with arbitrary arrangements, like with a spatially triggered sampler with a matrix of touchpads such as the Akai MPC series, originating in the mid 1980s. Most systems provide for several mappings with different musical geometries for different layouts of squares and their functions and some are highly programmable, such as the *JazzMutant Lemur* [13].

Figure 7. A grid layout on the Lemur
[Reproduced with permission from http://www.jazzmutant.com/lemur_overview.php]

3.6 Paths

A path describes a series of steps or movements around the grid. Paths are created by moving around on the grid, and may be rendered visible by the system typically by leaving a trail of highlighted cells, as on the KAOSS Pads shown in figure 8. While all movement around the grid creates a path, they become more significant when paths can be defined and reused. Paths can be created by recording user movements, or by notating some path function as a series of rules or commands. Paths are sometimes said to be 'tracked', for example with *KAOSSPAD.*

Figure 8. Korg KAOSS Pad KP3
[Reproduced with permission from http://www.korg.com/Product.aspx?pd=269]

Our *HarmonyGrid* system, amongst others, provides for a real-time path 'recording', which is then represented as a moving sequence of icons.

Paths may be abstracted as gestures or contours and subject to scaling or other transformations, but this feature is found on few grid music systems. A path may represent a small data set such as a tone-row, that can be 'performed' in different ways, for example to generate output at multiple timescales or different path orders (forward and backwards). Paths can be 'played' as a cannon or segmented by varying multiple start and stop times.

Music grids use the grid to provide spatialised paths that may intersect at specific points, and have similarities with network topologies. The spatialised paths allow for 'clocked' musical and graphical events to occur, over time. The topologies become a network operating over time, a sort of graphical score, that may remain fixed, change or evolve slowly over time, or be very active (and interactive). (There is some breakdown of the network analogy, when we consider that time is generally taken to be 2D, moving forward.) *Nodal* [16] uses the grid dimensions to precisely measure timings of events. For example, a simple square path of 4 nodes (on the corners), may be set up, to 'play' as 4 crotchets in 4-4 time, and subsequent placement of nodes between the others will provide exact subdivision timings. Most other systems provide for rhythm – a sequence of timed events – where the active path runs over both active and inactive squares, thereby triggering or not triggering sounds in a pulsed sequence. Many operate in 'step-sequencer mode' where the x-axis is time; and the y-axis may be pitch, or a range of tracks with sound events; and a pulse runs across the grid activating it column by column. The resultant rhythm is often simply composed of crotchets, quavers and semi-quavers.

By using paths created and recorded on a grid in a variety of ways, a grid music system can produce complex results, and can be very flexible in its methods. Even given the small data set of paths, and the significant constraints of small grids, it is not hard to imagine how these methods of path creation and use can generate sophisticated outcomes.

4. RELATED SYSTEMS

Other systems, including block systems, touchpad and touchscreen systems, and 'light table' systems demonstrate other methods for music-making, and provide further context for grid music systems.

Block systems are somewhat parallel systems to grid systems, having many commonalities around the options and limitations of spatial relations between cells and blocks. Block music systems use custom-made cubes, somewhat like toy building blocks, which may be placed next to each other to make paths, as in dominoes, or they may be stacked vertically as well. Examples include *BlockJam* [17], shown in figure 9, and Siftables [15].

Figure 9. The BlockJam
[Reproduced with permission from http://www.sonycsl.co.jp/IL/projects/blockjam/contents.html#interacting]

In common with grid systems, block systems may effectively use square grid layouts, allow for construction of paths, and typically use several types of blocks to differentiate path directions or musical outputs. These systems may be quite sophisticated, as in *BlockJam* [17], *Percussa Audiocubes* [19] and the *Tangible Sequencer* [3], where blocks communicate with each by physical contact and communicate with a computer, and the software supports various musical or synthesis patterns created by arranging them. Paths are made by the arrangement of blocks and placement of special path-directing blocks. Sound output is via an external system rather than from the blocks themselves. The *MorphTable* [4] uses blocks somewhat differently. Being encoded with graphical designs, the blocks are identified by the camera and software and trigger musical events. They may be freely moved around on the table to affect music-making.

M, the “Intelligent Composing and Performing System” from *Cycling '74* [7] is a music-making program from the mid 1980s. M allows interactive composition in that “You can control your music by clicking and dragging the mouse on the computer screen, by "conducting" in a Conducting Grid" [7] which is a 2D grid area on the interface. M uses a one dimensional grid system for cycling around pitch, rhythm, dynamics etc independently of one another. The Patterns window, contain the Patterns, which are collections of pitches, rhythms and dynamics that may be transformed in any way. "A Voice in M is a "path" through the program that begins with a Pattern." [7]

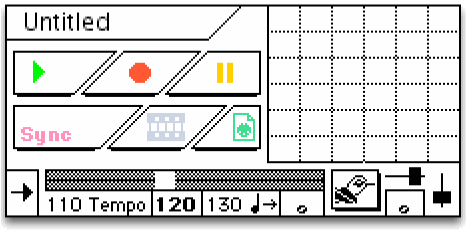


Figure 10. The conducting grid from the M software
[Reproduced with permission from http://www.cycling74.com/products/M]

Additionally there have been many other 2D surfaces developed to control music, or affect sound, for example AudioMulch's *Metasurface* uses a Voronoi/Delaunay mesh structure [2] in software, to navigate between parameters, sounds or parts of a composition. The geometry may be user-defined.

5. THE HARMONY GRID

The *HarmonyGrid* was designed so that a proficient musician could improvise with and control a generative music system. To that end, the *HarmonyGrid* is a real-time, open-ended format, creating music triggered by the performer's location. Musically, the system is not designed to be sufficient unto itself, but when acoustic improvisation is combined with controller input a satisfying ensemble can be created. Another aim was to provide a spatial component to musical performance that might become part of the theatrical presentation on stage. Spatiality is provided by both the added significance of performer location on stage, and the use of quadraphonic audio projection.

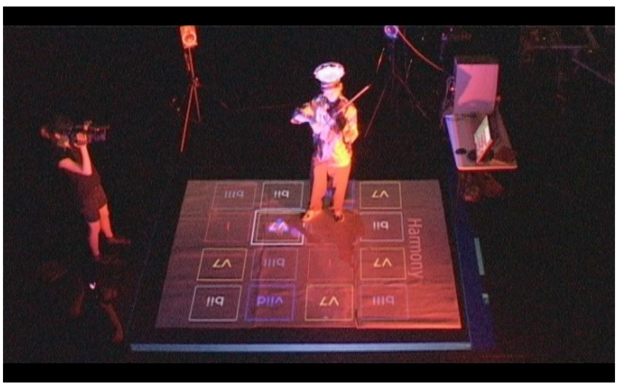


Figure 11. Performance with the *HarmonyGrid* system.

The system can be seen as an extended instrument[1] [1] that augments the instrumental performer, and is designed to provide an interface to electronic/computer music processes that go some way towards the kind of control available on an acoustic instrument.

The *HarmonyGrid* is an immersive grid music system where the performer moves about a large grid projected onto the stage whist playing an instrument, such as a violin or saxophone. The performer both improvises on their instrument while controlling a generative music system by walking around the projected grid. A camera positioned above the performer, near the projector, tracks the performer's position on the grid. The performer wears an electronic remote controller that can change the settings on the grid and how the position tracking is to be mapped or interpreted.

5.1 Movement on the Grid

The performer may move around on the grid in any way he or she pleases, but the tempo determines when squares are activated. For example, rapidly running around the grid with a slow tempo setting, is not effective. A particular feature is Dance mode, where single pitches are triggered in free time, by 'dancing' around the grid. After performing on the grid awhile, and switching to Dance mode, the main pulse is switched off, yet triggering remains somewhat quantized to maintain some rhythmic integrity. Dance mode now features as a creative interruption, and then the performer can switch back to the previous musical material. Naturally Dance mode adds more of a choreographic quality to the performance, and further consideration of more dance-like movement in the ordinary modes is a subject of future development for the system.

5.2 Music Spaces

The four grids of volume, rhythm, timbre, pitch (or harmony), provide very different musical and graphical environments. Moving on one of these grids alters that musical parameter for the particular square activated.

The program commences with the Volume grid, with its slowly shifting blue discs representative of volume levels, that, combined with the subtly changing arpeggiator, provides a rather static yet gentle ambience. The Rhythm grid, representing rhythmic patterns for each square, tends to be bright and lively, with its capacity for freely changing rhythmic patterns or playing a fixed cycle of rhythms. The Timbre grid is the most colourful, and adds musical timbral effects per square triggered to the musical patterns currently assembled.

1 "'Hyper-instruments' are large scale musical systems which enable the control of complex musical events by a performer." [1]

The Harmony grid presents an intriguing display of squares with Roman numeral notation for harmonies, which may facilitate an intellectual approach to selecting harmonic movement. Each grid's musical output can be enriched by adding paths from the other parameters; to good effect.

5.3 Paths and Voices

The performer may choose to record a path from 2-8 steps, which is then stored as the path belonging to the current parameter, e.g., the Volume path. Paths are played, when one or both of the Voices are active. Paths may be written from the current grid to another parameter at any time, e.g. the Volume path may be copied to the Rhythm grid. Or separate paths may be stored for each grid. Paths for each of the other parameters may play whilst on the current grid, so that whilst a harmony pattern plays on the Harmony grid, it may be modulated by the Volume path, the Timbre path (i.e. have differing volume and timbres for each step on the path), and play rhythmic patterns.

The 'active' voice is triggered live by the performer, and has an arpeggiator option. Once a path is recorded, Voice 1 (the upper voice) may be started, with its arpeggiation, and rhythmic options. Voice 2 (the lower voice) (one or both voices may play) runs over the same path, but can be offset by several squares. It has independent phrasing options with rhythms and pitches.

The *HarmonyGrid* currently carries options for harmony grids of various layouts, musical scale selection, choice of 32 instruments per voice, and percussion options. The layout of the harmonies in the harmony grid has been the subject of much research and testing. One layout comes from Euler, and features intervals of sixths running one direction and thirds the other way. Aside from providing a variety of options, the layout does seem to be much less critical than was originally assumed.

5.4 Musical results

As with most grid music systems, *HarmonyGrid* is most suited to loop-based styles, including minimalism, simple groove, and modern dance music. The loops can function as canons, either as chords or single notes, and can scan at very slow to very fast tempi. Very sparse ultra-slow music, to fast frenetic loops are possible. Because loops can function as bass lines, the upper voice can accompany as arpeggiated patterns with subtle changes, providing styles such as simple pop, or richer synth-based dance music to improvise with. Synthesizer voicings provide categories for dance, ethnic, tonal, and more unusual sounds.

5.5 The Physical Controller and visual feedback

The function of the physical controller is to fully enable the performer to move freely around the grid, and not have to return to control the computer via mouse. The controller is really the secondary control, with the primary control being the spatialised triggering of projected grid squares.

The physical controller is a 14x10x4cm custom-made box hung on the performer's chest. A collection of buttons switches and knobs input to an *arduino* board, which communicates via BlueTooth to the computer. A four layered control display shows on the secondary screen at the side of the performance area, providing visual feedback. The secondary screen also runs the Pd program main page, with its own onscreen mouse-driven controls, as a backup. The controller stops and starts the program, voices, recording paths, playing paths (Voices 1&2), arpeggiators, and rhythm generators; and enables selecting instruments for the voices, and displaying grids. Rotary knobs provide tempo and volume controls. Dance mode and Show-All-paths modes are selectable.

The controller sends rather erratic data that requires smoothing (the non-Bluetooth *arduinos* perform quite stably); with some workarounds in the programming. This early version of the switching layout on the controller, and current version of the control display; make the system somewhat difficult to operate, and more refined versions are anticipated for the future. For instance, familiarity with the switches and display layout is necessary, although all changes are shown onscreen. An important consideration is the necessity to stop playing one's instrument in order to use the controller.

5.6 Experiences of performing with the *HarmonyGrid*

A germinal design idea was for a simple hands-free arpeggiator and pattern accompaniment, and this function is fully realized. With the addition of the multi-parameter path capability, the system has grown into a reasonably sophisticated music generation tool, somewhat unique for its visual display directly mapped onto the performance area. Viewers are stimulated by the overall design concept of direct music triggering, over a graphical environment. Participants have enjoyed themselves, and one small girl danced on the grid for a full ten minutes. Viewers seem to enjoy a walk-through explanation.

Currently the system is still in early development, and quite some practice and experience with it are required to present a smooth performance. As mentioned earlier, the controller and its display screens are somewhat awkward to use, but the paradigm and the overall design takes some digestion and thoughtful consideration on the part of the performer; the instrument needs to be learnt. Spatial determination of volume and timbral paths are simple, yet for harmony, movement control is seemingly a new experience. The whole fascinating subject of spatial representation of music, both mentally, and physically, is a topic for future publication.

5.7 Comparison with other Grid Music Systems

HarmonyGrid is largely different from other grid systems, in the following areas and ways. *HarmonyGrid*:

- uses whole body movement on a projection
- is sonically immersive in a quadraphonic field
- is graphically more engaging than smaller devices with lights or buttons
- requires an improvising musician who understands and can play with written harmonic symbols
- is designed to be an accompanying musical system to live improvisation
- provides an electronic controller
- provides separate 'grids' for musical parameters
- provides layering up of paths from different musical parameters
- doesn't operate in step-sequencer mode
- doesn't provide path direction-switching icons
- features Dance mode, which is 'unclocked'

Similar to most other systems, it provides triggering by squares, and subsequent path activation and recording, leading to the layering up of paths or loops. *HarmonyGrid* doesn't provide output to send to other computer systems such as DAWs, or, for example, act as a controller for *Ableton Live*.

6. CONCLUSIONS

In this paper we have discussed the variety of approaches to grid music systems. We have surveyed a range of systems and discussed salient features of these systems as a class, towards developing a taxonomy of them. Against this contextual background we have described the *HarmonyGrid* system we have developed and provided reflections on its use.

In performance testing, the *HarmonyGrid* has shown to be an engaging and viable means of music-making in a larger-scale grid music system. *HarmonyGrid* demonstrates that spatial control by body location is a viable means to navigate the musical data space. The system is unique in more than several ways; visibly it is unique in that the user walks on a projection of the grid, operates a controller, and in that the system is designed to accompany an acoustic improvising musician. In addition, the system allows layering of paths from various musical parameters.

In future papers we intend to discuss other aspects of grid music systems including compositional considerations, the use of the spatial dimension in traversing musical data, and provide more detailed insights into performance considerations that the *HarmonyGrid* provides as an extended instrument.

While grids as a construct have been used as a musical organizing construct across many cultures and times we hope to have demonstrated that in the digital age they continue to provide opportunities for organizing and elaborating on music possibilities for the improving musician.

6. REFERENCES

[1] Anderson, T. “Using Hyper-instruments for the re-distribution of the performance control interface” Reprinted from the *Proceedings of the International Computer Music Conference*, Arhus, Denmark, 1994.

[2] Bencina, R. “The Meta-surface – Applying Natural Neighbour Interpolation to Two-to- Many Mapping.*” Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05*), Vancouver, BC, Canada, 2005.

[3] Bernstein, J. T. *Tangible Sequencer.* 2007. Retrieved 2 May, 2009, from http://www.tangiblesequencer.com/

[4] Brown, A. R., Wooller, R. and Thomas, K. *The Morph Table: A collaborative interface for musical interaction.* In A. Riddel and A. Thorogood (eds.) Trans: Boundaries / Permeability / Reification. Australasian Computer Music Conference. Canberra, Australia: ACMA. 2007.

[5] Burton, M. *ReacTogon.* 2008. YouTube www.youtube.com/watch?v=AklKy2NDpqs.

[6] Crabtree, B. and K. Cain. (2008). "Monome." 2008. Retrieved 2 May, 2009, from http://monome.org

[7] Cycling '74. *M 2.7: The Classic Intelligent Composing and Performing System.* 2007. Retrieved 4 May, 2009, from http://www.cycling74.com/products/M

[8] Doornbusch, P. “A brief survey of mapping in algorithmic composition” *International Computer Music Conference*. Gotëborg, ICMA**.** 2002, pp 205-210.

[9] Paine, Garth. *Interactivity, where to from here?* Retrieved 4 March, 2009, from http://www.activatedspace.com/Papers/Papers.html

[10] Griffith, D. *Al Jazari*. London, 2008, Pawfal.org.

[11] Hesse, H., A. McDiarmid, et al. "Bubblegum Sequencer. 2007. http://courses.ischool.berkeley.edu/i290-13/f07/node/382.html.

[12] Iwai, T. *Tenori-on*. Yamaha Music. 2008.

[13] Largillier, G., Joguet, P. and Oliver, J. *Lemur. Bordeaux: JazzMutant.* 2004.

[14] Levitt, D. D. and E. Joffe *Harmony Grid*. MIT, Cambridge, MA, 1986.

[15] Merrill, D., Kalanithi, J. and Maes, P. Siftables: “Towards Sensor Network User Interfaces.” *1st International Conference on Tangible and Embedded Interaction (TEI'07)*, Baton Rouge, Louisiana, 2007, pp.75-78.

[16] McIlwain, P., McCormack, J., Lane, A. and Dorin, A. “Composing With Nodal Networks” *Australasian Computer Music Conference*, Adelaide, Australia, 2006, pp.101-107.

[17] Newton-Dunn, H., Nakano, H. and Gibson, J. *Block Jam.* San Antonio, Texas, SIGGRAPH, 2002, pp.67.

[18] Parrot, C. and J. Mustard. “Dis-patch/Practice-led Researchers.” 2004. Retrieved 10 June, 2009, from http://practiceledresearch.ci.qut.edu.au/pages/index66b9.html?q=work/shaaron-boughen/dis-patch

[19] Schiett, B. "AudioCubes." 2009, Retrieved 2 May, 2009, from http://www.percussa.com/.

EXTRACTING AND RE-TARGETING EXPRESSIVE MUSICAL PERFORMING STYLE FROM AUDIO RECORDING

Simon Lui
Department of Computer Science and Engineering, Hong Kong Univeristy of Science and Technology, Hong Kong
virtuoso@cse.ust.hk

Andrew Horner
Department of Computer Science and Engineering, Hong Kong Univeristy of Science and Technology, Hong Kong
horner@cse.ust.hk

ABSTRACT

Expressive musical performing style involves more than what is simply represented on the score. Performers imprint their personal style on each performances based on their musical understanding. Expressive musical performing style makes the music come alive by shaping the music through continuous variation. It is observed that the musical style can be represented by appropriate numerical parameters, where most parameters are related to the dynamics. It is also observed that performers tends to perform music sections and motives of similar shape in similar ways, where music sections and motives can be identified by an automatic phrasing algorithm. An experiment is proposed for producing expressive music from raw quantized music files using machine-learning methods like Support Vector Machines. Experimental results show that it is possible to induce some of a performer's style by using the music parameters extracted from the audio recordings of their real performance.

1. INTRODUCTION

It has been a hot topic recently to develop computational methods for expressive music performance. Expressive musical performance is more than just a simple variation in tempo and dynamic. The performing artist is an indispensable part of the music, deriving information from their understanding and musical knowledge. Not every expressive performance feature can be represented in music notation – something composers are well aware of. Hence, to understand expressive performance, we must study the musical behaviour of performers. Typically, researchers have built formal models of expressive performance based on real musical performance. In order to build models with strong empirical foundations, inductive methods must be introduced, such that a large amount of real-world performance data is used as the basis for the model. To deal with the complexity of such a large amount of data, we make use of machine learning and data mining.

There are a large variety of musical descriptors that can be investigated. These descriptors range from low-level features, such as RMS envelope and spectral shape, to high-level descriptors such as terms like "delightful" and "sad" music. High-level terms can also be described by a combination of low-level audio descriptors. A common question of interest is, whether it is possible to represent expressive styles in terms of these descriptors in a digital format. Previous research has shown that music styles can be represented, to certain extent, as the deviation of three fundamental parameters: dynamics, tempo and articulation [1]. Previous research suggests that different musicians usually perform the same piece in a similar way in aspects like dynamics, due to music context, structure, common musical sense, and so on. However there are also slight differences between different musicians [2]. Each musician has a unique performing style, where some particular performing features will uniquely and frequently appear in different pieces played by the same performer [3].

Experimental results also show that by collecting several pieces performed by the same musician, it is possible to train a set of performance style parameters from the performance data, where the trained data can be used to distinguish the performance style of that particular performer from others [4]. Successful learning from even extremely limited training data can still be achieved by making use of ensemble learning. Once learned, the extracted "performance style" can be applied to a raw note list to make it expressive.

We propose an experiment where musical style is induced by multiple Support Vector Machines and applied to MIDI note lists in order to produce expressive musical performances. We first describe some musical facts observed from several expressive performance excerpts. We then discuss the implementation of our proposed experiment. Finally we conclude with the listening and statistical test results, as well as our future plans.

2. OBSERVATIONS

Our program has been developed on the basis of the following observations, which have been carefully tested, supported with strong reasons, and prolific examples. This is an essential step in our program development.

2.1. Global dynamic trend

Figure 1 shows a smoothed dynamic graph and a smoothed pitch graph of the Sonata No.1 in G minor BWV 1001, second movement (Fuga), by J.S. Bach, performed by Jascha Heifetz. The complete piece of music is smoothed by a sliding window of 8 bars (32 beats). The trends of the two graphs are very similar. This is not an isolated case. We sampled 6 different performers as well as different movements of the Bach Sonata and Partita, finding that they all return similar trends for the two graphs. Four of the results are shown in Figure 10. The global dynamic trend closely follows the global pitch trend. In most cases, the higher the pitch, the higher the dynamics. The only difference between different performers and different music is the trend ratio.

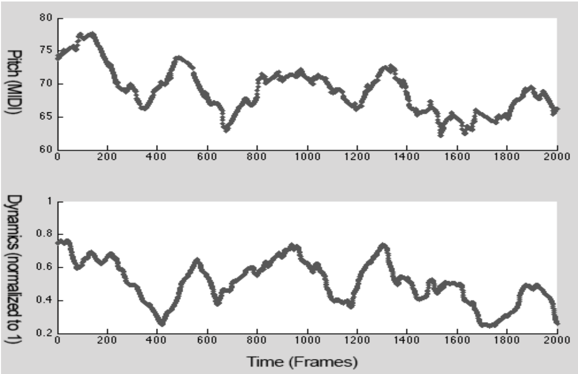
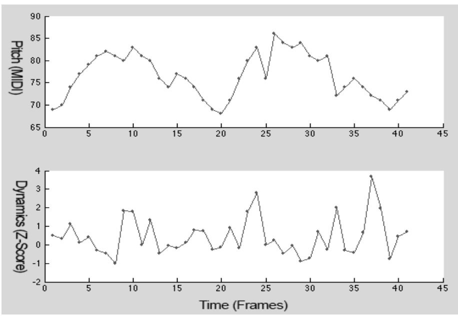


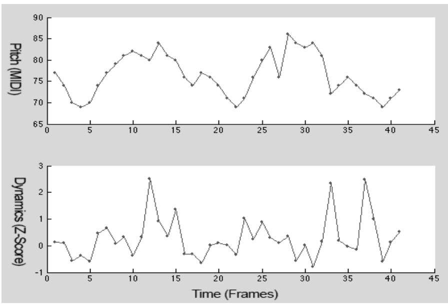
Figure 1. *A smoothed graph from the Sonata No.1 in G minor BWV 1001, second movement (Fuga) by J.S. Bach, performed by Jascha Heifetz. Top: The smoothed pitch graph. Bottom: The smoothed dynamic graph.*

2.2. Local dynamic change

We performed careful observations on every position for several movements of the Bach Sonata and Partita. We first magnified a small portion of two bars into a full screen on our computer. We compared local dynamic changes with corresponding local pitch changes, using several preprocessing methods including logarithmic, smoothing, standard score, deviation chart and so on. We eventually found that the standard score (Z-score) reflects the relationship between local dynamics and pitch changes well. Figure 2 shows two examples. The pitch trend in (a) is similar to (b), and the trend of Z-score of their relative dynamic levels looks very similar.



a)

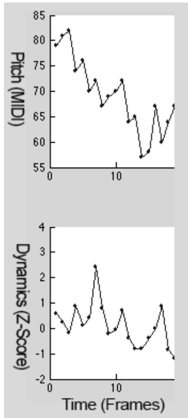


b)

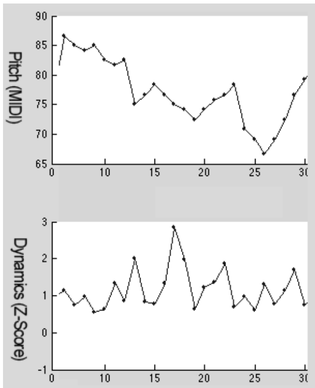
Figure 2. *Two excerpts from the Partita No.2 in D minor, BWV 1004 4th movement (Giga) performed by Itzhak Perlman.*

Next, we found that phrases with similar pitch patterns but different scales also had similar dynamic patterns in their own scales as shown in Figure 3.

To conclude, we believe it is possible to describe a performer's dynamic style by combining his/her global and local dynamic trends. As a violinist, this assumption well-matches my behaviour as a performer: imagine when we first look at a piece of music, we initially picture the whole piece from a global point of view, planning for the roles of different sections. However for each small motive we customize it to a personal performance style. We are very likely to perform with similar dynamic patterns for phrases with similar fingering patterns, hence it is reasonable for similar local pitch trends to have similar local dynamic trends.



a)



b)

Figure 3. *Two excerpts from the Partita No.2 in D minor, BWV 1004 4th movement (Giga), performed by Itzhak Perlman.*

2.3. Feature vectors

It is possible to use a few fundamental features to fully describe a performance style. Gerhard Widmer showed that dynamics, tempo, and articulation are adequate in representing the performance style of a piano solo. He successfully classified performance style of different pianists by using dynamic and tempo relationships.

From our point of view, dynamics and articulations are essential features for describing an expressive performance style. Moreover, for vocal, wind, brass, bowed string instruments and vocal, performing techniques like vibrato and glissando are also essential features for describing their performing style. We describe these features as pitch bend. Tempo is also essential for piano solos and perhaps for most other solo performances such as harp and guitar. However, much other type of music is performed in ensemble form. These are as important as solo performances, and probably there are more ensemble recordings than solo recordings in existence. If the ensemble music is classical music, the performers usually have to follow the tempo of the conductor or the quartet leader; for pop music, the lead instrument or singer usually has to follow a metronome or the tempo of the drummer, as most drummers are actually following a metronome beat through a headset. We measured the tempo of a number of pop pieces, symphonies and solo concertos, using the beat tracking program by Dixon [5]. Dixon's algorithm ranks the 1st in the Audio Beat Tracking task in Music Information Retrieval Evaluation eXchange (MIREX) 2006, which is very accurate and efficient. We found that the global tempo was steady most of the time. So, we believe global tempo is a good style indicator for solo performances, but not for music performed by more than one player. Although there is still a little local deviation in tempo, but the deviation is limited within a small range of the global tempo, where the performer cannot go ahead too much and he has to return to the original tempo in a bar or two. Since the global tempo is almost steady, we interpret this tempo deviation as lengthening and shortening of notes, which is the scope of articulation.

It is observed that the dynamic feature can describe the articulation feature. Articulation refers to the length of music note and it can be described by its note-end position: it is a note-end when the dynamic value drops below certain threshold. To conclude, we believe that dynamics, sometimes together with pitch bend, can fully represent music performance style. In this paper, we first focus on dynamics, since it applies to all musical instruments.

3. EXPERIMENTAL SETTING

3.1. Music data

Expressive performance recordings were extracted from an audio CD. Wave audio files in simple PCM format (44 kHz, 16 bit) were used. The input music data were assumed to be in MIDI format 1, following the GM (General MIDI) standard in order to standardize the velocity and channel parameters.

We chose the unaccompanied Sonata and Partita for the violin solo BWV1001-1006 by J.S. Bach, since it is a large set of accompanied works that can provide many clean samples for training. It is also one of the most famous violin masterpieces, hence it is easy to find many different versions by different performers. The reason for choosing Bach's music is because his music has dense and conscientious musical structures, for example, fugue and counterpoint. We believe that it is relatively easy to find re-occurring patterns in Bach's music, hence it should be a good starting point. For each piece, we prepared certain versions performed by different famous violinists including Itzhak Perlman, Jascha Heifetz, Midori Goto and Gil Shaham. The author, Simon Lui, also included performances for some excerpts as well, since we can record the same piece for an unlimited number of times, which can help us find the differences and similarity of a sample piece performed by the same performer. The MIDI data were downloaded from the Classical MIDI archive. All MIDI files were quantized and tidied for ready use. We used the Vienna Symphonic Library Strings Pro edition as an output sound sampler, and we used Logic Pro 8 as a sequencer to process the string instrument library.

3.2. Support Vector Machine

For the SVM machine, our experiments used LIBSVM Version 2.88 developed by the National Taiwan University, which was implemented by Chih-Chung Chang and Chih-Jen Lin [6].
Since SVMs require each data instance to be represented in numerical format, we used the GM (General MIDI) digital value to represent the vector value. Each support vector contained 64 frames, and each data represented the pitch value (1-127, in the GM standard) at a certain time.

In this support vector design, both rhythmic and pitch changes were included. 64-frame-level data was considered since most pieces rarely have notes durations shorter then a 128th notes. One support vector represents the features within 8 beats (i.e., 2 bars), so 8 / (1/128) = 64-frame-level data is considered. If the 256th note is present, then probably an 8th note instead of a 4th note could be tracked as a beat, so 128th notes could be identified by the program relatively as a 64th notes. Also, the inexpressive MIDI score files used were quantized and non-expressive, so there were no grace notes or acciaccatura. Hence the support vector of the 64-dimensions has fine enough resolution.

4. IMPLEMENTATION

4.1. Note identification

First, we identify musical notes from an audio file by pitch extraction. We used a modified pitch extraction algorithm suggested by Peeters [7] which is a very fast and accurate. Assume the fundamental frequency of a harmonic tone is f. The FFT of this note should peak at f and its multiples. On the other hand, the cepstrum of this note should peaks at f and its divisors, because cepstrum shows the repeat rate of the peaks and crests in the corresponding FFT graph, while the highest repeat rate of peaks in the FFT graph are probably the fundamental frequency and its divisors.

Hence to conclude, the cross product of a FFT and cepstrum should produce a graph that peaks at the fundamental frequency. Peeters suggested that the cross product of the autocorrelation of a Discrete Fourier Transform and cepstrum can reach the highest accuracy of 97%, while the cross product of a FFT and cepstrum has the second highest accuracy of 91.4%.

We further increase the accuracy of the pitch estimation by a low-pass filter of 20Hz and doing a natural logarithm in order to sharpen the peak of the FFT and cepstrum. Moreover, since we extract pitch data for every frame, and each note actually lasts for at least 8 frames, an error correction technique can be applied to fix almost all the errors: firstly, discard discontinuous frames with lengths less than a 32nd note; secondly, for gaps shorter than a 32nd note, are filled in with the next/previous pitch.

We tested the accuracy for an excerpt from the Partita No.2 in D minor, BWV 1004 4th movement (Giga) performed by

Itzhak Perlman. The final pitch estimation was improved to 96% with this simple error correction technique. The remaining 4% error was mostly due to the performer's slightly off-key performance. We carefully looked into some miscalculated cases, and all of them were actually off-key. In conclusion, there is almost no room for further improvement to the pitch estimation accuracy for monophonic audio.

Sometimes we still got a dirty cepstrum graph even though the music was monophonic. This was mainly the reverb from the previous note. This error was solved by sharpening the graph by a prior natural logarithm. After obtaining the fundamental frequency for each note, we converted them to the nearest pitches, and then to the General MIDI parameters.

4.2. Beat Tracking

The audio file data was then compared with the MIDI file. However, before doing the comparison, the audio file data has to be mapped to the corresponding notes in the MIDI file. To do this, the beat position of each note has to be calculated by beat tracking.

First, we calculate the IOI (inter onset interval) for all note-on times, and then perform clustering for the IOI values. IOIs of difference less than 25% were joined together. When all the IOIs were merged, clusters were merged with a 25% threshold. Finally, the cluster with the most number of members was taken as the beat of the piece.

Beat tracking was performed for both audio data and source MIDI. The beat tracking result was not 100% accurate, sometimes it over- or under-estimated the beat by a scale of two or one half. However it did not affect the accuracy of our machine, since we are only looking for reference points in the same piece of music, while the reference points between MIDI and audio data always match perfectly.

Since we only target the largest final cluster, it is a waste of time to merge clusters after we processed each IOI. Instead, we can merge the clusters once only after all the IOI are merged, and the result is exactly the same as our last implementation.

4.3. Note Mapping

After beat tracking, we locate a set of reference points in the MIDI and audio files, hence we can perform note mapping between the audio and MIDI data. We map every MIDI note to its corresponding audio note instead of the reverse. This is because the pitch data in the audio files, which are extracted by the pitch extraction algorithm, might not be 100% correct even though the error rate is small. On the other hand, the MIDI file is a source reference file and hence its pitch data is 100% correct. Hence we should do the mapping from the MIDI notes to audio notes.

We use a 96-frame window for mapping notes in a 64-frame vector, which is 150% of the target size. We search for the correct mapping pattern in a binary string format. The binary string with the largest sum of values is the best-matched sequence. The pseudocode of the note-mapping algorithm is as follows:

```
for trial = 1 to 2^windowSize-1,
    for bit = 1 to windowSize,
        bitString(windowSize-bit+1)
        = floor(mod(trial/(2^(bit-1)),2));
    end
        match midiString with bitString,
        producing matchedString composed of 1 and 0.
        count how many 1 are there in matchedSting,
        break if at least 90% match.
end
```

Finally we prepare data in the SVM vector format. We use the pitch as a vector feature, and the dynamic cluster code as the class. Since the note mapping results are mostly 100% full mapping, it is more efficient to try from an all '1' sequence in order to save computation time.

4.4. Segmentation

After note mapping is done, we segment the music into phrases. At an early stage of development, no segmentation was done, and we considered each note by a sliding window with a fixed size of 8 beats. However the learned performing style was not accurate. This implementation did not match how a performer thinks: a performer makes judgment about their own performing styles by motif, not by phrases of fixed length.

Hence in the second stage, we processed the input phrase-by-phrase, where the phrase segmentation was done manually. The whole piece was first divided into a few (four to eight) musical forms, and then within each form similar musical phrases were identified and motives of a half-bar to 4-bars were identified.

However, in order to build up an accurate SVM machine, we needed a large amount of segmented music samples. It was too slow to do the segmentation manually and we needed an automatic solution. Hence a modified version of the Phrase Stealing Algorithm by Lui [8] was used to perform auto segmentation. The originally Phrase Stealing Algorithm identifies music phrases by tying individual music notes together according to a voice leading table. In this experiment, we tie music notes together according to harmony progression. First, we segmented the music by long notes and rests, resulting in sets of music chunks. Music chunks shorter than 64 frames could be readily used as phrases. We did further segmentation for those music chunks which were longer than 64 frames. Within each chunk, for each note, a list of expected chord was calculated, resulting in an 2D "expected chord matrix". The elements in the expected chord matrix were tied according to a self-made "chord progression table". Finally all the tied phrases were viewed as motives.

Motives of different lengths were normalized to a fixed length of 64-frames. This process is based on the observation quoted in section 5.3.2 that similar pitch trends of different scales also have similar dynamic trends. As a result, we get a set of note vectors, all of length 64-frames.

The FFT process is actually the most serious bottleneck of the whole program. However, the FFT process for different portions of music is independent and can actually be done in parallel. The FFT process can be speeded up by dividing the piece of music into several chunks, and performing FFT calculations simultaneously in different threads. Originally, we used Matlab 2006a which is a single thread application. However, we can still perform multitasking with limited power by making use of Basic Linear Algebra Subroutines. We have to do these environment variable settings outside Matlab:

```
BLAS_VERSION        mkl.dll
OMP_NUM_THREADS     (number of threads)
```

Finally, we switched to Matlab 2008a which supports multi-threading. Setting can easily be done within the Matlab code, and the performance was greatly improved.

4.5. Absolute dynamic data

To calculate absolute dynamic levels, the root mean square (RMS) value of the signal amplitude was taken and only the RMS peak of each frame was used. The dynamic value of the whole piece of music was generalized to the MIDI scale of a range of 0-127, where the quietest note in the whole piece is indicated as volume 0 and the loudest note is indicated as volume 127. We choose a relative dynamic measure rather than an absolute measure because of human perception. Most people cannot tell if a certain single tone is loud or quiet, but everyone finds a 60db voice louder than a 30db voice. Hence we do not describe the dynamics in absolute values such as p, mp, mf and f.

In order to represent the articulation information in the dynamic vector, we set the dynamic threshold of defining music note-ends. We first measure the ratio of silent period within the whole MIDI score. Then, we plot an accumulative histogram of the dynamic value of the whole audio recording. The dynamic value at the index of the silent-period-ratio is the dynamic threshold of defining note-end. For audio frame with dynamic value below this threshold, the dynamic value is set to be 0.

4.6. Global dynamic ratio

To calculate the global dynamic ratio, the whole list of dynamics and pitch was first smoothed by a window of 8 bars, which is around the size of two to four motives. The global trend ratio is in a linear form as follows:

$$\mathrm{GlobalTrendRatio} = \frac{1}{\mathrm{numOfSample}} \sum \ln(\mathrm{dynamic}) / \ln(\mathrm{pitch}) \qquad (1)$$

This global trend ratio can represent the global dynamic of each musical section.

4.7. Local dynamic vector

To calculate the local dynamic vector, the RMS value of the dynamic data was not used directly, but we further reduced the global factor by using a standard score (Z-Score). The Z-Score calculates the local change of a note compared with the local mean, regardless of the standard deviation of the population. The Z-score can be calculated as follows:

$$Z = \frac{x - \mu}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}} \qquad (2)$$

Each dynamic vector is smoothed by a window of one beat. We can borrow notes from the next / previous window when smoothing the beginning and end of each vector. For the beginning and end of the piece, we simply decrease the length of the window.

The smoothing process helps in representing the general trend of the relative dynamic change within a vector. It actually sounds more natural to express the general intention of the performer rather than reproducing each digit from the data source. To clarify this, we played a short excerpt from the same movement of the Bach Partita three times, all with the same dynamic intention of a crescendo and then diminuendo. The three dynamic graphs look a bit different but the smoothed versions looks almost the same. Here we conclude that with the same expressive intention, the resulting original data can be different, while the smoothed data will look very similar. Further, we seldom find more than 2 global peaks in each dynamic vector. Perhaps it is possible to generalize the vector by a formula. We will try this at the next stage of development.

Next, all elements in each pitch and dynamic vector are subtracted by the value of the first element in the corresponding vector. Hence, the vector represents the relative pitch and dynamic change compared with the first note.

After smoothing and scaling, we perform clustering on the set of dynamic vectors. Each vector joins a cluster if the difference between the cluster value and the squared sum of its feature components is the minimum among all clusters and is less than 300. We choose 300 as a threshold based on the observation that the z-scores mostly range from -4 to +4. The difference between the absolute crescendo and absolute diminuendo vector is 1430. In this case, the two vectors should never be in the same cluster:

$$2\sum_{i=1}^{32}\left(\frac{4+4}{64/2}i\right)^2 = 1430 \qquad (3)$$

The difference between two almost parallel vectors which have linear average values of +1 and -1 is around 2 x 2 x 64 = 256. In this case, the two vectors should be in the same cluster, while this should be the upper bound threshold for a vector to join a certain cluster. Since we aim at re-targeting individual and expressive performing styles, it is fine to over-fit the clustering process since the style can still be preserved in different clusters. However, we have to avoid loose clustering which alter the shape of the performing styles too much. Hence a threshold of 300 will be 4 times away from 1430 and just fit the squared sum difference of 256. Clustering values can be fine tuned in the future, which only alters the number of clusters produced and the precision of the style data.

Instead of the brute force cluster merging approach, we speeded up the merging process by only considering clusters that have just been changed. However, we cannot leave all the cluster merging work to the end as we did in beat tracking (see session 6.2.3), since we need to assign the cluster number to each dynamic vector after it is clustered. Hence we need to update the cluster list for every vector calculation run.

4.8. SVM training and prediction

We use SVM for training because we do not want to over-fit the data. In a real world example, there are always inseparable feature vectors. An over-computed separation formula will waste a lot of computation time. The trade-off between using a less complicated method is perhaps a few incorrectly classified points. Actually it is more practical and accurate to give up a few scattered feature vectors.

We estimate the SVM Kernel parameters by using a systematical optimal model parameter search. The best way is to start with n-fold cross validation. We first divide the training set into n subsets of equal size, sequentially one subset is tested using the classifier trained on the remaining n-1 subsets. The most suitable parameters should give the best results in cross validation tests.

We chose the radial basis function (RBF) kernel [9] among the four existing kernels (linear, polynomial, RBF, and sigmoid). The linear kernel is too rough and should not be used. The polynomial kernel has too many hyper-parameters, which increases the complexity and hence the running time of the parameter search. The sigmoid kernel behaves like the RBF kernel in many cases, but with no real advantages over it. For some parameters the sigmoid kernel is not valid. To avoid unnecessary failure of the program, we choose RBF kernel which is proved to be the best kernel of all for our purposes.

Originally, we developed a graphical version of LIBSVM, where the user only needed very little input, and the remaining textboxes were automatically filled with suggested parameters. However, we finally achieved an automatic estimation of all parameters so we revised the skeleton and embed the code into the Matlab structure as a one-click design.

4.9. Re-targeting music

Lastly, the selected expressive performance style is re-targeted to a raw note list of MIDI. The global dynamic data is first calculated with the global trend ratio of the selected performer. Then the local dynamic data is predicted with the SVM machine. The local dynamic data is then merged with the global dynamic data in order to produce the actual dynamic level of the performance. The resulting dynamic data is then converted to the MIDI GM1 data format which produces an expressive MIDI performance file. The expressive MIDI file is finally rendered with a software sampler to produce an expressive audio performance file. An overview of the whole process is shown in Appendix I.

5. TESTS

5.1. Listening test setting

First of all, we decided not to compare the machine's output with the original wave file. Comparison is fair if and only if it is performed under the same environment. It would be unfair if the files to be compared were produced from different sound sources. Hence by comparing audio file output from the same source, the following tests reflect the performance of our machine rather than the quality of the original audio or the sound sampler.

Twelve listeners were invited to do the test. Eight of them were musically trained while seven of them could play the violin and knew the Bach Partita very well. Four of them were not musically trained but all of them enjoyed listening to music. The listeners were required to sit in a quiet room using headphones. After the following sequences were played: o*riginal file, predicted output file, original file,* they had to rate on the performance of *predicted output file:* 7 being the best, and 1 being the worst, as shown in Table 1.

All clips were music excerpts of 6 seconds, each listener performed 4 sessions for each of the three tests. The tests were very short since the tests were intensive in nature where fatigues highly affect the accuracy of the test.

5.1.1. Test 1: basic accuracy

First, the expressive performance data of an audio file was extracted. The expressive data was directly applied to its MIDI source file, producing an expressive *MIDI file A*. Then the expressive data was used to build an SVM machine. Using the SVM machine, performance parameters are predicted using the MIDI source file as input, producing an expressive *MIDI file B*. Both *MIDI files A and B* were passed through the sound module to produces two audio files, then the listeners judged their performances.

5.1.2. Test 2: influence of extra feature vectors

Similar to test 1, the expressive performances from four audio files were extracted. The SVM machine was built with these four different pieces of music, producing four pairs of audio files. Listeners judged the performance between different pairs of output files.

5.1.3. Test 3: ability of predicting unseen data

Similar to test 2, the expressive performances from four audio files were extracted. However, the SVM machine was built with three of them only, in order to predict the performance of the remaining unseen piece of music. Listener judged the performance between different pairs of audio output files.

5.2. Listening test result

The test results are shown in Table 1. Listeners found all the music very natural in tests 1a, 2a and 3a, because the style data were extracted from real performances, and the design of describing dynamics as a combination of global and local portion was successful.

For classification between styles, both tests 1b and 2b obtained good results, while test 3b was just fine, because test 1b and 2b include the original wave file in the training set. The result of 3b could be improved after the implementation of articulation learning.

Table 1. The *listening test result.*

Test	1a	1b	2a	2b	3a	3b
Rate	6.52	6.21	6.67	6.25	6.33	5.71

5.3. Statistical test setting

The tests in section 5.1 were performed again but evaluated in a statistical way. It is actually difficult to perform a statistical test. This is that our machine re-target a performance trend rather than directly copying each dynamic level, by smoothing every vector before training. Therefore the original performance will not be reproduced even though the training data and predicted data are the same, and hence it is difficult to do a statistical test to fully evaluate the performance of the machine. However, it is still possible to evaluate the machine's performance in a statistical way by comparing the cluster code between the output files. Moreover, we can calculate the run time to evaluate the efficiency.

5.4. Statistical test result

The result of the statistical test is shown in Table 2.

Table 2. The s*tatistical test result.*

	Test 1	Test 2	Test 3
Accuracy	93.2%	91.7%	68.2%
Run Time	42.43s	67.21s	58.05s

First of all, the run time of the machine is as expected and very efficient. Test 2 has the longest run time because it has a larger training set. We will keep working to shorten the run time.

Both tests 1 and 2 show excellent results, proving that the SVM feature vector design is appropriate, where test 2 shows that excess training data will not affect the accuracy, and nearly no input vector will run into an unseen situation.

The result of test 3 is also as expected and does not mean the experiment was unsuccessful. There are a lot of limitations in performing statistical tests for test 3. The most important reason is that many similar dynamic trends will not merge into the same cluster. As described in section 6.26, we have tight criteria for cluster merging. A loose threshold will result in very few clusters, where all will being plain, uninteresting and unexpressive. When we carefully looked into some incorrectly classified vectors, we found that they were actually very similar but belonged to different clusters. For example, some of them only had different ending dynamics, while some had a sudden raise or drop of dynamics in the middle. However, the rest of the clusters were almost the same. Actually the performance of this machine is difficult to describe in a mathematical comparison. However, it does reflect the machine's ability to certain content.

6. FUTURE WORK

Here is a list of the proposed future work.

6.1. Feature vector with more parameters

Research on articulation parameters is the first priority to be settled among all the future work. It will be used to build up a 2-D performance trend with the z-score dynamic trend. One of the main problems needed to be solved is how to measure note ends. More research on human perception will be done.

6.2. Global trend ratio

The global trend ratio is currently just a single number. It should be possible to describe it as a formula, for example, in the form of a regression or polyphonic equation.

6.3. Work on Polyphonic music

The current machine is actually optimized for a polyphonic approach already. The only remaining problem needed to be solved is melody extraction. This topic is highly problematic and needs another full paper to discuss it. However, we will try to find a robust and efficient approach which extracts pitch only. We will discard the MFCC and tone features. This should be adequate and feasible for this machine.

7. CONCLUSION

Building computers that can learn musical performance style has been a hot topic in the field of artificial intelligence. In the past few years, research in AI and music has been creating systems that mimic human perception in order to recognize musical structures like a trained musician. Previous experiments show that it is possible to classify music into genre by learning; hence there exists some common style in music of the same genre. The next question is whether it is possible to extract music performance style parameters and reproduce expressive musical performance through a black box.

For completely automatic conversion of expressive music, while there has been some success in specialized problems such as beat tracking, most truly complex musical capabilities are still well outside of the range of computers, for example, identifying form and motif structure. From a practical point of view, the current technology is not advanced enough for the computer to understand music as a professional musician does, but it is intelligent enough to help and support musical applications. The automatic production of expressive music at present still requires human intervention in some form.

In the future, we will continue to work on increasing the accuracy and enhancing the run time of the program. For accuracy, more research on human perception and more observations of articulation parameters from real recordings will be done. For the run time, the workflow will be further optimized and we will find a more simplified approach which does not affect the accuracy. Our experiment already shows that it is possible to induce some of a performer's style. This is a stepping-stone for the next stage of our research.

8. ACKNOWLEDGEMENT

This work was supported by RGC grant 613508.

9. REFERENCES

[1] P. Zanon, G. Widmer. Recognition of Famous Pianists Using Machine Learning Algorithms. XIV Colloquium on Musical Informatics (XIV CIM), Firenze, Italy, 2003.

[2] G. Widmer, S. Dixon, W. Goebl, E. Pampalk, A. Tobudic. In search of the Horowitz Factor. AI Magazine, 2003

[3] S. Dixon, W. Goebl, G. Widmer. The Performance Worm: Real Time Visualisation of Expression based on Langner's Tempo-Loudness Animation. International Computer Music Conference, Göteborg, Sweden, pp 361-364, 2002.

[4] G. Widmer. Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report. AI Communications 14(3), 149-162, 2001.

[5] S. Dixon. Automatic Extraction of Tempo and Beat from Expressive Performances. Journal of New Music Research, 30 (1), pp 39-58, 2001.

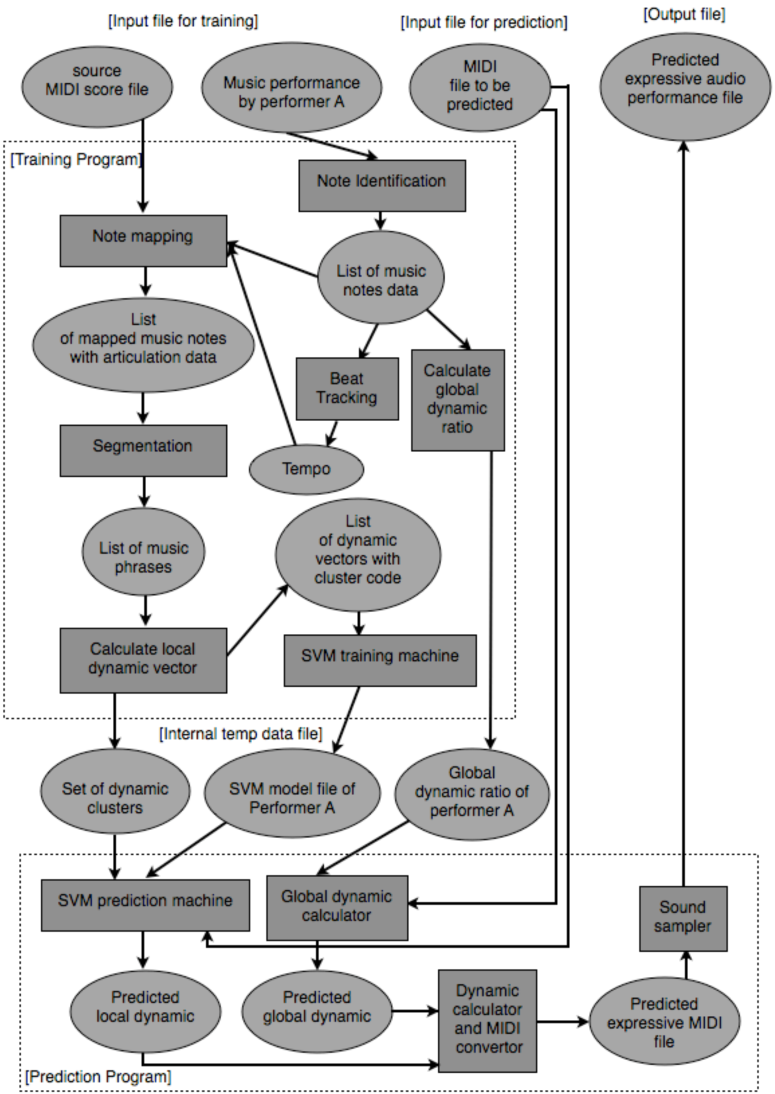
[6] C.-W. Hsu, C.-C. Chang, C.-J. Lin. LIBSVM: a library for support vector machines. National Taiwan University. 2004.

[7] G. Peeters. Music Pitch Representation by Periodicity Measures Based on Combined Temporal and Spectral Representations. IEEE International Conference on Acoustics, Speech and Signal Processing, 2006.

[8] S. Lui, A. Horner, L. Ayers. An Intelligent SP-MIDI Polyphonic Reduction Algorithm. IEEE Transactions on Multimedia, Volume 13, Issue 2, pp.52-59, 2006.

[9] C.-W. Hsu, C.-C. Chang, C.-J. Lin. A practical guide to support vector classification. National Taiwan University, 2004.

10. APPENDIX I: AN OVERVIEW OF OUR IMPLEMENTATION



TIEM SURVEY REPORT: DEVELOPING A TAXONOMY OF REALTIME INTERFACES FOR ELECTRONIC MUSIC PERFORMANCE

Dr. Garth Paine
School of Communication Arts
University of Western Sydney
Australia
ga.paine@uws.edu.au

Dr. Jon Drummond
MARCS Auditory Laboratories
University of Western Sydney
Australia
j.drummond@uws.edu.au

ABSTRACT

The TIEM (**T**axonomy of Realtime **I**nterfaces for **E**lectronic **M**usic Performance) online survey, examines the practice and application of new interfaces for real-time electronic music performance.[1] This project seeks to develop a theoretical base for new interfaces for electronic music performance. In this paper we discuss approaches to creating taxonomies of musical instruments with specific focus on attempts to classify Digital Musical Instruments. An overview of the TIEM online survey is provided and initial results both quantitative and textural are discussed.

1. INTRODUCTION

Despite the continuing and strong interest in the design and creation of new Digital Musical Instruments (DMI) [11] there is little consensus in current approaches to developing a coherent taxonomy of DMI. The now familiar and accepted taxonomy of acoustic instruments is based on the initial vibrating element of an instrument that produces its sound. Developed by Mahillon [9] and later expanded by Hornbostel and Sachs [6] the taxonomy consists of four top-level classifications—Aerophones, Chordophones, Idiophones and Membranophones. Each of these top-level classifications is in turn broken into numerous sub-categories creating over 300 basic categories in all. Sachs expanded the classification system in 1940 to include a fifth top-level group, *electrophones* for instruments involving electricity. In Sachs' classification system the electrophones were separated into three sub-categories—

1. instruments with an electronic action
2. electro mechanical, acoustic sounds transformed into electric through amplification;
3. radioelectric, instruments which are based on oscillating circuits.

With its basis on the initial sound making element, this classification into *electrophones* fails to capture the richness, diversity and trends of current digital musical instrument design. Unlike acoustic instruments, DMI's are not acoustically coupled to a direct link between interface and sound generating process. In designing a new DMI potentially any gesture can be mapped to any synthesis parameter. By placing the focus on the initial sound making device, the differences, similarities and relationships between new DMI's such as the eShofar [5], tooka [4] and T-stick [10] are lost.

More recent approaches to developing taxonomies of DMI have focused on the sensor types used, the nature of the interface, the way gestures are captured and the mappings between interface and sound generating functions [11]. Other researchers have proposed multi-dimensional spaces to represent DMI [14]. Pringer [13] compared DMI with respect to expressivity, immersion and feedback. While Birnbaum et al. [1] have proposed seven dimensions to represent the interactive potentials of DMI—

1. Role of Sound
2. Required Expertise
3. Music Control
4. Degrees of Freedom
5. Feedback Modalities
6. Inter-actors
7. Distribution in Space

2. REVISING DEFINITIONS

The TIEM project, although still in its early stages, reveals a wide range of innovative approaches to electronic music performance. Whether seen as an instrument or interface (a more detailed discussion about proposed definitions follows), it is clear that their principle focus is live music making.

Underlying all of the instruments currently listed on the TIEM web site[2] is a foundation concept of 'Instrument'. It is useful to unpack that concept to illuminate the influence it has on design and development.

Daniel J. Levitin [8] discuses musical schemas in his book *This is Your Brain on Music*. The relevance

[1] This project is part of an ARC Linkage project based at MARCS Auditory Laboratories the University of Western Sydney in partnership with Electronic Music Foundation, Infusion Systems and The Input Devices and Music Interaction Laboratory at McGill University.

[2] http://vipre.uws.edu.au/tiem (viewed 19/6/09)

to this research is his discussion of perceptual expectations and how these inform musical expectations and establish constraints and limitations in musical practices. They also form the basis for idiomatic writing for any instrument.

Levitin points out how trained and untrained people can sing 'happy birthday' regardless of the starting pitch. We hold a schema that is isomorphic, it can be applied to any starting point, is widely shared and always retains its integrity. Furthermore, it is context sensitive.

Organologies [11] [7] present a method of categorising musical instruments, but they do not explicitly detail an underlying schema, a generic concept of musical instrument. A musical instrument schema clearly exists, however an examination of organologies fails to illuminate such a schema. An examination of their application through musical performance is very helpful, as it essentially forms a design brief. This project seeks to develop a unified theory of practice for the application of new interfaces for real-time electronic music performance. The resulting taxonomy will be used to develop a design template that can be applied broadly in the development of new interface for electronic music performance.

3. METHOD

The online TIEM Questionnaire[3] consisted of 72 questions examining the practice and application of new interfaces for real-time electronic music performance. The questions consisted of a mix of textural and numeric, qualitative and quantitative, arranged into six sections—

1. General Description
2. Design Objectives
3. Physical Design
4. Parameter Space
5. Performance Practice
6. Classification

Participants were not required to answer all questions and were able to revisit the questionnaire to complete their submission. The questionnaire was launched in June 2008 and as of December 2008 we have had over 800 unique survey views with 70 completed responses.

4. OUTCOMES

An online database of the interfaces/instruments submitted to the survey (if they elected to be listed publicly) is available at the TIEM website. Table 1 presents an overview of the responses given to the quantitative questions. From the responses there is a clear preference evident for creating polyphonic (88.33%), multitimbral (86.67%) process based (60.71%) interfaces/instruments. There is also a strong preference evident for interfaces that are touched (83.33%) i.e. the performer has a physical connection with the interface/instrument. However, despite this preference for tactile interfaces, only 36.84% reported providing haptic (tactile/kinaesthetic) feedback to the performer.

Would you describe your interface/instrument as –	**Count**	**Percent %**
Polyphonic	53	88.33%
Monophonic	7	11.67%
Multitimbral	52	86.67%
Monotimbral	8	13.33%
Do you need to touch it?		
Yes	50	83.33%
No	10	16.67%
Does the interface/instrument provide haptic (tactile/kinaesthetic) feedback to the performer?		
Yes	21	36.84%
No	36	63.16%
Would you describe your interface/instrument as –		
Process Based	34	60.71%
Event Based	22	39.29%

Table 1. Summary of responses to some of the quantitative questions in the survey.

It is clear from the responses, that performers have a strong need for a physical connection with their instrument. A crucial step in the development of new musical interfaces therefore is the design of the relationship between the performer's physical gestures and the parameters that control the generation of the instrument's sound [15] [1]. This process is known in the computer science and engineering worlds as control mapping [15] [2], however the musician perceives it as a more homogenous engagement, where agency is decisive.

We asked how they thought of their system (instrument, interface, composition or other.) There was a slight preference for describing the systems as instruments (67.16%) versus interface (55.22%) with some (16.42%) also thinking about their systems in terms of a composition (Figure 1). Responses under the other category included—All the above; Composition tool; Experimental playpen; Holistic approach to sound; Installation; Interactive environment; Interface and composition; Performance device; Performance environment; Rhythm generation system; RTDJ interface; Semi-automatic improviser.

It should be noted that this notion of interface/instrument considered also in terms of a composition, while familiar to those working in the area is of course radically different from the concept of a traditional acoustic instrument. The survey participants' distinctions between interface and instrument were further revealed in some of their

[3] http://tiem.emf.org/survey (viewed 29/01/09)

textual responses. A selection of these is presented in Table 2.

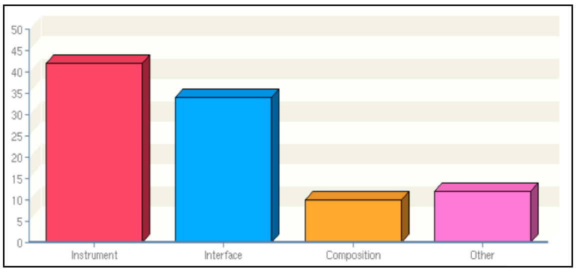


Figure 1. Question 10: How would you describe the system - as an interface, instrument, composition or other?

"I have never been able to draw a distinction between these two."
"I think an instrument is anything which is performable, through which a human can control parameters to afford expressive results."
"I consider it an instrument in the sense that I am dedicating my life to improving it, much like a violin builder dedicates his/her life to making the best violin they can."
"The more memory ("state") and autonomy a device has, the more likely I would be to call it an interface as opposed to an instrument."
"In my opinion, it's no clear distinction between an interface and an instrument."
"An interface is a controller that cannot create sound without some extra interface/tone generator/software. My interface becomes an instrument when I integrate it with the software I specifically developed for it to create the actual sound."
"A "musical instrument" includes mapping, synthesis, and sound production in the system. An interface can be part of a "musical instrument"."
"I think the distinction is blurred, but by instrument we can talk about the physical object, by interface we focus on the actions that are needed for making music with the instrument"
"the aesthetics of the actual interface controller makes an audience believe its an instrument."
"The interface is the mediator between the performer and the sound generator"
"all instruments have interfaces. But many interfaces are not instruments. ... I consider a musical instrument a tool that allows you to express yourself musically through interaction with it."

Table 2. Selected responses to Question 46: In your opinion what differentiates between an interface and an instrument? Is there an objectively definable distinction?

The survey also asked how controllable participants thought their interfaces/instruments were. A selection of typical responses is presented in Table 3. They reveal differing approaches to the concept of control—ranging from a desire for complete control with recognisable tight links between gesture and sonic response to systems that are unstable and difficult to control, yet are able to inspire and surprise with their sonic outcomes.

"As a composition system, gestures are not repeatable. In fact, the idea is that the same gesture will create different results each time."
"Different users will not create different music, as it is a reflection of my musical aesthetic."
"Doesn't have the instantaneous satisfaction of an acoustic instrument, it takes quite a lot of work to get what you want. I am playing, not merely controlling."
"This is kind of the whole idea: that gestures create recognisable responses"
"There is an important amount of surprise as you cannot always know exactly what the instrument will do in the next moment. This, however, enables me to react on it. It's a real challenge which can in result into wonderful sound worlds that one would never think of."

Table 3. Selected responses to Question 56: How much "unstable" or "non-deterministic" behaviour is there in your interface/instrument? How repeatable is each behaviour of the interface/instrument? Do you feel that you always have full control over it?

The survey asked what body parts are used in performing with the interface/instrument (Figure 2). Not surprisingly there was a strong preference for the use of both hands followed by (in descending order) fingers, fingertips, eyes, whole body, right hand only, feet, heel of palm, left hand only, mouth, head and lips. Entries listed under other included—any body part, arms, ankles, shoulders, ears and voice.

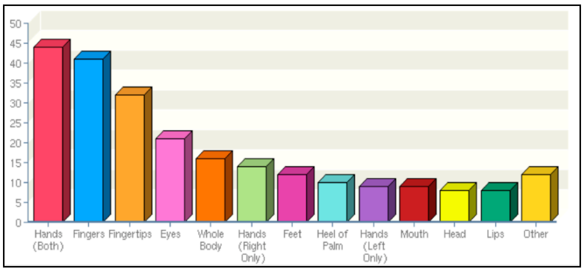


Figure 2. Question 18: What body parts are used in performing with the interface/instrument?

A recent study [12] carried out by the first author at the University of Western Sydney examined the fundamental control parameters utilised by expert musicians on traditional instruments. The model developed proposed the musical parameters; Dynamics, Pitch, Vibrato, Articulation and Attack/release as the focus of the physical instrument control, and of primary focus in achieving a well-developed instrumental tone, the principle concern for all musicians interviewed in the study. The model further identified four primary physical controls used to achieve musical outcomes—pressure, speed, angle and position.

Building on this research the survey asked participants to rank musical control parameters in order of importance (Table 4). Expression was clearly rated the most important while Vibrato was rated the least important. Dynamics, Tone Colour, Articulation, and Volume were grouped closely in the middle.

We also asked participants to select what types or qualities of movement are needed to play their interface/instrument (Figure 3). Position ranked highest (81.13%) followed by Speed (71.70%), Pressure (58.49%) and then Angle (49.06%). Under the other category were listed—Acceleration, Change, Coordination, Dexterity, Fingering Combination, Muscular Force, Rotation Surface Area, Torque, Types of Plucking and Bowing.

Value	Average Rank
1. Expression	0.898
2. Pitch and Intonation	1.218
3. Dynamics	1.380
4. Tone Colour	1.398
5. Articulation	1.407
6. Volume	1.422
7. Attack, Release, Sustain	1.550
8. Duration	1.593
9. Vibrato	1.936

Table 4. Question 9: Rank in order of importance with respect to your interface/instrument.

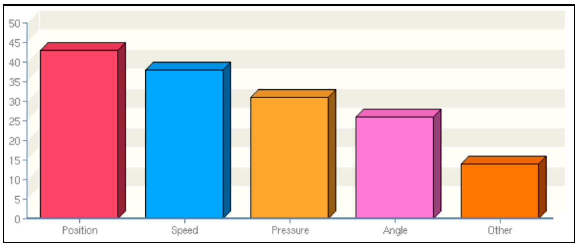


Figure 3. Question 19: What types/qualities of movement are needed to play the interface/instrument?

5. FUTURE PLANS

This paper presets just a brief overview of the data we have collected through the TIEM online survey. The broad scope and diversity of the field of DMI design resists fitting into a simple classification system. Yet a number of trends and groupings are starting to become apparent. The process of analysing the data is continuing and there are many sections of the questionnaire not covered in this paper. We are currently conducting an in depth qualitative analysis of the textural data captured and the TIEM online database will be expanded to include existing interfaces/instruments already documented in the literature.

6. REFERENCES

[1] D. Birnbaum, R. Fiebrink, J. Malloch, M. M. Wanderley, "Towards a Dimension Space for Musical Devices." *2005 International Conference on New Interfaces for Musical Expression (NIME'05)*, Vancouver Canada, pp. 192-95, 2005.

[2] J. Chadabe, "The Limitations of Mapping as a Structural Descriptive in Electronic Music." *2002 International Conference on New Interfaces for Musical Expression (NIME'02)*, Dublin, 2002.

[3] P. Cook, "Principles for Designing Computer Music Controllers". *2001 International Conference on New Interfaces for Musical Expression (NIME'01), CHI 2001*, Extended Abstracts, Seatle, 2001.

[4] S. Fels, L. Kaastra, S. Takahashi and G. McCaig. "Evolving Tooka: from Experiment to Instrument." *2004 International Conference on New Interfaces for Musical Expression (NIME04)*. pp.1-6, 2004.

[5] R. Gluck, "eShofar as a Culturally Specific Live Electronic Performance System." *SEAMUS*, Vol. 19, Society for Electroacoustic Music in the United States, Fall, 2006.

[6] E. M. Hornbostel and C. Sachs. Systematik der Musikinstrumente : Ein Versuch. *Zeitschrift für Ethnologie* Translated by A. Bains and K. Wachsmann under the title "A Classification of Musical Instruments." *Galpin Society Journal*, [1914] 1961.

[7] M. J. Kartomi, *On Concepts and Classifications of Musical Instruments*. Chicago: University of Chicago Press, 1990.

[8] D. J. Levitin, *This is your brain on music: the science of a human obsession*. New York, N.Y: Dutton, 2006.

[9] V. C. Mahillon. *Catalogue descriptif et analytique de Musée* Instrumental du Conservatoire Royal de Musique de Bruxelles, vol. I, second edition. Paris: Gand, [1880] 1893.

[10] J. Malloch and M. M. Wanderley, "The T-Stick: From Musical Interface to Musical Instrument." *2007 International Conference on New interfaces for Musical Expression (NIME07)*, New York, USA, 2007, pp. 66-69, 2007.

[11] E. R. Miranda and M. M. Wanderley, *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Middleton, Wis: A-R Editions, 2006.

[12] G Pain, I. Stevenson, and A. Pearce, "The Thummer Mapping Project (ThuMP)." *2007 International Conference on New Interfaces for Musical Expression (NIME'07)*, New York City, NY, 2007.

[13] J. Piringer. "Elektronische musik und interaktivität: Prinzipien, Konzepte, Anwendungen." Master's thesis, Technical University of Vienna, 2001.

[14] J. Pressing. "Cybernetic Issues in Interactive Performance Systems." *Computer Music Journal*, 14(2):12-25, 1990.

[15] D. Wessel, and M. Wright, "Problems and Prospects for Intimate Control of Computers." *Computer Music Journal*, 26(3), 11-22, 2002.

[16] T. Winkler, "Making Motion Musical: Gestural Mapping Strategies for Interactive Computer Music." *1995 International Computer Music Conference*, San Francisco, 1995.

ECO-STRUCTURAL DATA FORMS

CLASSIFICATION OF DATA ANALYSIS USING PERCEIVED DESIGN AFFORDANCES FOR MUSICAL OUTCOMES

Timothy Opie
Queensland University of Technology, 2 George Street, Brisbane, QLD, 4000

Box Hill Institute,
1000 Whitehorse Road,
Box Hill, VIC, 3128, Australia

Andrew Brown
Queensland University of Technology, 2 George Street, Brisbane, QLD, 4000, Australia

ABSTRACT

This paper explores a method of comparative analysis and classification of data through perceived design affordances. Included is discussion about the musical potential of data forms that are derived through eco-structural analysis of musical features inherent in audio recordings of natural sounds. A system of classification of these forms is proposed based on their structural contours. The classifications include four primitive types; steady, iterative, unstable and impulse. The methods presented are used to provide compositional support for eco-structuralism.

1. INTRODUCTION

"*One final and strangely bizarre possibility presents itself. We might imagine a music whose logic was based entirely upon the logic of the evolution of natural events as evidenced by natural morphologies of the sound-objects used.*" [17]

Eco-structuralism is a compositional technique that uses the structures or natural morphologies inherent in natural sounds as templates to generate new musical material. The templates are produced by attribute studies that examine specific functions within an audio event. Currently these attribute studies are used to gather information regarding the structural information of an audio event relating to amplitude, dominant frequency, timbre, and space. Each attribute is analysed separately, generating data specific to the attribute, without direct reference to other attributes. The data is collected in an XML file, which also contains information including audio format, recording and analysis dates, attribute analysis parameters, and descriptions of the original audio event that was recorded, including location and spectromorphology. The attribute data is then used in a strict serial fashion, abiding with the rules of eco-structuralism to define musical sequences, define musical articulations, to organise the macrostructure of a musical work, to resynthesize the original sound, synthesize new tones, or define some other musical event [13, 14]. Multiple data files (not necessarily related) can be used concurrently to define a multitude of musical parameters and create more complex musical structures.

Eco-structuralism has been informed by compositional techniques such as Musique Concréte, Serialism, soundscape composition and eco-composition. An ongoing debate amongst composers from these traditions relates to the use of sonic signification in musical works. Pierre Boulez stresses that any sound which carries too heavy an anecdotal burden should not be integrated with Musique Concréte [1]. However, this sentiment is not shared by R. Murray Schafer or other composers of the soundscape initiative. For example, Damián Keller describes eco-composition as soundscape composition in which the contextualisation of sound events and parameters are vital to their perceived meaning within particular societies and cultures. Keller has used eco-composition as a compositional tool to re-enact history from a sonic perspective, acknowledging that personal experience and stimuli will give each listener a personal understanding of the piece which will differ from others [6, 7]. In a similar spirit, Luke Windsor states that audio recordings carry "extra-musical" (or anecdotal) concerns, which insist upon being dealt with as cues for events [16]. These two opinions are sympathetic with views expressed by Luc Ferrari that the structural qualities of Musique Concréte could be used in conjunction with extra-musical concerns of the sonic events to bring the music and reality together to tell a story [2].

The role of sonic signification is made even more intricate in Eco-structuralism, as this compositional technique makes use of perceptual cues, but does so in a more schizophrenic manner than other forms of soundscape or eco-composition. This technique totally obfuscates the original recording but attempts to maintain structural elements that carry with them design potential. A brief glimpse at the technique might suggest that the origins of the audio recording are being eradicated in processes of deconstructing the sound recording into independent attribute data streams. This could be seen to result in wiping away all of what Trevor Wishart calls 'anecdotal' information through the data analysis and reduction process. On the contrary, however, eco-structuralism attempts to retain the aesthetic and perceptual signatures that were inherent in the original sound event. It strives to reveal the anecdotes inherent in sound recordings, even to emphasize them. The eco-structural composer employs recordings of natural sounds purposefully chosen as desired sonic events. This aesthetic interest in natural sound sources builds on the fact that nature already provides a limitless array of intrinsically interesting sounds, all with natural morphologies, that include many states of complexity and simplicity that provide a huge palette of possibilities. One of the problems with this sheer number of possibilities is finding ways of cataloguing and using the data in some effective manner. This paper presents a system for designating musical potential to structures in data generated by eco-structural analysis.

2. MUSICAL POTENTIAL IN GESTURAL MORPHOLOGY

There is a tradition, or history, of gestural morphology in composition upon which our work builds. In 1964 Stockhausen described 68 unique gestures within the score for Mikrophonie I. He included terms such as cracking, grating, groaning, and whimpering. It was claimed that “For the first time a perceptual equivalent to totally organized structure has been discovered, and it is particularly significant that this has been done with very simple means” [8].

Other composers have also made attempts at capturing morphological structure. Morton Subotnick developed the Ghost Box in order to capture subtle auditory gesture. The Ghost Box converted gestural information into control voltages which were then used to adjust the frequency or control the amplitude envelope. It is referred to as a ghost score because the score for the Ghost Box is a mono audio recording which is simply processed and translated, without being heard directly by the audience [9]. Subotnick's first piece utilizing the Ghost Box was Two Life Histories, written in 1977.

The UPIC (Unité Polygogique Informatique de CEMaMu) was a computer system designed by Iannis Xenakis that employed graphical drawn gestures to represent the progression of sonic parameters. It allowed the composer to draw various musical elements onto a large electromagnetic drawing board, using an electromagnetic pen. Xenakis completed his first piece for UPIC in 1978, entitled Mycenae Alpha [5].

It is instinctual to look for shape, form and structure in music and so it is not surprising that this theme persists. A particular advocate of composing with sounds based on ideas presented as shape, form and gesture was Trevor Wishart. He proposed that all sounds can be ascribed a particular shape, form, structure or gesture. Wishart identified categories for different types of gesture. In his book On Sonic Art he suggests a methodology for labelling sound morphologies. He suggests three basic states; Continuous, Iterative and Unsteady [17]. He also explains how these three types can be combined with each other to form more complex variations on the type. Dennis Smalley expanded greatly upon the ideas of shape form and structure in his research upon spectromorphology. He added many new descriptors and models that can be used to analyse and identify sound perceived within electroacoustic music [18]. These include descriptors for motion, growth, behaviour, spectra, space and density. This was later expanded and perhaps superseded by Smalley’s models of perception on acoustic space and form [19]. This model measures events against space, rather than time, as spectromorphology and most other analysis tools do, and allows the listener to identify sonic events in regards to how they relate to other events within the space, or around the space. Within this model Smalley has identified many forms of spaces from ecological spaces to digital spaces [19].

Another method for classifying data lies in the way we want to be able to use the data. When analysing an eco-structural data stream, a question that comes to mind is, what is the musical potential of this data? One useful perspective on potential is to think of the data as a signifier or a perceived affordance [12]. The term affordance was coined by J. J. Gibson [4] to describe an understanding of the opportunities presented by the world in terms of how we may personally interact with it. The affordance of an object or environment is influenced by our accumulated experience of similar objects and by our current needs and motivations. Affordances are somewhat subjective, based on our perception of the opportunities for action we choose how to interact with the object, usually in an intuitive non-calculated manner. “When affordances are perceptible, they offer a link between perception and action” [3]. In general, affordances resonate with our intuitive knowledge of how to interact successfully with the world around us. As Gibson stated “affordance cuts across the dichotomy of subjective-objective and helps us understand its inadequacy” [4]. Luke Windsor has extended this research into the auditory domain by discussing the ecological approach to musical perception, which he has explained using the spectromorphology framework, bridging these ideas and applying affordance perception to acousmatic music [16, 20, 21]. Damián Keller has created musical installations using an eco-composition approach, which seek to afford an ecological response [6, 7].

The case can be made however that sound does not afford anything, it merely suggests or notifies us of prospective affordances, which may or may not exist. In order to overcome this potentially problematic usage of term affordance we turn to the work on design signifiers or perceived affordance for design by Donald Norman. On discussing design for icon-based interaction on a computer, Norman states, “Those displays are not affordances; they are visual feedback that advertise the affordances: they are the perceived affordances” [12]. We suggest that it is useful and appropriate to think of musical structures and sonic morphologies in the same way; as offering *perceived* affordances. Norman goes on to state that, “Symbols and constraints are not affordances. They are examples of the use of a shared and visible conceptual model, appropriate feedback, and shared, cultural conventions” [12]. A designed object can signify something if it was designed to act in a specific routine, and if social and cultural conventions agree that that was the correct routine. Norman suggests that “Designers can invent new real and perceived affordances, but they can not so readily change established social conventions” [12]. With this in mind, it becomes important to identify potential from within established social and cultural conventions.

Fortunately music has very long standing conventions of form. The notions of form provided by gestural morphology and spectromorphology provide an excellent set of identifiers for classification, however we are cautious of using spectromorphology as a classification method because it relates primarily to sound as it occurs in electroacoustic composition. Focusing on space-form, as defined by Smalley, provides excellent identifiers for classification. However, the problem associated with each of these ideas is that they deal with the identification and perception of sound, particularly the function, causation and interrelation of spectra or sonic events, within an electroacoustic composition [20]. Listening is a key feature in the function of these frameworks.

Unlike Acousmatic composers, the eco-structuralist composer however does not actually compose with the original sound recordings. The data files contain no audio. The eco-structural XML file contains a spectromorphological analysis of the original sound event. The data that is used for composition contains only a subset of data, that pertaining to a particular attribute of the original sound. Any spectromorphological or space-form analysis of the original sound will become redundant after the data reduction process. It may provide hints as to how the data may be shaped, but that depends on which sound attribute has been analysed and what parameters were used in the process. We need new categories to identify the musical potential of the data, unconstrained by the source sound. We also need to ensure that the composer understands that the data cataloguing terminology refers to the morphology or musical potential of the data, and not to the audio event from whence it came.

In creating a classification model, we have drawn on some of the terminology used by Wishart and added some components of spectromorphology to give the terms greater meaning, but these terms are used as a reference to the data, not the sound, and therefore we have been careful to avoid terms that refer to specific sonic or auditory phenomena. We have also been careful in choosing terms that are too culturally loaded, and which suggest specific musical potential.

3. CLASSIFICATION

After examination of a large number of data sets, a classification of morphological types has been derived similar to the three suggested by Wishart, although we use some different terms to make the meanings more clear. Also a fourth type has been identified. These types will be introduced and discussed in this section. Each form also contains configuration types that help define the musical potential more accurately.

Each type will be labelled as a form, as this is more suggestive of an encompassing type, which holds many variations on similar themes. It must be noted that there is no ideal form in eco-structuralism as change and variation is integral to all forms of music.

Eco-structural forms are currently classified by generating a graph of the data, and then visually examining and comparing this to the forms below to find the best match. They need to be analysed in a visual manner because they no longer contain an audio component. Other possible methods of analysis will be discussed later. The compositional potential applies only to the attribute data. The data could be used to create tones, resynthesise sounds, control synthesiser and effect parameters, designate musical notes, dynamics, articulation and musical sequences, or control the macrostructure of a musical composition, etc.

3.1.The Steady Form:

Description: Relatively flat data. Can contain some minor discrepancies.

Graphical Example: Data derived from the spatial attribute study of a very short stereo segment of a recording of an ocean wave as it approaches, but before it crashes.

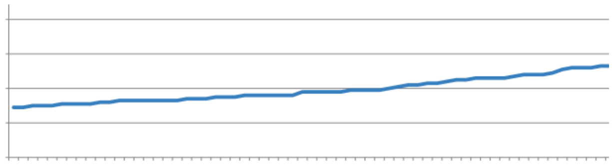


Figure 1: The Steady Form in Ascent.

Configurations: Ascent, Plane, Descent.

Compositional Potential: Stasis, Stability, Drone, Basso Ostinato.

Notes: The steady form is similar to what Wishart termed continuous, although continuous can be misleading as all sounds can be continuous in any number of forms. Finding stable forms in natural sounds is quite rare. R. Murray Schafer points out that only man made devices will emit an unchanging audio signal [15]. This is of course far from ideal as the musical potential would be quite dull, unless it was combined with something less steady. The term stable might have been used, but stability implies a certain accuracy and reliability that steady does not. A signal can be steady without being stable, reliable or accurate, which is more functional in this context.

3.2.The Iterative Form:

Description: Mostly patterned data.

Graphical Example: Data derived from the amplitude attribute study of a mono recording of waves crashing on the beach.

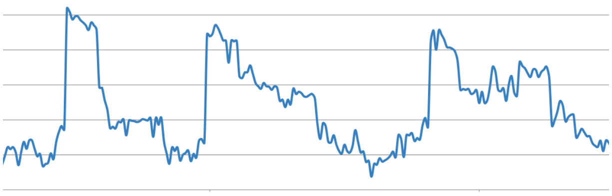


Figure 2: The Iterative Form, Rectangular.

Configurations: Rectangular, Sinusoidal, Triangular.

Compositional Potential: Motif, Melody, Harmony, Macrostructure, Effects, Curves, Smooth Transitions, Repetition.

Notes: This is the same term used by Wishart. The form is characterized by identifiable symmetry and repetition. The presence of iteration, or repetition, in music is well understood. Indeed the ability of humans to tolerate repetition in music is surprising given our quick boredom with it in other media such as language, visual design, and haptics.

3.3.The Unstable Form:

Description: Fluctuating data in a mostly rough manner, but still with some linearity.

Graphical Example: This data was derived from the dominant frequency attribute study of a mono recording of a babbling brook.

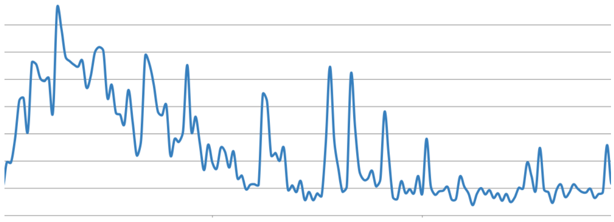


Figure 3: The Unstable Form, Sinusoidal.

Configurations: Fluctuating, Rectangular, Sinusoidal, Triangular.

Compositional Potential: Fluctuation, Solo, Macro-structure, Effects, Surprise, Variety.

Notes: Wishart refers to this form as unsteady. The term unstable has been used instead of unsteady as it is more familiar. Data with this form can be used to provide interest and variation particularly in moderate amounts and as a modulation source for steady or iterative forms.

3.4.The Impulse Form:

Description: A discrete impulse that does not match the surrounding data.

Graphical Example: Data derived from the amplitude attribute study of a mono recording of a campfire crackling.

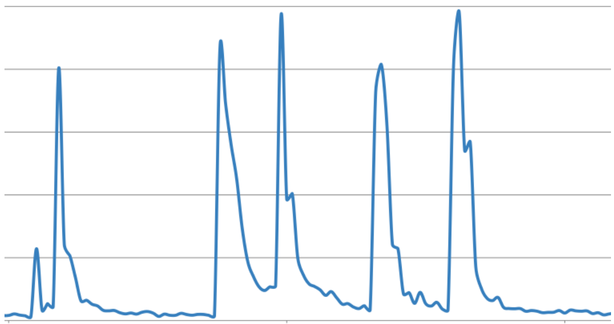


Figure 4: The Rapid Sharp Impulse Forms.

Configurations: Blunt, Sharp, Rapid, Rare.

Compositional Potential: Switch, Gate, Trigger, Pulse.

Notes: The impulse form is one not discussed by Wishart, although after analysing various data streams, it is apparent that this phenomena crops up quite regularly. If the impulse occurs multiple times in one stream it may be useful to describe the periodicy of the impulse. The actual sample used here in the example is a graphical representation of a crackling campfire. The sound will be familiar to everyone. From a distant perspective the sound of the crackling fire seems quite steady. There is a lull to it that seems constant and unchanging. On closer inspection however, there appears to be two layers at work. There is a steady floor, which is the dim humming of the small flames, but interspersed is an unstable impulse which punctuates the sound. Each impulse is quite similar, so it could be stated that there is an iterative unstable impulse. That impulse is the sound of a crackle. Because they occur quite regularly and are similar in impulse, they only appear as an impulse on close inspection. Another example of a similar combined form that occurs naturally is a thunder storm. The rain forms a steady floor, which is punctuated by unstable impulses of thunder.

As the fire sound example illustrates, these archetypal forms will rarely be found in data from natural sounds. Much more likely is the use of this classification language as a way of describing aspects of a data stream and tendencies of the structural morphology. Clearly there will be a vast number of combination forms, including second order descriptions of stable or iterative combinations and so on.

3.5.Combinations:

Description: Data that seems to switch between two or more forms.

Graphical Example: Data derived from the dominant frequency attribute study of a babbling brook.

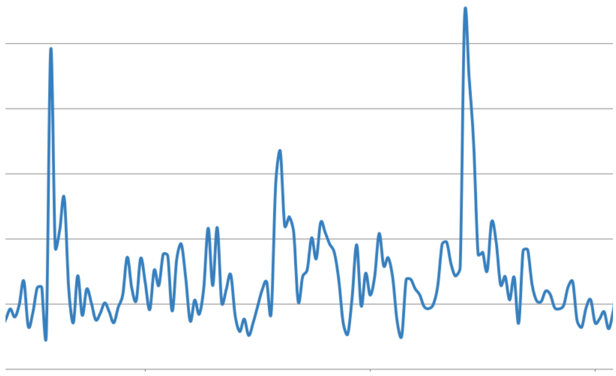


Figure 5: Unstable Sinusoidal with Fluctuating Sharp Impulse Combination

Configurations: Fluctuating, Dissipating, Accumulating.

Compositional Potential: Solo, Variety, Fluctuation, Macrostructure.

Notes: The use of the combination has already been described in the previous form, whilst explaining the thunder and crackle impulses, there are however multiple combinations that could occur. This example structure could be described as unstable with unstable impulses.

There is a final point to make about the classification of a data stream as exhibiting particular structural characteristics, and therefore particular compositional opportunities. The scale at which the data (or sound) is examined may change our perception of morphological characteristics. Consider again the example of the fire sound. As depicted it appears as an unstable series of impulse forms. However, the time slice under review is very short, this is a description of its micro structure. Zooming out to a larger time scale, a crackling fire provides a stable, but interesting, background texture. This is the kind of analysis that is assisted by the language of form categories and highlights the creative utility of this approach for the eco-structuralist composer.

4. APPLICATION

The eco-structuralist data structure is contained within an XML framework with labels identifying many characteristics of the sound and the parameters used to capture the data. Form classification is therefore an intrinsic element of the data structure. The classification system creates a pool of data by which a composer can more easily make compositional choices without having to constantly re-evaluate the data. Once a classification is made, the classification can be added directly to the XML data. When data is first extracted it is given the classification form "unknown", so that the composer knows it still needs classifying. The four forms steady, iterative, unstable and impulse can be assigned to any data stream with any number of permutations. Once the classification has been made the data file can be placed with the other data files.

There may be a pool of hundreds of data files that are now ready to be accessed by the composer. Data forms can be easily retrieved using a simple XML tag search. Using the terms given, a composer can search for data that for example has a "steady descent" in the *Form* tag. They may retrieve seven data files in the search that have a steady descent. Using the rest of the information in the eco-structural XML file the composer can then make a more refined decision, based on the description in the file. If the composer has already decided to work only with beach data files, they can include "beach" as a search term for the *Location* tag as well as steady descent in the *Form* tag. This may reveal two data files, which can then be investigated further by the composer, using the rest of the information available in the data file.

As we have mentioned, finding data in which these forms appear in a pure state has so far been very elusive. There are definite moments when data is in a pure steady or iterative form, but sometimes that moment is quite fleeting. In one sense this is a good thing, as was noted earlier, change and variation are integral parts of music. When sections of steady or iterative data have been identified they can definitely be used as motifs, sustained pads and chords structures. However, it can make the classification of a data stream in the XML framework approximate.

Going back to the gestural form and categorization of the crackling fire, it was described as having a steady floor, punctuated with an unstable impulse. There are a number of musical potentials that can be explored. Firstly, the steady floor offers a good sustained pad control mechanism. If we combine three separate data structures from recordings of crackling fires, we will find the floor is always different. These can be used to create sustained chords on a synth pad. As the floor steadily tends to ascent and descend in some examples, it will create an evolving synth pad sound that works as an interesting musical foundation. The impulses could then be used for triggering. It could trigger notes based on the height the impulse reaches, or it could trigger another note or chord derived from another structure. Another musical device is that it could trigger an inversion of the pad chord. There are many different musical potentials to consider. Each form stores different potential.

If the same crackling fire structure were to be used in a musical macrostructure, the floor could indicate the key or pitch class for each musical section, with the impulse indicating the start of each new section. For the macrostructure to be more effective, it would need to be scaled over time, which is a simple task for eco-structure [14].

These applications are just suggested approaches for working with eco-structural data forms, but composers may prefer different approaches to the musical organisation of their works.

5. FUTURE RESEARCH

As an extension to this classification process, it would be beneficial if the classification process could become automated. The XML Form tag already exists, to be edited manually. The task remaining is to write the algorithm to identify the forms. As automatic identification of forms is not a critical component for our research on eco-structuralism at this time, we have deferred its exploration. Another extension would be to provide segment form tagging in the XML file so that sections within the data stream could be independently tagged. An advantage of the automated process would be to make the process of selecting musical material much quicker. It could also provide an accurate combination of multiple forms within a single structure. As a drawback however, it may produce a long list of forms that alternate throughout the structure.

6. CONCLUSION

Eco-structuralism is capable of generating representative structures from natural sounds and sonic events. By creating a cataloguing system based on identification of gestural forms we have provided a method of filtering and managing the large amounts of data often generated during analysis. It also suggests a descriptive framework for thinking about the features and opportunities of the data. We have provided a theoretical foundation for this classification that suggests there are perceptual affordances in gestural morphologies. Providing a classification, links the possibilities for compositional design with the data features arising from eco-structural analysis. The perceptual affordances, or design opportunities, presented by the data form classification, we suggest, will both assist the compositional process and continue to reinforce the link between the aesthetic interest of the sound source and the resulting composition. In a practical way the eco-structural data form classification should provide the composer more time to work with sound, rather than sifting through data. When the composer has a musical idea, they can match it against musical potential identified in the data Form tags and use the found structures as building devices to create a musical work.

7. REFERENCES

[1] Boulez, P. (1971) *Boulez on Music Today*. Faber and Faber, London

[2] Pauli, H. (1971) *Für wen komponieren Sie eigentlich*? Fischer, Frankfurt Am Main, Germany.

[3] Gaver, W. (1991) Technology Affordances. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, p.79-84. New Orleans, LA, USA.

[4] Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum, New Jersey.

[5] Harley, J. (2002) The Electroacoustic Music of Iannis Xenakis. In *Computer Music Journal*, Volume 26, Number 1, Spring 2002, pp. 33-57

[6] Keller, D. (2004). *Paititi: A multimodal journey to El Dorado*. Doctor in Musical Arts Thesis. Stanford, CA: Deparment of Music, Stanford University.

[7] Keller, D. & A. Capasso (2006) New Concepts and Techniques in Eco-Composition. In *Organised Sound* 11(1): 55–62. Cambridge University Press, UK.

[8] Maconie, R. (1972) Stockhausen's Mikrophonie I: Perception in Action. In *Perspectives of New Music* 10, no. 2 (Spring-Summer): 92–101.

[9] Subotnick, M (2003) *Morton Subotnick: Technical Notes*. http://www.mortonsubotnick.com/program.html#technical. Accessed: 4/5/2009

[10] Norman, D. A. (1999). *Affordance, conventions, and design*. In *Interactions* 6(3): 38-43.

[11] Norman, D. A. (1988). *The Psychology of Everyday Things*. New York: MIT Press.

[12] Norman, D. A. (2008) Signifiers, not affordances. In *Interactions* 15(6): 18-19.

[13] Opie, T and A. R. Brown (2006) An Introduction to Eco-structuralism. In *Proceedings of the International Computer Music Conference*. New Orleans, LA, USA.

[14] Opie. T. (2008) Rendering Soundscapes. In *Proceedings of the Australasian Computer Music Conference 2008*, Sydney, Australia.

[15] Schafer, R. M. (1993) *The Soundscape: Our Sonic Environment and the Tuning of the World*. Inner Traditions International, Limited

[16] Windsor, W.L. (1995). *A Perceptual Approach to the Description and Analysis of Acousmatic Music*. Doctoral Thesis, City University.

[17] Wishart, T. (1985). *On Sonic Art*. New York: Imagineering Press.

[18] Smalley, D. (1997) Spectromorphology: Explaining Sound-Shapes. In *Organised Sound* 2(2): 107–126. Cambridge University Press, UK.

[19] Smalley, D. (2007) Space-Form and the Acousmatic Image. In *Organised Sound* 12(1): 35–58. Cambridge University Press, UK.

[20] Windsor, W. L. (1997) Frequency structure in electroacoustic music: Ideology, Function and Perception. In *Organised Sound* 2(2): 77–82. Cambridge University Press, UK.

[21] Windsor, W.L. (2000) Through and around the acousmatic: The Interpretation of Electroacoustic Sounds. In *Music, Electronic Media and Culture.* Edited by Simon Emmerson. Ashgate, Sydney, Australia.

ARTIST TALK: GHOST WINDS TALKING

Lydia Ayers
Tower 14, Flat 7A
HKUST
Clear Water Bay, Kowloon
HONG KONG

Andrew Horner
Department of Computer
Science and Engineering
HKUST
Clear Water Bay, Kowloon
HONG KONG

1. INTRODUCTION

One day the wind blew across the open bathroom window as though it were playing harmonics on a giant flute. Ayers recorded the sounds, which included amazing glissandi and harmonics, a "staccato whistle" and a "Chinese opera soprano," which was especially expressive filtered and fluttered. Amplified rain clicks and a car door slam produced percussive effects. The singing, whistling and general moaning of the wind reminded her of ghost voices, and inspired her to contrast those sounds with the ghosts of real composers' voices in her composition, ***Ghost Winds Talking***.

2. THE GHOST VOICES

We analyzed speech samples from four composers: Harry Partch, Nadia Boulanger, Lou Harrison and Aaron Copland. We don't know whether anyone else can recognize their vocal timbres, but perhaps that is appropriate for ghosts.

The Csound additive synthesis model used in our previous *didgeridoo* synthesis design [1] was a good starting point for this project, but it needed some refinement. The new model morphs together a random group of vocal timbres as it slurs a pair of pitches.

2.1. The Spectral Snapshots

From the waveform of a fragment of Harry Partch saying "it's a kind of English" [2], we chose a group of three sounds, long "a," "k" and long "i." After picking a snapshot time for each sound, we examined the phase vocoder analysis graphs.

The vocal formants of the long "a" and "i" vowels are quite distinct. The "k" sound serves as a noisy bridge between the surrounding vowel sounds. The amplitude is distributed more evenly among the harmonics than for the two vowels, but the formants for the vowels bleed through a little bit.

2.2. Morphing the Sound Segment

We morphed the sounds together to make part of the combined sound, "a kind." Each harmonic is one component signal for the spectrum, with its own frequency and amplitude line segments for the group of morphed timbres. The vowels "a" and "i" sound reasonably like the timbre of Partch's original voice, but using this method with one spectral snapshot does not capture the full quality of the noise in the "k" sound. If we wish to replicate Harry Partch speaking those words, we'd be better off using the recorded samples. If we make Partch say something else, the words are not totally clear, but the result is useful enough for our musical purposes.

3. MUSICAL EXPRESSION OF THE DESIGN

Ayers used the ghost voices in her composition, *Ghost Winds Talking*. She stored the amplitude data from the graphs of the sounds in two groups of Csound function tables, one for consonants and one for vowels. The ghost sound typically begins with a sound from the consonant bank to simulate an articulation, and then morphs a string of vowel timbres. It also contains a pitch-bend inflection, either up or down. We used random cells in a spreadsheet program to choose the vocal sounds and vary their parameters. Repeated performances of the same sound will have varied spectra, making the performance more expressive. Although the speech sounds like unintelligible English, it is very suitable for the ghost voices used in the composition.

4. ACKNOWLEDGEMENTS

This work was supported by the RGC Competitive Earmarked Research Grant 613806 and 613508. A more detailed paper on this research will appear in the *Proceedings of the International Computer Music Conference*, 2009, Montreal.

5. REFERENCES

[1] Ayers, L. and Horner, A. 2007. "Didgeridoo Synthesis Using Timbre Morphing," *Australasian Computer Music Conference 2007*, Australasian Computer Music Conference. Canberra, Australia, pp 18-25.

[2] Partch, H. 1995. *Enclosure Two: Harry Partch: Historic Speech-Music Recordings from the Harry Partch Archives*. Innova 401.

INFORMATION SHARING IN NETWORKED MUSIC APPLICATION

Norikazu Mitani
Arts and Creativity Lab, IDMI
National University of Singapore

Lonce Wyse
Arts and Creativity Lab, IDMI
Communications and New Media Programme
National University of Singapore

ABSTRACT

Two key issues for facilitating networked ensemble music are information sharing and message routing. The Bridge application has been developed for networked music composition and performance which implements these functions. The Bridge is designed to support networked music composition and live ensemble performance, supporting dynamic connectivity between applications or physical devices and executing message transfer between applications and devices.

1. INTRODUCTION

Electroacoustic music is often part of rich media environments that may include multiple channels of display and control through graphics and physical devices. Creating a work involving the coordination and communication across many media components is a significant challenge typically involving lots of technical arcane. In particular, there may be many components (instruments, devices, graphical objects) designed by composers or designers with different interface capabililties for real time interaction. If these components are distributed across a network, the challenge of managing the addressing, message protocols, and communications between all the components can be daunting. The problem calls for automated support so that composers can focus on the musical issues of interaction rather than the technical issues.

In the context of networked music, Weinberg [1] proposed a theoretical framework for music network architectures. In terms of that framework, we have designed the Bridge application which supports synchronous dynamically reconfigurable decentralized or centralized networks. We consider "message routing" between applications and devices to be one of the most important processes for both composition and live performance. A second related functionality, that of "information sharing", is necessary for effective message routing, particularly in a dynamic environment. When multiple applications and devices are gathered in a network, the list of component inputs and outputs for designing and manipulating message routes between applications and devices must be shared and readily accessible to all.

2. RELATED WORKS

The NRCI[2] PD tool suite has several similarities to the Bridge is spirit and intent. It is designed for exchanging music control data and communication between networked musicians and uses OSC (Open Sound Control)[3] as its underlying protocol.

NRCI includes three important protocols. The request protocol provides the capability to initiate specific control data broadcasting from another networked musician. The command protocol supports sending command names and values to a target networked musician. Finally, a chat protocol supports the exchange of text messages between all musicians in one network.

NRCI PD tool suite users can request types of control data: pitch, amplitude, duration and onset, but does not provide support for advertising the availability of other streams named by the users. This limitation is mitigated somewhat by the ability to broadcast data with arbitrary names (e.g. modulation index), but effective use of this data requires negotiation between sender and receiver and/or preplanning making it difficult to use in an improvisatory context. These issues are addressed in the design of the Bridge.

3. BRIDGE APPLICATION OUTLINE

The Bridge embodies our approach for networked music design and performance in a small local network [4] with

- a connection interface for music applications, video applications and physical devices,
- a flexible I/O parameter design for application/device designers,
- sharing application I/O parameter information,
- message routing between applications and devices.

The Bridge works as a messaging hub/router of OSC datagrams. This application supplies an interface for creating connections between audio/video applications or physical devices. Currently, applications and devices that use OSC (Max/MSP, PD, Super Collider, Processing, etc.) and MIDI interfaces are supported.

Bridges communicate with each other be sending either configuration information (which changes dynamically) or music control data via "sender" and "receiver" parameter units. Each application or device with sender parameter registered with the Bridge can send music control data to other application/device with registered receiver parameter (Figure 1). The Bridge is implemented in Sun Java SE 6 for platform interoperability. It has been tested on systems running Microsoft Windows and Mac OS/X.

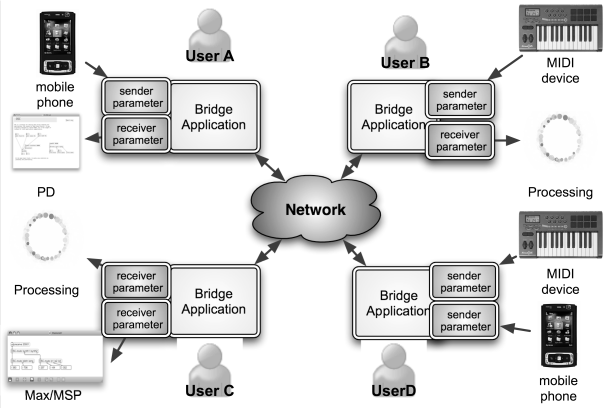


Figure 1. Bridge overview. Each user may have multiple devices or applications that register performance data sender and/or receiver objects with the Bridge.

4. INFORMATION SHARING

The Bridge is responsible for communicating information about the available senders and receivers among all performers.

When a new Bridge instance wants to join the network, it communicates with the other Bridges with a UDP broadcast message containing

- Bridge information (its name, IP address and peer-to-peer IP port),
- I/O parameter information (sender/receiver parameters),
- patch information (control data routing from sender to receiver).

Bridges all maintain the same information at all times. Bridge users can dynamically modify sender/receiver parameters during music performance. Bridge users can also add and delete message routes between any sender and receiver at any time. Modified parameter and routing information will immediately be shared with all other Bridges. These three pieces of information listed above are visualized in control panel of the Bridge which also serves as a graphical user interface for changing the network configuration.

When Bridges receive music control data from sender parameters locally or from other Bridges, the configuration information is locally updated so that each Bridge knows how to map and route future data (Figure 2), and so that the Bridge can provide visualization and a graphical interface for manipulating the current architecture. This is how the Bridge routing functionality is supported by information sharing.

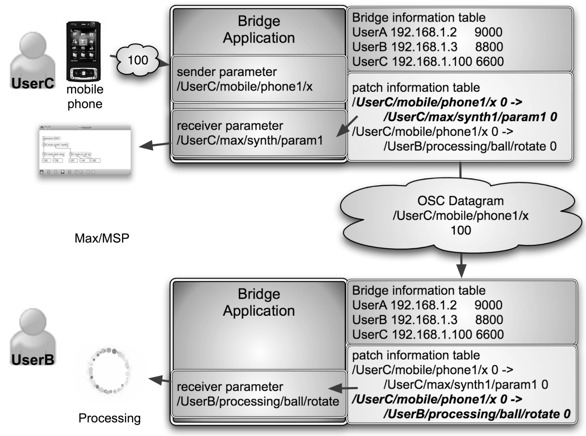


Figure 1. Information sharing with the Bridge. The Bridge determines control data destinations according to the Bridge information table and patch information table.One patch contains both sender and receiver information.

5. EVALUATION AND FUTURE WORK

In this paper we have described the information sharing aspects of the Bridge architecture for networked music. We have successfully used the input/output parameter list sharing and message routing by the Bridge in several demonstrations. In the demo, each Bridge users can append/delete parameters and change the patch at any time. Our next step is to get the Bridge system in to the hands of composers and media creators for real-world testing and user feedback. Increasing the range of device and application support is another important topic also. One of our targets is a TCP socket interface so that the Bridge support Action Script 3 for Flash and Flex applications. The ultimate goal of the system is hide the technology and support research in musical communications.

REFERENCES

[1] Weinberg, G. “Interconnected Musical Networks: Toward a Theoretical Framework" *Computer Music Journal* 29/2 2005.

[2] Burns, C., and Surges, G. "NRCI: Software Tools for Laptop Ensemble." *Proceedings, International Computer Music Conference* Belfast, Northern Ireland, 2008.

[3] Wright, M., and Freed, A. “Open Sound Control: a new protocol for communicating with sound synthesizers”, *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997, pp 101-4.

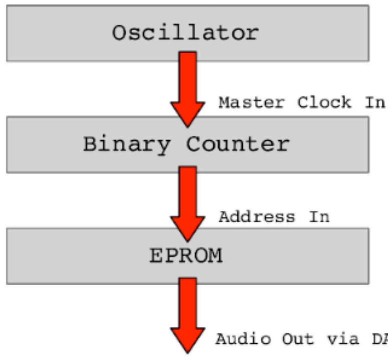
[4] Wyse, L. and Norikazu, M. “Bridges for networked musical ensembles”, *Proceedings of the International Computer Music Conference*, Montreal, Canada, 2009.

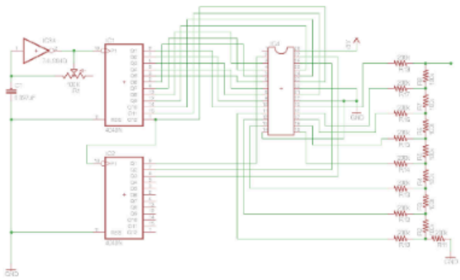
EPROM MUSIC

By Sebastian Tomczak
Electronic Music Unit, Elder Conservatorium of Music, University of Adelaide

CONCEPT

The idea behind EPROM MUSIC is to explore digital sound and music that does not require any programmable parts of any variety. An oscillator is connected to a binary counter, which in turns clocks out samples stored in an EPROM chip. A DAC circuit provides an audio output.



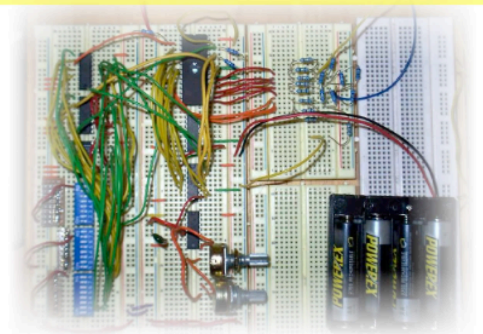


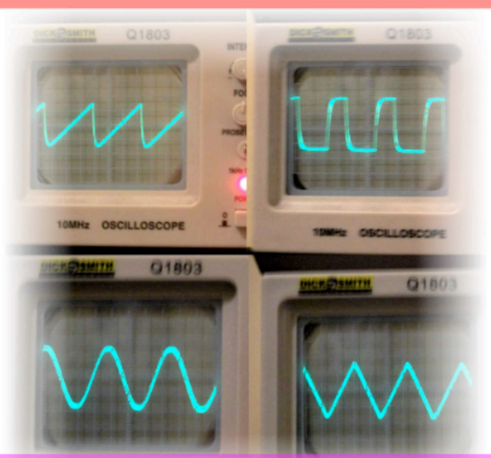
SAMPLE LOOP PLAYBACK

A short loop can be loaded onto the EPROM , and played back at a speed / pitch determined by the frequency of the initial oscillator. By manipulating the address inputs of the EPROM, sections of the loop can be isolated and stuttering and glitching can be achieved easily.

DRUM MACHINE

Multiple drum samples can be loaded onto the EPROM, and then a series of buttons or switches can trigger the address range corresponding to each individual sound. Thus, a simple logic-based drum machine can be constructed. The highest address bit can toggle between two separate drum kits.





WAVEFORM SYNTHESIS

A set of many waveforms can be loaded onto the EPROM. The waveforms are shaped and formatted using a Max/MSP patch, specially designed for the task. Each waveform contains 128 samples. The higher address bits are then used to select between waveforms. By manipulating the address bits, grain and glitch-like artefacts can be introduced into the audio output.

EXTENSION

This concept can be extended easily. The system is modular, in that multiple EPROMs can run from the same clock. Different grooves and triplets can be achieved by adding a decade counter, for instance.

Schematics, patches etc: http://little-scale.blogspot.com/search/label/eprom%20music

ISSN: 1448-7780