Ashitaka: An Audiovisual Instrument

Niall Moody MMus BEng

Submitted in fulfilment of the
requirements for the degree of Doctor of Philosophy

Department of Electronics and Electrical Engineering
Engineering Faculty
University of Glasgow

28/09/2009

Abstract

This thesis looks at how sound and visuals may be linked in a musical instrument, with a view to creating such an instrument. Though it appears to be an area of significant interest, at the time of writing there is very little existing - written, or theoretical - research available in this domain. Therefore, based on Michel Chion's notion of synchresis in film, the concept of a fused, inseparable audiovisual material is presented. The thesis then looks at how such a material may be created and manipulated in a performance situation.

A software environment named Heilan was developed in order to provide a base for experimenting with different approaches to the creation of audiovisual instruments. The software and a number of experimental instruments are discussed prior to a discussion and evaluation of the final 'Ashitaka' instrument. This instrument represents the culmination of the work carried out for this thesis, and is intended as a first step in identifying the issues and complications involved in the creation of such an instrument.

Contents

List of Tables

List of Figures

List of Accompanying Material

DVD:

- Thesis Videos 1-10;

    1. Footsteps Original
    2. Footsteps Sine
    3. Footsteps Abstract
    4. Linear Motion Original
    5. Linear Motion Sine
    6. Linear Motion Abstract
    7. Linear Motion No Envelope
    8. Linear Motion On Screen
    9. Periodic Pulse Example
    10. Discontinuous Jump Example

- Ashitaka in Performance;
  - Mononoke Hime Improvisation (with the performer visible).
  - Second Improvisation (showing only the instrument's visual output).

CD-ROM:

- Heilan source code (including that of Ashitaka and the various experimental instruments)
- Heilan source code HTML documentation
- Heilan Windows binary executable
- Heilan OSX binary executable
- Heilan Front End source code
- Interface microcontroller source code
- SerialToOSC program, plus source code
- Particle Fountain VST plugin for Windows
- Brush Strokes VST plugin for Windows, plus source code

Acknowledgements

I would like thank, firstly, my two supervisors, Dr. Nick Fells and Dr Nick Bailey, without whom this thesis would have been a very different (and surely lesser) piece of work. In addition, I am grateful to everyone who worked at the university's Centre for Music Technology while I was there. Particularly Tom O'Hara, who designed and the built the electronics for the instrument's physical interface (as well as the eight-speaker cube I used for my experiments), and whose assistance throughout the project was invaluable. I would also like to thank Craig Riddet (now Dr.), who shared a flat with me for most of the four years I worked on Ashitaka, and who appears as a pair of legs in two of the videos on the accompanying DVD. Finally, I am indebted to my parents and brother, without whom I may have never ended up studying for a PhD.

Author's Declaration

I hereby declare that I am the sole author of this thesis, and that it has not been submitted in any form for another degree at any other university or institution.

- Niall Moody (21/9/08).

Chapter 1

Introduction

Though it has not always found its way into mainstream culture, there is a long tradition of artists who have sought to link sound and visuals for musical purposes. In doing so, many of these artists have built audiovisual tools equivalent to musical instruments, from Father Louis Bertrand Castel's Clavecin Oculaire in the sixteenth century to Golan Levin's more recent Audiovisual Environment Suite. With the rise of film, television and computers throughout the twentieth and twenty-first centuries, there would seem to be more interest than ever in audiovisual interactions. Having said this, there does however appear to be very little in the way of research available which examines the ways sound and visuals may be linked in a performance context. As a result, any artist or instrument designer starting out in this field essentially has to start from scratch.

This thesis aims to investigate the ways in which sound and

visuals may be linked in the design of a musical instrument. Beyond this it aims to establish a starting point for further work in the field, and to go some way to correcting the relative lack of information available for artists and instrument designers interested in the combination of sound and visuals in a performance context. As proof of concept an audiovisual instrument - Ashitaka - was developed and will be discussed in the thesis.

The thesis starts with an extended Background chapter, which tries to cover as wide a range as possible of artists involved with, and research related to, this particular field of audiovisual interactions. The focus is primarily on media which use abstract visuals however, as the visuals for the final instrument will be abstract, and time constraints have meant that more theatrical media like opera and dance are largely absent from this thesis. The chapter begins with a look at the various artforms which incorporate sound (or music) and vision in some way, starting with a brief look at the influence of music on painters such as Kandinsky. From here we move on to abstract film, in particular the so-called 'visual music' tradition of artists such as Oskar Fischinger and the Whitney brothers. This leads onto a brief discussion of the influence of visual music ideas on mainstream cinema, followed by a more detailed look at music video. The more recent phenomenon of VJing is given a brief examination, after which a number of artforms are discussed, under the heading 'Computer Art'. The term 'Computer

Art' can refer to any number of artforms, but is used here to refer to a specific audiovisual subset, including the demoscene, music visualisation software, and computer games. From here we move on to an investigation of research more directly concerned with the design and construction of musical instruments. This starts by outlining the concept of a 'Digital Musical Instrument', along with the specific issues involved in creating an instrument where the means of excitation is separated from the sound generating mechanism, then looks at the complex (and essential, in a Digital Musical Instrument) issue of mapping. A number of existing instruments (both purely sonic, and visual/audiovisual) are also examined. To finish, we delve into psychological research, to look at the ways in which sound and visuals may be linked in the human brain.

Following on from this, we come to the idea that forms the basis for most of my work in this thesis; Michel Chion's notion of synchresis. Chion's concept is elaborated upon with some speculation as to how and why it works, with a view to defining certain principles I can use to create synchresis in my own audiovisual instrument. This is accompanied by a brief look at psychological research which focuses on roughly the same area of audiovisual perception or illusion.

After this outline and elaboration of synchresis, I then attempt to set out a mappings framework by which visual motion can be mapped to aural motion and controlled in a performance context. My first, film-derived attempt (based on the work of foley artists)

is discussed, together with the reasons I ultimately found it unsuitable for my purposes. This is followed by a look at the second framework I developed, and which is implemented in the final Ashitaka instrument. This chapter also includes a brief examination of techniques which may be used to detect particular gestures in realtime.

Before moving on to a discussion of the development and implementation of the Ashitaka instrument, I take some time to map out the various artistic and technical issues raised by the previous chapters, and offer some suggestions as to how they might be resolved.

The discussion of Ashitaka's implementation begins with a look at the software environment - Heilan - I developed as a base for it. Heilan's various capabilities are enumerated, together with a brief examination of the program's code structure.

In the process of developing the mappings frameworks mentioned previously, I developed eight experimental audiovisual instruments, prior to the development of Ashitaka. For the thesis, each instrument is discussed and evaluated, with a simple marking scheme used to determine the most fruitful approaches taken to the issue performer-audio-visual mappings.

Following this, the Ashitaka instrument is discussed in detail, looking at the physical interface and design of the software in turn. The instrument was intended as a kind of ecosystem, and the various elements and interactions of this ecosystem are examined and

outlined. A thorough evaluation of the instrument comes next, followed by a chapter outlining various areas for possible future work.

Chapter 2

Background

2.1 Audiovisual Art

Providing a complete overview of the entire range of artworks which incorporate audio and visual elements is outwith the scope of this thesis. This section will, however, attempt to cover as wide a range as possible of audiovisual media, with a focus on those which use abstract visuals[1]

2.1.1 The Influence of Music on Painting

In the late nineteenth and early twentieth centuries, visual art practice (and in particular painting) began to borrow ideas from music. The rise of instrumental music in the nineteenth century

[1] As mentioned in the Introduction, a discussion of media like opera and dance is largely absent from this thesis, due to time constraints and my intention for the final Ashitaka instrument to use abstract visuals and motion as opposed to the more theatrical modes of these media.

had influenced a number of painters, who perceived the 'pure' abstraction of such music as an ideal to which painting should aspire. As such, they moved away from figurative, representational painting towards a style more concerned with the interplay of abstract form and colour.

A particularly significant figure in this movement towards music-inspired abstraction is Wassily Kandinsky. Having started painting relatively late in his life (at the age of 30), Kandinsky developed a theory of art based somewhat on the abstraction of instrumental music, as exemplified in the following quote:

> *"the various arts are drawing together. They are finding in music the best teacher. With few exceptions music has been for centuries the art which has devoted itself not to the reproduction of natural phenomena, but rather to the expression of the artist's soul, in musical sound."*[2]

This musical influence can be further seen in the way Kandinsky named many of his paintings as either *'Improvisations'* (referring to a painting which was conceived and realised over a short period of time) or *'Compositions'* (referring to a work produced over a greater period of time and with presumably a greater degree of attention to detail). It is also visible in his distinction between *'melodic'* and *'symphonic'* composition in painting:

> *"(1) Simple composition, which is regulated according to*

[2] p19, *'Concerning the Spiritual in Art'*[75].

an obvious and simple form. This kind of composition I call the melodic.

(2) Complex composition, consisting of various forms, subjected more or less completely to a principal form. Probably the principal form may be hard to grasp outwardly, and for that reason possessed of a strong inner value. This kind of composition I call the symphonic."[3]

Kandinsky's belief in a connection between music and painting went somewhat further than the simple borrowing of terms, however. Kandinsky perceived art as having a psychic effect on its audience - he saw colour and form as producing spiritual vibrations, and art as essentially being the expression of these fundamental spiritual vibrations. He perceived a common denominator behind all art, and as such it can be surmised that the translation of one artform into another is a distinct possibility in Kandinsky's philosophy.

There is some evidence that Kandinsky may have been synaesthetic[4]. This is an idea which is supported by the way he would often describe paintings in musical language, or music in visual language (for example, describing Wagner's *Lohengrin*; *"I saw all colours in my mind; they stood before my eyes. Wild, almost crazy lines were sketched in front of me."*[5]).

[3]p56, Ibid.

[4]See p149 and p151, *'Arnold Schoenberg Wassily Kandinsky: Letters, Pictures and Documents'*[99].

[5]p149, Ibid.

Perhaps the most interesting example of Kandinsky's belief in a common denominator in art is his abstract drama *'Der Gelbe Klang'*, written in 1909 (in addition to two other, unpublished, dramas; Schwarz-Weiβ and Grüner Klang[101]). Significantly, Kandinsky sought to link music, colour and dance, remarking in *'Concerning the Spiritual in Art'* that:

> *"The composition for the new theatre will consist of these three elements:*
>
> *(1) Musical movement*
>
> *(2) Pictorial movement*
>
> *(3) Physical movement*
>
> *and these three, properly combined, make up the spiritual movement, which is the working of the inner harmony. They will be interwoven in harmony and discord as are the two chief elements of painting, form and colour."*[6]

'Der Gelbe Klang' is also interesting as it highlights Kandinsky's relationship with Arnold Schoenberg, whose *'Die Glückliche Hand'* is in a number of ways quite similar to Kandinsky's drama. For example, in *'Analysing Musical Multimedia'*[56], Nicholas Cook notes that Schoenberg's coupling of colour to sound owes a lot to Kandinsky's own ideas on the subject[7]. Indeed, both artists seem

[6] p51, *'Concerning the Spiritual in Art'*[75].

[7] *"Schoenberg, like Kandinsky, couples the violin with green, deep woodwinds with violet, drums with vermilion, the lower brass instruments with light red, and the trumpet with yellow."*; p47, *'Analysing Musical Multimedia'*[56].

to have had some influence on the other, with Kandinsky initiating the relationship after he attended a concert of Schoenberg's music in 1911. The concert seems to have resonated strongly with Kandinsky, who also documented it shortly after with his painting *'Impression III (concert)'*[8]. Kandinsky saw Schoenberg's atonal compositions as representative of the kind of art he wanted to create himself.

At around the same time as Kandinsky was coming to prominence, the two artists Morgan Russell and Stanton Macdonald-Wright were developing their own theory of art based on musical analogy; *Synchromism*. This theory was somewhat derived from the theories of Ogden Rood, taught to them by their teacher in Paris, Percyval Tudor-Hart. Rood held that colours could be combined in harmony by progressing around the colour wheel in 120 degree steps, and that triadic 'chords' could be formed in this manner[77]. Tudor-Hart had developed a complicated mathematical system by which colour and musical notes could be compared to one another, and taught that it was therefore possible to create melodies of colour. While synchromism was somewhat simpler than Tudor-Hart's theories, it did retain a musical basis. Morgan Russell, quoted in *'Visual Music'*:

> *"In order to resolve the problem of a new pictorial structure, we have considered light as intimately related chro-*

[8] See p32, *'Visual Music: Synaesthesia in Art and Music Since 1900'*[51].

matic waves, and have submitted the harmonic connections between colors to a closer study. The "color rhythms" somehow infuse a painting with the notion of time: they create the illusion that the picture develops, like a piece of music, within a span of time, while the old painting existed strictly in space, its every expression grasped by the spectator simultaneously and at a glance. In this there is an innovation that I have systematically explored, believing it to be of a kind to elevate and intensify the expressive power of painting."[9]

Motion was an important part of synchromism. As noted in *'Morgan Russell'*[77], most of Russell's compositions rely on a central point, around which the rest of the painting revolves. For Russell, movement in a painting could then be produced in three different ways; *"circular and radiating lines, spiral movement (around a center), or a combination of two different directions"*[10].

This interest in motion led to the two Synchromists devising a machine that would incorporate time in a way not possible with a static painting; *The Kinetic Light Machine*. The machine was based around the idea of lights shone through 'slides' - paintings done on tissue paper and mounted on a wooden frame. Russell's conception of synchromism was heavily influenced by light (in *'Morgan Russell'*, Marilyn S. Kushner describes his paintings as being

[9] p43, Ibid.
[10] p88, *'Morgan Russell'*[77].

"abstractions of light"[11]), and this influence led him to experiment with actual light, as opposed to a representation of it. It appears that motion was to have been less important in the design of the instrument than the control of light, with most descriptions making it sound like a somewhat advanced form of slideshow. Though the two artists came up with a number of designs, they were not able to build the machine during the period they spent most time working together (1911-1913), and it was not until Russell visited Macdonald-Wright in California in 1931 that they were able to finally experiment together with a working example.

2.1.2 Abstract Film

In the early 1920s, artists began to see film as a way to incorporate motion directly into their art, rather than merely suggesting it as the synchromists did. The earliest artists to do so were Walter Ruttmann, Hans Richter and Viking Eggeling, all based in Berlin[12]. All three created primarily monochrome, abstract films, somewhat influenced by music's temporal nature[13]. Ruttmann's films feature a combination of geometric and more dynamic, amorphous shapes, moving in rhythmic patterns. Richter was also interested in rhythm, working primarily with rectangular shapes and often playing with the viewer's perception of perspective, as

[11] p105, Ibid.
[12] p.100, *'Visual Music: Synaesthesia in Art and Music Since 1900*[51].
[13] Evidenced to some degree in the titles of their pieces; see Richter's *'Rhythmus'* pieces, Eggeling's *'Symphonie Diagonale'*.

some squares seem to recede into the distance while others remain static. Eggeling's one surviving film, *Symphonie Diagonale*, differs quite greatly in style from the other two artists. Concentrating less on the motion of objects, it is primarily made up of otherwise static images, of which parts are removed or added to over time. The shapes themselves are also quite different from those of Ruttmann or Richter, with Eggeling often making use of lines (both curved and straight) far more than the polygonal shapes of the other two artists.

The films by Ruttman, Richter and Eggeling are sometimes termed 'Visual Music'. Perhaps the most significant artist in this area is Oskar Fischinger, originally inspired by Ruttmann's films to create abstract films of his own[90]. While both Richter's and Eggeling's films were primarily silent, Ruttmann's films were accompanied by music composed specifically for them, in a loose synchronisation with the visuals. Fischinger would take this audiovisual connection further with his *'Studies'*, in which existing pieces of music were tightly synchronised to monochrome animations made up of dancing white shapes on a black background. In my own opinion these films have yet to be bettered when it comes to such a tight connection between sound and visuals. The connection is such that the motion of the white shapes appears to correspond exactly to that of the accompanying music, as the shapes dance and swoop in tandem with it.

Fischinger amassed a large body of work throughout his life, at

one point working on Disney's *'Fantasia'*[59], before he quit as a result of the studio's artists insisting on making his designs more representational[14]. Particularly interesting with respect to this thesis are his experiments with the optical soundtrack used on early film stock. This stock included an optical audio track alongside the visual frames, meaning the soundtrack could be seen as an equivalent to the waveform view in audio editor software. Fischinger essentially subverted the original intention for this soundtrack (i.e. to hold *recorded* sound) by drawing geometric shapes directly onto it, which would be sonified as various *'pure'* tones. By doing this he made use of a kind of direct connection between sound and visuals. As he said himself:

> *"Between ornament and music persist direct connections, which means that Ornaments are Music. If you look at a strip of film from my experiments with synthetic sound, you will see along one edge a thin stripe of jagged ornamental patterns. These ornaments are drawn music – they are sound: when run through a projector, these graphic sounds broadcast tones of a hitherto unheard of purity, and thus, quite obviously, fantastic possibilities open up for the composition of music in the future."*[15]

Two other significant figures in this field are the brothers John and James Whitney. Working together in the early 1940s, the

[14] p.178-179, *'Analysing Musical Multimedia'*[56].
[15] *'Sounding Ornaments'*[65].

brothers created *'Five Film Exercises'*, five films which exhibited a very tight connection between sound and visuals. As Kerry Brougher notes in *'Visual Music'*[51];

> *"The images and sound seem inextricably linked. One is not a result of the other; rather, sound is image, and image sound, with no fundamental difference."*[16]

To create these films, John had to construct much of their equipment himself, building first an optical printer to create the visuals, then a complicated system of pendulums to create the sound. This system worked by having the pendulums attached to the aperture of a camera, so that when set in motion, the pendulums would be captured on to the film's soundtrack, similar to Fischinger's *'Sounding Ornaments'*. The brothers then synchronised this optical sound track to visual frames created by light shone through stencils[51].

Following this, both brothers continued to make films, with James creating films such as *'Yantra'* and *'Lapis'*, and John producing films such as *'Permutations'* and *'Arabesque'*. John also developed a theory, which underpins all his later works, termed *Digital Harmony*. Outlined in his book *'Digital Harmony: On the Complementarity of Music and Visual Art'*[111], the theory is based on the idea that patterned motion is the link between music and image. John felt that the primary motion present in music came

[16] p125, *'Visual Music: Synaesthesia in Art and Music Since 1900*[51].

from harmony, and consonance and dissonance. He also felt that consonance and dissonance were present in specific visual motion. His later films primarily consisted of multiple points of light, moving in specific patterns. At specific times in the films, these points would come together to highlight certain visual patterns (consonance), before moving away and obscuring such obvious patterns (dissonance). In this way, John sought to link visuals with music via harmony.

2.1.3 Hollywood

As evidenced by Fischinger working with Disney, some of the 'Visual Music'-type ideas discussed in the previous section did find their way into more mainstream films. *'Fantasia'*[59] is still probably the best known example of visual music in mainstream film. Prior to its creation, Hollywood had for some time been aware of the experimental films being made by the likes of Eggeling and Fischinger[17], and receptive to at least some of their ideas. Kerry Brougher, for example, notes that;

> *"Disney showed an interest in surrealism and abstraction in the Silly Symphonies of the late 1920s and early 30s, as did Busby Berkeley in his choreography for films such as Dames(1934) and Gold Diggers of 1935(1935)."*[18]

[17] p105, *'Visual Music: Synaesthesia in Art and Music Since 1900'*[51].
[18] p105, Ibid.

Films such as *'The Skeleton Dance'* featured visuals that were closely linked to the accompanying music, and Fantasia was intended as an extension of this idea, as well as a way of bringing high art (e.g. Stravinsky's *'The Rite of Spring'*) to the masses[19]. Indeed, though Fischinger essentially disowned it, the film retained a number of his ideas, and abstract elements are present throughout much of the film, in tension with the more representational elements. In his analysis of The Rite of Spring sequence in Fantasia, Nicholas Cook notes:

> *"in general the various sequences in 'Fantasia' are either abstract or representational (in the cartoon sense, of course), and there seems to have been little attempt to impose an overall visual unity upon the film as a whole; different teams worked on each sequence, and there was little overlap in the personnel. In the case of the Rite of Spring sequence, however, there were apparently separate directors for the various sections; and possibly as a result of this, it includes both abstract and representational animation, together with much that lies in between these extremes."*[20]

The other major example of visual music ideas making their way into mainstream film is Stanley Kubrick's *'2001: A Space Odyssey'*. The stargate sequence in the film was created using a technique originally devised by John Whitney known as slit-scan

[19]p.174, *'Analysing Musical Multimedia'*[56].
[20]p179, Ibid.

photography[21]. This technique involves an opaque slide with a slit cut into it being placed between the camera and the image to be photographed. By moving the camera in relation to the slit during the capture of a single frame, only particular sections of the image are captured and, due to the motion of the camera, are elongated along the direction of motion.

2.1.4 Music Video

Before discussing music video in particular, it is perhaps necessary to take in a wider view of popular music's visual side. As John Mundy argues in *'Popular Music on Screen'*[92], popular music has always had a strong visual element, whether it is the musician's performance when playing live, or the more complex array of multimedia elements surrounding an artist like Madonna. Recorded sound has perhaps made it possible to conceive of music without some kind of associated imagery, but music in general is still at the very least distributed accompanied by a record/CD cover[22].

Music as an artform has its origins in performance, and as such popular music has always - with some exceptions[23] - been oriented around performance. It can be further claimed that musical performance in general has a strong visual dimension, even if it is for

[21] p151, *'Expanded Cinema'*[113].

[22] I would note that even if the internet eventually makes hard copies of music obsolete, sites like iTunes accompany their lists of downloads with the music's associated cover art, and would seem likely to do so for the foreseeable future.

[23] For example electronic dance music since the 1980s.

the most part secondary to the sound[24]. These ties between music and performance are also present in the vast majority of music videos, where - whatever else the video depicts - the musician(s) will usually be depicted performing the song at certain points, if not necessarily throughout the whole video.

In Britain, *Top of the Pops* played a significant part in the birth of the music video. Up till that point (1964), music television had primarily relied on filming live performances by the artists featured. In contrast to the other music shows broadcast at the time, *Top of the Pops* had a format which required the producers to feature the artists currently doing well in the charts, and always had to end with the current number one single[25]. This requirement meant that often the program would need to feature artists who were not able to show up at the studio due to prior engagements, and when this happened, the show's producers would either improvise with news footage of the artists (e.g. the Beatles' *'I Want to Hold Your Hand'* on the very first show[26]), or would use footage of the act shot while they were on tour.

The Beatles themselves also played a significant part in the development of the music video. From *'Popular Music on Screen'*:

[24]A fairly recent exception to this rule is live laptop-based music. This type of performance is, however, often criticised for lacking any significant visual stimuli, and to make up for the absence of expressive human performers, such a performance is often accompanied by computer-generated visuals.

[25]One of the aims of *Top of the Pops* was to reflect the public's record-buying habits, rather than deliberately influence them, as sometimes happened with the music shows on ITV and in America.

[26]See p.206, *'Popular Music on Screen'*[92].

"Following their experience filming for the BBC documentary The Mersey Sound, directed by Don Haworth in 1963, the success of A Hard Day's Night *and* Help!*, and the taping of* The Music of Lennon and McCartney *for Granada in November 1965, the group decided to produce and videotape a series of promotional video clips, with the effect, as Lewisholm puts it, of 'heralding the dawn of pop's promo-video age'."*[27]

The Beatles' promos were significant in that they both represented a break with the previous traditions of music television (which was still at that point mainly concerned with featuring artists performing), and also reflected a growing understanding of the international market for pop music, and the power of moving imagery to define the pop 'product', and market it successfully.

Carol Vernallis presents a thorough and wide-ranging analysis of music video in the book *'Experiencing Music Video: Aesthetics and Cultural Context'*[107]. Vernallis identifies the three key elements of music video as being music, lyrics, and image. Throughout a typical music video, each of these three elements will at different points gain prominence, come together in moments of congruence, and come into conflict, actively contradicting each other. Significantly, though the alleged intent behind music video is to showcase the song, no single element is allowed to gain the upper

[27] p.207, Ibid.

hand. Directors work to create a kind of equilibrium between the three elements, where each element may be more or less prominent at separate points, but which are never allowed to take over the entire video.

Each of the three elements can also be seen to influence the others. As Vernallis writes:

> *"In a music video, neither lyrics, music, nor image can claim a first origin. It can be unclear whether the music influences the image or the lyrics influence the music. For a form in which narrative and dramatic conflicts are hard to come by, some of the excitement stems from the way each medium influences the others."*[28]

In addition, each medium may carry a different sense of temporality, a different representation of the passing of time, and the combination of the three can make for a distinctly uncertain, ambiguous temporal context. It also further demonstrates the influence between media, with the visuals in music video often serving the purpose of emphasizing and exposing the structure of the music, a higher level feature of a medium that is generally experienced in a visceral, immediate fashion.

Of course, music video is to a large extent defined by the music it is created to showcase; pop music. This clearly imposes a number of conditions and constraints upon the directors of music

[28] p.79, *'Experiencing Music Video: Aesthetics and Cultural Context'*[107].

videos. For example, Vernallis notes that music videos are very rarely traditionally narrative, as pop music tends to be more concerned with *"a consideration of topic rather than an enactment of it"*[29]. Pop songs tend to have a cyclic, repetitive structure which does not lend itself well to straightforward storytelling, and creating such a narrative for the visual domain would result in the music being reduced to the role of background sound (a role that may suit cinematic purposes, but clearly works against the intention to showcase the song). One of the ways directors deal with pop music's repetitive nature is to make use of imagery which is slowly revealed upon each (musical) repetition. By doing this they link certain imagery to a particular musical phrase or hook, and are able to create a larger meaning from this gradual revelation. Video directors often also match musical repetition to simple visual repetition, something which both links the visuals to the music, and imparts a sense of cohesion to the visuals (which will tend to consist of a series of seemingly disconnected tableau). Directors will often choose to work with a specific, limited selection of visual materials for this reason.

Music video is primarily concerned with the immediate, with embodying the surface of the song. It is *"processual and transitory"*, as Vernallis puts it:

> *"Both music-video image and sound - unlike objects seen in everyday life - tend to be processual and transitory*

[29] p.3, Ibid.

rather than static, and to project permeable and indefinite rather than clearly defined boundaries. The music-video image, like sound, foregrounds the experience of movement and of passing time. It attempts to pull us in with an address to the body, with a flooding of the senses, thus eliciting a sense of experience as internally felt rather than externally understood."[30]

This appeal to visceral experience can be seen in the way colour and texture are used in music videos to elicit an immediate response, and draw attention to particular aspects of the song. Textures are often linked to particular musical timbres, drawing our attention to certain instruments (a distorted guitar might be linked to a jagged, rough texture, for instance). They also evoke particular tactile sensations based on our experience of the physical world. The physical, immediate response which music video aims for can be further seen in the importance of dance to the artform. Vernallis recounts a telling anecdote[31] of viewers only beginning to appreciate Missy Elliot's music once they had seen how she moved in her videos. Music videos can provide a demonstration of how to navigate a song's landscape, how it relates to the body, and how the viewer should move in response.

The other significant element of music video which foregrounds the immediate, present-tense nature of the medium is editing. Un-

[30] p.177, Ibid.
[31] p.71, Ibid.

like cinematic editing, which aims to preserve the flow and continuity of the image, music video editing is frequently disjunctive and disorientating. It is meant to be noticed, in contrast with the editing most common to cinema, which aims to be all but invisible. One of its key purposes is to keep the viewer in the present by refusing to allow them time to reflect on what they have seen. It also serves to maintain the equilibrium between the three elements of music, lyrics and image (as discussed earlier) by never allowing a single element to dominate the video.

The key factor with music video is that it is always in service of the song. All the effort that goes into maintaining the equilibrium between the three elements is for the purpose of keeping the viewer's attention, to get them interested in buying the associated single or album. As such, while it is arguably the most widespread (non-narrative) multimedia artform, music video is always based upon a one-way relationship between its media. Directors will often prefigure certain musical events visually, and there can be no doubt that the combination of visuals and music can change the viewer's perception of the music, but ultimately, the song always comes first.

2.1.5 VJing

VJing is a term referring to the practice of VJs, or "video jockeys". One of the earliest uses of the term VJ was to describe the

presenters on MTV, as they essentially presented music videos in much the same way as a radio DJ. More recently, however, VJing has become a separate performance practice more akin to musical performance than the simple introduction of a series of videos. VJing today is primarily concerned with the live performance and manipulation of visuals to accompany music. VJing in this sense of the term was born out of clubs playing electronic dance music, where there seems to have been a desire to provide some kind of visual stimulus more engaging than the relatively static, relatively anonymous DJ in charge of the music. More recently, it has also become common for more traditional bands and musicians to collaborate with VJs when playing live. In terms of visual content, VJing encompasses a wide range of styles, from abstract to representative and narrative forms. The methods of displaying a VJ's work also vary. Originally, VJs worked almost exclusively with standard projectors and aspect ratios (i.e. the 4:3 TV ratio), though many VJs now work with arbitrary screen shapes and sizes. In the article *"Visual Wallpaper"*[48], David Bernard argues that VJs more concerned with narrative tend to use more standardised, cinemaesque screen ratios, while the more abstract Vjs may use a higher number of more idiosyncratic screens, being more concerned with the general visual environment than a single visual focus point.

As an outsider, establishing a history of VJing is extremely hard, as there appears to be very little in the way of writing which tries to follow the various developments of the artform since its origins.

What written 'history' there is tends to focus on technical developments with regards the software and hardware developed for VJs, with very little said about the artistic developments which must have taken place simultaneously. *'The VJ Book'*[104] by Paul Spinrad gives a brief overview, placing the movement's origins in New York's Peppermint Lounge in the late 1970s. With the birth of MTV, however, many clubs settled for simply displaying the music channel on TVs, with no interest in live visual performance. It was not until house music took hold in Britain in the late 1980s, giving birth to rave culture, that VJing started to build momentum as a discipline in its own right. Pre-eminent among the VJs of the 1990s are the DJ and VJ duo Coldcut, who appear to have been influential to the movement, not only with their own performances but also as a result of the VJing software they helped develop; VJamm[26].

While VJing as a performance practice is concerned with much the same issues as this thesis, there appears to be a significant lack of critical thought applied to the artform[32], making it hard to gain much insight into the processes involved. The book *"VJ: audio-visual art and vj culture"*[62] primarily consists of interviews with a wide range of VJs, and there are some themes that re-appear across a number of interviews. For instance, the Japanese artist *united design* notes that:

"If sound and image are mixed well, each factor becomes

[32]The website `http://www.vjtheory.net/` appears to be dedicated to filling this void with the publication of a book on the subject, though it is relatively sparsely populated at the time of writing.

more impressive than it would be on its own; in live performances, they create new forms of expression unexpectedly."[33]

The unexpectedness described suggests that (at least some) VJs are somewhat in the dark when it comes to how visuals and music connect, relying on their intuition rather than a more formalised theory of how VJing 'works' (though it could be argued that the preceding quote may just describe the process of improvisation). Another theme that comes across in a number of the interviews is related to how the interviewees link their visuals to the music they're accompanying, with a number of VJs displaying pride at how closely their visuals sync (temporally) with the music. From the (admittedly rather short) interviews, the reader is left with the impression that this temporal synchrony is the most important aspect of what these VJs do, with visual form and colour appearing to take a distant second place.

Finally, though it appears in an article primarily concerned with music videos for dance music, as opposed to VJing, music critic Simon Reynolds makes an interesting point in the article *"Seeing the Beat: Retinal Intensities in Techno and Electronic Dance Music Videos"*[97]:

"There's a good reason why all clubs take place in the dark, why many warehouse raves are almost pitch black.

[33] p.135, [62].

It's the same reason you turn the lights low when listening to your favorite album, or wear headphones and shut your eyelids. Diminishing or canceling one sense makes the others more vivid. Dance culture takes this simple fact and uses it to overturn the hierarchical ranking of the senses that places sight at the summit. Visual perception is eclipsed by the audio-tactile, a vibrational continuum in which sound is so massively amplified it's visceral, at once an assault and a caress. The body becomes an ear."

...Which, while ignoring the culture of VJing at some such clubs, makes a case that VJing in these clubs is both an unnecessary visual distraction, and a contradiction of the music(the culture)'s privileging of the aural over the visual. It also suggests an explanation for the movement of VJs towards the accompaniment of live musical performance over dance clubs, as such a performance situation has always had an important visual component.

2.1.6 Computer Art

The term 'Computer Art' is intended as a kind of catch-all to encompass audiovisual artforms which depend on the use and manipulation of computers. Naturally with such a broad topic definition there are bound to be omissions, and as such this section will focus on a few select areas particularly relevant to this thesis' topic. The areas covered are the demoscene, music visualisation

software, and computer games.

Starting with the demoscene subculture - one of the original inspirations for this project - the wikipedia definition is:

> *"The demoscene is a computer art subculture that specializes itself on producing demos, non-interactive audio-visual presentations, which are run real-time on a computer. The main goal of a demo is to show off better programming, artistic and musical skills than other demogroups."*[32]

The demoscene has its roots in the software piracy of the 1980s[70], and is based in particular on the exploits of the programmers - known as crackers - who would work to defeat the copy protection implemented in commercial software so it could be freely copied, and distributed to a wide network of people unwilling to pay for the software. Initially the 'cracked' software would be distributed with a simple text screen with the name of the programmer who cracked it, displayed on startup. Eventually, however, crackers started to display their programming prowess with more complicated intro screens with animation and sound. With the limited amount of disk space available, and the limited computing power of the machines used, such intros required the programmer to push at the limits of what was possible on the system, and an in-depth understanding of the computers used was essential. As increasing numbers of these more complex intros were released, different cracking groups started to actively compete with each other, and

attempt to squeeze the most out of the limited resources at their disposal. Eventually, the programmers responsible for these intros broke away from the illegal cracking scene, and started releasing their productions independently. These productions became known as demos (short for demonstrations), which led to the term demoscene being used to describe the subculture in general.

As alluded to in the previous paragraph, demos are generally the product of demo groups. A group will typically consist of (at least) a graphic artist, a composer, and a programmer[106]. Those involved in the demoscene tend to meet up at large events known as demoparties, where there are competitions for different types of demos, and which provide a chance to meet other people involved in the scene. The competitions maintain the competitive element of the artform, with the party-goers voting for their favourite productions, and some kind of prize awarded to the winners. As a subculture, the demoscene is relatively closed off, in that the intended audience for a demo is other members of the demoscene. As such, the scene has developed its own aesthetic rules, and works which appeal to a wider audience while disregarding those rules tend to be looked down upon by those within the scene.

Before the (originally IBM) PC became the dominant general purpose computer worldwide, demos were produced for systems with fixed, standardised hardware (such as the Commodore 64, Atari ST, Commodore Amiga etc.). As a result, programmers all essentially started from the same position, and the demoscene's com-

petitive aspect was focused on how much performance they could squeeze out of their machines, with specific records being set and subsequently broken (for example, how many moving sprites could be displayed on screen at any one time[68]). As the PC gained marketshare, however, its open-ended, expandable hardware (where the owner can easily upgrade their graphics and sound cards, etc.) meant that demo makers found themselves producing demos for an audience with widely-varying hardware. In order to produce the most impressive graphical effects, some programmers would tend to buy expensive, high-end graphics cards, with the result that their productions would only run on this high-end hardware. As such, recent demos tend to be distributed as videos, in addition to the executable (which will usually not run on an average PC). This shift has meant that such demos tend to be judged now more for their artistic merit (still according to the demoscene's particular aesthetic rules) than their technical achievements. It has also resulted in the establishment of particular types of demos (termed 'intros' within the scene), which, for example, seek to squeeze the most content into a 64kB executable (you can also find 4kB, and even 256-byte intros), and which therefore retain the aspect of technical competition. A relatively famous example of this kind of programming prowess is *'.kkreiger'*[105], an entire FPS (First Person Shooter) game contained within an executable of 96kB, quite a feat when compared to similar games which may require entire DVDs full of data.

There are perhaps two main criticisms to be made of the demoscene. Firstly, the emphasis on technical prowess could be seen as being to the detriment of any artistic merit. Through their focus on displaying the most flashy, spectacular graphics, it could be argued that demos are the equivalent of big-budget, hollywood blockbusters; all style and no substance. Secondly, though demos are an audiovisual medium, the connection between sound and visuals rarely appears to have had much thought applied to it. For the most part, demos tend to display fairly simplistic audiovisual connections, with, for example, the camera jumping to a new position on every beat, or basic audio amplitude to visual parameter mappings. And while visually, demos encompass a wide range of styles and aesthetics, the music used tends to be almost exclusively nondescript dance music.

Along a similar vein, a brief discussion of music visualization is perhaps necessary. Such software is generally included with media player software such as iTunes[10] and Winamp[41], or as addons or plugins for such software. For the most part, music visualizers rely on basic parameters derived from the played music[34] to modulate parameters of a visual synthesis engine. As such, the audiovisual connection is very basic, and moments when the visuals appear to actually 'connect' with the audio tend to appear at random, and relatively infrequently.

[34]Parameters such as amplitude envelope, frequency content, possibly beat detection.

There appear to be very few references available with respect to the history of music visualizers. There are clear links back to the history of colour organs, and rock concert lightshows, but most visualizers appear to have been created for primarily commercial rather than artistic purposes. At the time of writing, the wikipedia article[34] cites Cthugha[3] as the first visualizer, but Jeff Minter's VLM-0[85] (Virtual Light Machine) predates it by several years, making it one of the earliest music visualizers developed. Significant recent visualizers include Jeff Minter's Neon[16], included with Microsoft's XBox 360 games console, and the Magnetosphere[14] iTunes visualizer from the Barbarian Group, originally derived from a Processing[20] sketch. It is telling, however, that while these visualizers can be judged to have improved upon those that came before in terms of graphical expression, they still rely on the same means of connecting visuals to sound as the earliest visualizers. This is perhaps a problem inherent to any attempts to match visuals to sound automatically.

Being such a widespread part of modern culture, some mention must also be made here of computer games. Arguably the two most popular game genres today are the First Person Shooter (commonly abbreviated FPS) and the Role Playing Game (RPG)[35]. These two genres both share some clear influences from other media, such as

[35]Though figures on what are the most popular genres are hard to come by. It could also be argued that so-called 'casual games' represent the most popular genre today.

architecture (visible in the attention paid to level or world design) and cinema (the use of cutscenes to advance a story, the use of lighting to enhance mood...), and of course these games can be said to be the most prevalent form of virtual reality in existence today. The music in such games rarely seems to be intended to do much more than signify a particular mood or atmosphere, but the use of sound effects is perhaps more interesting. Relying on the same processes used by foley artists in cinema, sound effects in these kind of games seem intended to immerse the player in the game world, and heighten their perception of it. This is done by linking visual events (specifically impacts) with exaggerated impact sounds, often with a substantial bass frequency component. This then has the effect of making the impacts appear more visceral and physical (particularly when the player has the volume turned up), thus arguably making the experience more 'real' for the player. The paper *'Situating Gaming as a Sonic Experience: The acoustic ecology of First-Person Shooters'*[69] goes into some detail on the function of sound in FPS games, treating their sonic component as a form of acoustic ecology.

While the use of sound in FPS and RPG games seems to be largely (if not entirely, see [69]) derived from the way sound is used in cinema, other genres can be seen to make use of an audiovisual connection that is specific to computer games. Take dance games, for instance, of which Dance Dance Revolution[31] is perhaps the most prominent. Played with a special sensor-equipped mat, these

games present the player with a scrolling list of steps which the player then has to match, in time with the music. In this case the audiovisual connection is essentially mediated by the player, as it is their responsibility to match the visual dance steps to the audio beat. In a similar vein, the Guitar Hero[33] and Rock Band games present the player with plastic guitar/drum interfaces on which they are required to play along to particular songs, again following visual cues.

A perhaps more sophisticated approach to audiovisual relations can be found in the game Rez[35]. Released for the Sega Dreamcast and Sony Playstation 2 in 2001, Rez is essentially a rails shooter[36] with the sound effects removed. In their place is a sparse musical backing track, to which additional notes and drum hits are added whenever the player shoots down enemies. In addition, the game uses largely abstract visuals, making the game more a play of colour and sound than more traditional, representation-based games.

More interesting still is Jeff Minter's Space Giraffe[37], recently released (at the time of writing) for the XBox Live Arcade. Essentially a shooter derived from the Tempest arcade game[38] of the early 1980s (though it should be noted the manual emphatically states that *"Space Giraffe is not Tempest"*), Space Giraffe derives its visuals from Minter's Neon music visualizer. The use of a mu-

[36]A rails shooter is a game where the player has no control over their movement, instead being solely responsible for shooting down enemies.

sic visualizer engine rather than more traditional computer game graphics means the player is constantly bombarded with a huge amount of visual information, and their enemies are frequently entirely obscured by the hyperactive visuals. What makes Space Giraffe so interesting is that it is apparently still playable, as players talk about perceiving the game environment subconsciously[37]. The extreme amount of visual and sonic information hurled at the player means that it is all but impossible to play by following the usual gameplaying process of consciously looking for enemies, targetting them and pressing the fire button. Instead, as players learn the game, they start to react unconsciously to the visual and aural cues bombarding their senses. All enemies have a consistent aural and visual signature (including patterns of movement etc.), something which plays a significant part in aiding the player to navigate the environment. They may not be consciously aware of the position (or possibly even existence) of a particular enemy, but skilled players will still destroy it, relying on the audio and visual cues which have filtered through to their subconscious. As such, Space Giraffe is a game which is as much about altering the player's perception (*'getting in the zone'*) as it is about getting the highscore.

In recent years there has been a growing acceptance of the idea of computer games representing an emerging artform[93]. Presumably the result of artists having grown up immersed in such games,

[37]See [49] where the game is compared to *Ulysses*, and Stuart Campbell's review at [53].

there are an increasing number of websites focusing on so-called 'art games'[38]. For the most part, this fledgling movement is united by the belief that what can elevate games to an artform, above simple entertainment, is their interactive nature - their gameplay. This is perhaps an important point to note because criticisms of the idea that games are art often compare games to artforms involving a far more passive audience, such as cinema[61]. The focus on interaction also suggests an interesting parallel to this project's goal of an audiovisual instrument (games being inherently audiovisual), and it seems likely that in the near future there will be some very interesting developments in this area.

2.2 Musical/Audiovisual Instrument Design

This section focuses on the design of instruments which make use of computer software to either generate sound or visuals, or manipulate input signals. The reason for this focus is that this project will be primarily software-based, and such instruments face specific challenges not generally found in traditional, acoustic instruments.

[38] See, for example, http://www.selectparks.net/, http://www.northcountrynotes.org/jason-rohrer/arthouseGames/, http://braid-game.com/news/.

2.2.1 Digital Musical Instruments

The term *'Digital Musical Instrument'* (abbreviated DMI) refers to a musical instrument which consists of two fundamental (and separate) parts: an input device, and some kind of sound generator[109]. In a traditional instrument such as a violin, the interface with which the performer plays the instrument is also the means of sound generation - you play it by bowing a string which vibrates in response, and produces sound. By contrast, a DMI has a far more opaque connection between interface and sound generation, as any sound generated will be generated by algorithms running in software - the sound generator does not exist as a physical object in the same way that a violin string does. As such, in designing a DMI, the designer will typically have to create some kind of physical interface for the instrument, create some kind of sound generation algorithm, and then specify a set of mappings which will determine how the two interact with each other.

2.2.2 Mappings

The design of the mappings between the interface and the sound or visual generator in an instrument is a key factor in determining the instrument's playability - how easy it is for a performer to articulate complex gestures, and how much fun it is to play. The most basic mapping strategy which may be used is a simple one-to-one strategy, where one input signal is mapped to one

synthesis parameter. This is the approach taken, for example, by numerous commercial audio synthesizers, from both the original hardware devices to their more recent software derivatives. Research conducted by Andy Hunt and Marcelo Wanderley, however, suggests that one-to-one mappings are not the optimal strategy to take when designing musical instruments. In their paper *'Mapping Performer Parameters to Synthesis Engines'*[72], they describe their experiments regarding the different ways an input device may be mapped to a sound generator. For one of these experiments, they devised three interfaces and mapping strategies to be used; a mouse controlling 'virtual' sliders displayed on screen, with each slider mapped directly to a single synthesizer parameter; a hardware fader box controlling virtual sliders displayed on screen, again with a one-to-one mapping strategy; and a combination of mouse and hardware fader box with no visual feedback and a more complex mapping strategy, where multiple input signals were mapped to a single synthesizer parameter, and a single input signal may manipulate multiple synthesizer parameters. The participants in the experiment were asked to listen to a sonic phrase, then copy it as best they could using the interface they were assigned to. As well as testing how accurately the participants could copy phrases with these interfaces, the experiment was also designed to examine the learning process, and determine how quickly the participants could become proficient with the various interfaces. The results of the experiment showed that the third interface and mapping strat-

egy came out far ahead of the other two. Quoting from the paper:

- *"The test scores were much higher than the other two interfaces, for all but the simplest tests.*
- *There was a good improvement over time across all test complexities.*
- *The scores got better for more complex tests!"*[39]

The authors propose that the reason for these results is that the one-to-one mappings require a relatively high cognitive load - the performer has to consciously think about what parameter they want to manipulate, and these parameters remain separate. With the more complex mapping strategy, however, (aural) parameters are linked and, while the learning curve for such an interface is somewhat steeper, having learned it, performers find it easier to perform more complex tasks because of these connections. They also note that most participants found the third interface to be the most enjoyable to use.

In the same paper, Hunt and Wanderley also map out two distinct approaches visible in the existing mappings literature, i.e. the use of explicitly-defined mappings, and the use of generative mechanisms such as neural networks, for setting up and adapting mappings based on the performer's input. With the neural network-based approach, the aim is to try and match the sound output to the performer's gestures - it could be considered as an attempt to

[39] p.7, *'Mapping performer parameters to synthesis engines'*[72].

match the instrument's internal model to the performer's own idea of how it works. This approach, however, would seem to short-circuit the traditional approach of learning an instrument, where the performer gradually discovers how the instrument works, and brings their mental model of its inner workings closer to the truth. It would also appear to allow for less exploration of the instrument, and less chance of the discovery of unexpected sounds and timbres, since new gestures on the part of the performer would perhaps be simply treated as new data to be adjusted for. The use of a neural network for adjusting mappings seems to imply that there is a 'correct' sound for the instrument (if not necessarily a correct way of articulating it), and that the performer should be aided, or guided towards acheiving it. A neural network is also essentially a black box - by way of contrast, explicitly-defined mappings have the advantage that the instrument designer will have some kind of understanding of which mappings work, and how the instrument fits together.

In *'Mapping transparency through metaphor: towards more expressive musical instruments'*[63], Fels et al. set out one of the more advanced (explicitly-defined) mapping strategies. Their approach derives from their observation that musical expression is understood with reference to an existing literature, or general body of knowledge within a particular culture. This literature serves the purpose of informing both the performer and the audience of how

the performer's gestures are connected to the resultant sound of their instrument. The authors argue that:

> *"Both player and listener understand device mappings of common acoustic instruments, such as the violin. This understanding allows both participants to make a clear cognitive link between the player's control effort and the sound produced, facilitating the expressivity of the performance."*[40]

They go on to argue that the expressivity of an instrument is dependent upon the transparency of the mappings used. This transparency means slightly different things to the performer and the audience. For the performer, transparency is related to their own gestures - for mappings to be considered transparent, the performer has to have some idea of how their input to the instrument will affect the sound, and they need to possess a good degree of proficiency or dexterity with the physical controls. For the audience, however, physical proficiency is not necessary. Rather, they have to have an idea of how the instrument works, something which is based primarily on cultural knowledge (the literature). Without this understanding, they can have no idea of the performer's level of proficiency with the instrument, or what the performer is trying to communicate.

[40] p.110, *'Mapping transparency through metaphor: towards more expressive musical instruments'*[63].

Based on these ideas, the authors then designed a number of instruments whose mappings are derived from specific metaphors. The use of metaphors allowed them to situate the instruments within the existing literature and thus avoid some of the problems associated with DMIs which use entirely new control paradigms and audio algorithms. One such instrument was *'Sound Sculpting'*. This instrument was based on a kind of 'sculptural' control paradigm, where the performer plays the instrument using the same gestures a sculptor might use when working with clay, for example. The instrument used a pair of data gloves as the input interface, controlling an FM sound synthesizer. It also had a graphical output, visualising the virtual block of 'clay' that the performer was manipulating. In their evaluation of the instrument, the authors note that while certain parameters of the instrument were instantly understood (such as position and orientation), others were far more opaque and confusing for performers (the mapping of the virtual object's shape to timbre). The use of a metaphor that is based so much on tactile, physical interaction also meant that the instrument suffered from having no haptic feedback - the performer's only indication of resistance to their actions came from the visual feedback.

A second instrument designed by the article's authors was *'MetaMuse'*, an instrument which aimed to find a control metaphor to manipulate a granular synthesizer. In this case, the authors used a model watering can and a flat palette as the control interface,

with the metaphor being that of falling water (from the can's spout, onto the palette). The performer played the instrument by tipping the watering can and varying the slope of the palette. The instrument's internal model created a virtual stream of water (the flow of which was determined by the can's tilt), and when a virtual drop of water was determined to have hit the palette below, an audio grain was generated. By tilting the palette, and varying the height of the can from the palette, the sound could be further manipulated, as might be expected from such a metaphor. This instrument clearly embodies a very potent metaphor - it is quite obvious from our shared cultural literature how to play it, and how the performer's gestures connect to the sound it creates. The authors argue, however, that it also demonstrates the problems that arise from the use of metaphor in instrument mappings. In evaluating the instrument, and gauging the responses of various different performers, they found that while the metaphor helped the performers to understand the instrument to a degree, the fact that it didn't always react exactly like a real watering can actually restricted this understanding, and made the mapping seem more opaque than a system designed without the use of metaphor.

A different approach is discussed by Arfib et al. in the article *'Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces'*[46]. In this article, the authors base their mapping strategy upon the use of perceptual

spaces. One of their aims was to create a modular strategy, whereby different input devices and audio synthesizers could be quickly connected together without having to re-design the entire instrument from scratch every time. As such, they outline a three layer strategy, where the low level sensor data describing the performer's gesture is first mapped to a higher level (gestural) perceptual space, which is then mapped to a high level aural perceptual space, which is finally mapped to low level audio synthesis parameters. Some examples given by the authors of high level gestural perceptual parameters are:

> *"the variation of the localisation, the more or less great variability around a mean curve, the relative or absolute amplitude, the combination (correlation) of evolution of several parameters, as far as a physical gesture is concerned."*[41]

Their discussion of perceptual audio parameters includes psycho-acoustic terms such as loudness and brightness (roughly-speaking, the perceived high frequency content of the sound), as well as what they term *physical parameters*, *signal parameters*, and *meta parameters* [42].

[41] p.132, *'Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces'*[46].

[42] They define physical parameters as those sonic characteristics which are related to the physical construction of a sounding object - in this case they primarily relate to physically modelled sound. Signal parameters are related to psycho-acoustic parameters, but whereas psycho-acoustic parameters are defined from the perspective of a listening human, signal parameters are defined in more mathematical terms (i.e. fundamental frequency as opposed to pitch, root mean square amplitude as opposed to loudness). Meta parameters refer to the control of multiple parameters simultaneously.

The article outlines a number of instruments the authors developed according to these principles, variously using imitative, formant and scanned synthesis. Of these, the Voicer (formant synthesis) instrument is perhaps the most notable. Controlled by a Wacom stylus and a joystick, the authors categorised the timbre of the synthesized sound according to four perceptual attributes; openness, acuteness, laxness and smallness. They linked acuteness and openness to vowel shapes in the synthesis algorithm, then mapped the two attributes to the two axes of the joystick. The stylus was used to control pitch, with clockwise motion raising the pitch, and the tablet divided into twelve sectors for each semitone in the octave.

This focus on perception as the basis for a system of mappings would appear to be extremely useful, particularly as a way to aid the performer in learning the instrument (as such, it attempts to solve much the same problem as the metaphor-based approach). By defining an instrument around it's *perceptual* audio and control parameters, there would appear to be greater potential for the performer (and audience) to quickly grasp how their gestures articulate the sound.

2.2.3 Visual Instruments

For this section, 'Visual Instruments' refers to tools which enable real-time performance with visual materials, and though not a

term in widespread usage, is intended to represent visual counterparts to musical instruments. Compared with the field of musical instruments, the number of visual instruments developed over the years is very small. Certainly there is nothing that can compare with the long tradition associated with an acoustic instrument such as a violin, and in terms of works of art, there is little in the way of an existing *performance-based*, visual music tradition. There are certain threads and significant instruments to pick out in this area however.

One particular tradition which overlaps into this area is that of colour organs. The colour organ tradition is based on the idea that colour can be matched to specific musical pitches, and that, as such, it is possible to make a visual, colour-based music. As such, it is not necessarily an entirely visual tradition, though there have been instrument designers who worked only in the visual domain. Alexander Wallace Rimington is one such designer. Rimington, like most other colour organ designers, built an instrument based on a piano-style keyboard, whereby, when a key was pressed, a coloured light would be projected over the performer's head onto a translucent screen mounted behind them (with the audience seated on the other side of the screen). As such, the performer was able to perform with dynamic, diffuse patterns of colour. While it was based on a piano keyboard, though, Rimington did not accompany his colour performances with music. Instead he saw his performances as a new artform solely based on the ma-

nipulation of colour through time. In his book[98], he notes that audiences tended to find it hard to distinguish between the often subtle and rapid changes of colour in a performance. Having performed with the instrument for a number of years however, he had himself developed a more discriminatory eye, and one of his hopes was that such instruments could be used as educational tools to heighten the perception of colour among the population.

In a similar vein, Thomas Wilfred developed a keyboard-based instrument - the Clavilux - which was based on related principles. Whereas Rimington's instrument was based on the unmediated projection of light upon a screen, however, Wilfred's Clavilux appears to have been more complex, making use of multiple projectors, reflectors and coloured slides. As such, Wilfred appears to have had a far greater degree of control over visual form than Rimington. Wilfred is also significant for his conception of the resultant art, which he termed *Lumia*. Having seen previous colour organs which sought to link musical pitches with specific colours, Wilfred came to the conclusion that such a link was fundamentally flawed. As such, he positioned his work as being entirely separate from music, and to be performed in silence[43]. Wilfred also designed small scale versions of the instrument to be played at home, and

[43]A significant exception was his performance of a light composition in conjunction with the Philadelphia Orchestra's performance of Nikolai Rimsky-Korsakov's *'Scheherezade'*, though Wilfred was careful to note that his aim was to create a specific atmosphere around the music, as opposed to following it note-by-note (see p76-77, *'Visual Music: Synaesthesia in Art and Music since 1900'*[51]).

hoped (as did Rimington) that the performance of Lumia would become a practice as widespread as that of music.

An instrument that could be considered as somewhat separate from the colour organ tradition however (at least inasmuch as it was not performed with a piano-style keyboard), is Oskar Fischinger's *Lumigraph*. William Moritz describes the instrument thus:

> *"[Fischinger's] Lumigraph hides the lighting elements in a large frame, from which only a thin slit emits light. In a darkened room (with a black background) you can not see anything except when something moves into the thin "sheet" of light, so, by moving a finger-tip around in a circle in this light field, you can trace a colored circle (colored filters can be selected and changed by the performer)."*[91]

Unlike Wilfred, Fischinger frequently performed his instrument in conjunction with (recorded) music. While there is no evidence he subscribed to the idea of a 'colour scale' as did Rimington, his widow Elfriede notes that he was very specific about which colours should be used, and at which particular times, when accompanying pieces such as Sibelius' *'Valse Triste'*[64]. Similarly to both Rimington and Wilfred, Fischinger also had hopes that Lumigraphs would be manufactured on a large scale and used in the home.

2.2.4 Audiovisual Instruments

Perhaps the most significant specifically audiovisual instruments have been developed by Golan Levin, for his *Audiovisual Environment Suite*[80]. For the five environments in the suite (Levin uses the term Audiovisual Environment as opposed to Audiovisual Instrument), Levin used a painterly interface metaphor to define how the performer interacts with the systems. As such, all the environments make use of a pointing device (typically a stylus/Wacom tablet, sometimes in conjunction with a mouse) for their input, and make use of gestures common to drawing. Levin's reasoning for this approach is that the vast majority of the prior instruments in this field[44] have been designed from a musical point of view, and as such, have interfaces which privilege musical expression over that of the visual arts. His argument (perhaps slightly simplified) is that truly successful audiovisual instruments require an interface that gives the performer the same amount of control over visual dimensions such as colour and spatial form, as is provided for the sonic domain. His choice of primarily stylus-based interfaces then, was an attempt to leverage the performer's existing dexterity and experience with such an interface, in creating spatial forms. It could perhaps be argued, however, that the use of stylus-based interfaces, and indeed, the use of the word 'painterly', results instruments that privilege the visual over the sonic. Specifically, the

[44]Levin's thesis[80] includes a thorough history of audiovisual performance tools, which naturally overlaps with some of the areas discussed in this thesis.

performer's prior experience with the stylus as a tool for drawing would seem to position the instrument, at least initially, in the visual domain.

In the NIME[45] research community there are frequently papers published which discuss the design of audiovisual instruments. For the most part, however (and at the risk of making a sweeping generalisation), these papers tend to say very little about the connection between sound and visuals - why particular synthesis methods (audio and visual) were chosen, how they are linked, what approaches were more successful than others and why that might be, whether the connection is entirely subjective (and will thus vary from person to person), or whether there are elements which will be perceived in the same way regardless of a person's cultural background etc. Perhaps because these papers are published in a primarily musical field, they tend to focus for the most part on mappings between an input device and the sound generation mechanisms, with the result that the visual aspect of the instrument appears tacked on as an afterthought.

One such example is the paper *'Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis'*[86], by Ali Momeni and Cyrille Henry. In this paper the authors discuss a mapping method whereby the performer manipulates an internal

[45]New Interfaces for Musical Expression - I use the term to refer not just to the annual conference[17], but the wider body of research conducted into the design of musical instruments.

mapping layer with a defined set of behaviours (they use a set of mass-spring models), which is then mapped using what they call *"classic mapping layers"* to the sound and visual generators. They propose that the use of this independent mapping layer performs well as a way of connecting a small number of inputs (the physical interface) to a high number of outputs (sound/video generation), as well as providing a model which is relatively easy for the performer to understand. In discussing the visual element of the instrument, however, the authors give only a very general overview. They state that since a single independent mapping layer drives both sound and visuals, there is *"an intrinsic connection between the audio and the video that is a direct result of their common sources of control."*[46], however this statement is not explored further. The exact nature of the generated visuals is not made clear[47], and I would argue that just driving the two synthesizers from the same input is not necessarily enough to create a (perceivable) connection, especially since the materials generated in the two domains may be entirely at odds with each other. In one diagram[48] they show a second mapping layer directly connecting the audio and the visual generators, but this is never discussed in the text, and it is thus hard to decipher how it fits into their general approach.

[46]p.57, *'Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis'* [86]

[47]An audiovisual environment named *chdh* is mentioned, but the next section claims the authors are committed to using Max/MSP and Pure Data for their mapping strategies, and it is not made clear whether the two are related somehow, or are two separate projects.

[48]Figure 2, p.50, Ibid.

A slightly more satisfying discussion of these issues appears in the NIME paper *"A Structured Instrument Design Approach: The Video-Organ"*[50], by Bert Bongers and Yolande Harris. In it, the authors describe a kind of modular audiovisual instrument, where pre-built interface modules are assembled into an instrument, the design of which is determined by the piece or composition which it is to perform. Though they don't appear to mention what method they use to generate sound, the visuals consist of video clips, with varying playback parameters. In discussing how sound and visuals are linked in their instrument, the authors note that the primary similarity between the two domains is movement. In constructing the instrument for a particular performance, therefore, they appear to take a number of video clips and map the motion therein to gestures which may be performed with the interface modules, and the sound generator. Again, however, this fundamental idea is not elaborated upon, though the authors do include a brief discussion of the narrative issues which arise from this way of working.

It could be argued that the reason these papers spend so little time on the complicated issue of audiovisual relations is that they are published and presented in music journals and conferences, and therefore choose to focus on the more explicitly 'musical' elements of the instruments discussed. Having said this, however, there appears to be a wider lack of discussion about the audiovisual relationship with respect to instruments. While festivals such as ARS Electronica, and the concerts performed at confer-

ences such as the ICMC and NIME, demonstrate that there is a lot of interest in the creation of audiovisual instruments, there is a significant lack of writing or theory available which investigates the issues particular to such instruments. This deficiency is particularly stark when compared to the amount of research conducted into mappings for the design of (purely) musical instruments.

As an addition to this section, it should be noted that there is a related class of musical instrument which uses a visual output to communicate the instrument's state to the performer. In this case the visuals tend to take the form of a dynamic graphic representation of the underlying synthesis algorithm. Scanned synthesis[108] lends itself particularly well to this kind of approach[49], as the relation of the performer's gestures to the instrument's timbre may be somewhat opaque if the performer cannot see the motion of the haptic generator (typically a physically-modelled string). Generally, however, the visuals for this kind of instrument tend to be primarily functional and lacking in any real expressive range. Indeed, the impulse behind the designers' use of a visual output seems to be their desire to solve the familiar mappings problem of how to explain to the performer how the instrument works, rather than a desire to create a deliberately audiovisual instrument.

[49] See for example the *'Meta-control of Scanned Synthesis'* section in [46].

2.3 Psychological Research

2.3.1 The Unity of the Senses

In *'The Unity of the Senses'*[82], Lawrence E. Marks investigates the numerous ways in which human senses are interrelated. A significant point made in the book is that ultimately, the senses are fundamentally connected in at least one way. That is, our brain creates a coherent and consistent model of our environment from the data it receives from our various senses. As such, when we hold an object in our hand, we perceive the object that we can feel in our hand as being the same object that we can see is in our hand. Marks notes that there have been some studies which suggest that this ability is - at least to some degree - learned rather than innate. Referring to research done with patients who were blind since birth and then had their sight restored, Marks notes that it took the patients some time before they could match their tactile experiences with their visual ones.

One of the theories Marks discusses in the book is that of Primary Qualities (similar to what Aristotle termed the common sensibles). Espoused by various philosophers over the years, the theory holds that there are a number of qualities which are common across all the senses, and which link them to some degree. The primary qualities are generally enumerated as size, shape, quantity, and motion, with John Locke also including solidity in the list.

Secondary qualities refer to those qualities which are specific to a single sensory domain, such as colour, pitch, warmth and cold, and do not easily cross over to other domains (though it may be argued that the terms warmth and cold are frequently applied to other domains, as we will see later). The theory says that our perceptions of the primary qualities of a physical object resemble the object's actual qualities, while secondary qualities do not. The example Marks gives with respect to secondary qualities is that of colour; the wavelengths of light reflected by an object do not resemble the colour that is actually perceived.

One of Marks' main concerns in investigating the unity of the senses is to look at metaphor in poetry, and particularly how words describing the sensory attributes of one sense can be applied those of another sense. The fact that Rudyard Kipling can be understood when writing *"the dawn comes up like thunder"*[50] points to the human ability to construct inter-sensory associations that do not objectively exist in nature. It also suggests the role language plays in the construction and understanding of such associations. Marks emphasizes the findings that synaesthesia is far more common among children than adults, suggesting that as children become adults, language takes synaesthesia's place somewhat, providing a far more flexible, versatile system of inter-sensory correspondences than is possible with synaesthesia's 'hard-coded' links.

[50]See p.1, *'The Unity of the Senses'*[82].

2.3.2 The Visual and Auditory Senses

As this project is about investigating the ways in which sound and visuals may be linked, some discussion of the two senses' specific qualities is needed. As is perhaps obvious, the two senses specialise in different areas; sight offers greater spatial than temporal resolution, while sound offers greater temporal than spatial resolution. This suggests that, in linking the two, it should be possible to get the best of both worlds - to construct a better-realised spatial soundfield (at least, perceptually speaking) than would be possible with sound alone, and to create a visual experience with a far tighter (and possibly more visceral[51]) temporal aspect than you see, for example, in silent film. This is an idea which will be returned to in the section on synchresis.

An interesting experiment conducted by Wolfgang Köhler demonstrates the ways in which sight, sound and language are linked:

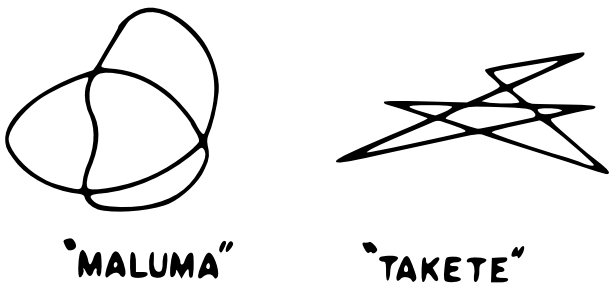


Figure 2.1: Matching Shapes to Words[76]

[51]It could be argued that sound is a far more visceral medium than sight. Sonic impacts, for instance, tend to carry a far greater weight than purely visual impacts and in cinema particularly it is the sonic information that conveys the visceral 'feel' of an impact, far more than the visual does.

Figure 2.1 shows two drawings which the subjects of the experiment were asked to match to the two made-up words *"maluma"* and *"takete"*. Most subjects matched the rounded, smooth figure on the left to the *"maluma"* word, with *"takete"* being matched to the jagged, angular shape on the right. The inference is that the subjects sonified the words in making their connections, with *"maluma"* having a longer duration and smoother sonic characteristics than the short, more percussive *"takete"*. In *'The Unity of the Senses'* Marks notes that this conclusion was backed up by further research by Holland and Wertheimer, with nonsense words and visual forms found to be described along the same dimensions, of *"angularity, strength, and size."*[52].

2.3.3 Synaesthesia

Any discussion of the psychological relationship between sound and visuals will necessarily have to cover synaesthesia, the condition whereby individuals will involuntarily associate the sensations from one sense with those of another. An apparently common example is the association of specific musical tones with specific colours, though there seem to be various other associations (sound to taste, etc.) linked to the condition.

Synaesthesia has been the basis for numerous audiovisual artworks, a common example given being Skriabin's *'Prometheus'*,

[52]p.78, Ibid.

where the composer included a part for a *'colour keyboard'* - a keyboard instrument which generated colour instead of sound. Skriabin created a specific 'colour scale' for the piece, where each note in the octave was related to a particular colour. This idea of 'colour scales' is one that has been proposed at various points throughout history, with the following table from rhythmiclight.com[55] giving a good comparison between the different scales created:

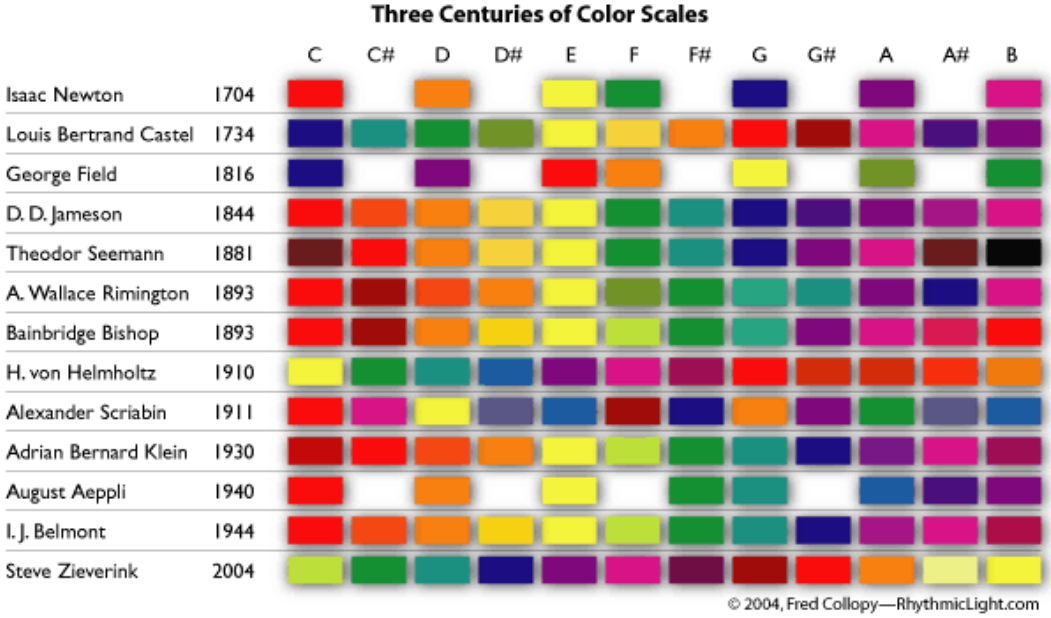


Figure 2.2: Comparison of Colour Scales

As an artistic concept, however, this form of synaesthesia has tended to receive quite a large amount of criticism. One of the problems is that there is relatively little correlation between different synaesthetes' associations between pitch and colour, and to a non-synaesthetic audience, the connections made by the artist of-

ten seem opaque and unintuitive. Synaesthesia is also usually associated with fixed, static relationships between sensory domains. As such, strictly synaesthetic artworks are predicated on the idea that, for example, red corresponds to middle C, and will *always* correspond to middle C. It could also be further argued that the addition of colour in this way adds very little to the music.

Despite these misgivings, however, in *'The Unity of the Senses'* Marks notes a number of interesting correspondences between synaesthetes (and even non-synaesthetes). For example, he observes that in coloured hearing synaesthesia (i.e. sounds produce colours in the synaesthete's mind), the brightness of the imagery varies directly with the brightness of the sound. He also notes that the size of the 'generated' image varies according to the pitch of the sound, and its loudness (high pitches equal small sizes, and loud sounds correspond to large sizes). When it is specifically music that is generating the visual illusions, he makes the observation that fast music generates sharper and more angular visual shapes than slow music.

Intriguingly, Marks also points to research which found that, in discussing cross-sensory analogies (e.g. 'the sound is warm'), non-synaesthetic individuals tended to make the same connections as those discussed above with regard to true synaesthetes. Visual size is frequently thought of as related to auditory pitch and loudness, with pitch also thought of as related to visual brightness, in the same way as above. In addition, he notes that colours are almost

universally related to temperature in the same way among individuals - i.e. that colours with long wavelengths (red, orange, yellow) are linked to heat, while those with short wavelengths (green, blue) are linked to cold.

What this all points to is that even though synaesthetic perception differs among synaesthetes, and strict synaesthetic relationships may appear obscure and opaque to non-synaesthetes, there are some perceptual connections between the different senses that nevertheless appear to be somewhat universal among human beings[53].

[53]Though cultural context may play a part too - for example, wikipedia claims that in China the colour red signifies success and happiness, as opposed to (roughly) 'warning' in the West (see the Symbolism section; [30]).

Chapter 3

Synchresis

3.1 The Audiovisual Contract

In the book *'Audio-Vision'*[54], Michel Chion describes the relationship between sound and visuals in film as an Audiovisual Contract. He uses this phrase to emphasize that the relationship is not something that naturally occurs due to some inherent connection between our perception of sight and sound, but rather that it is specifically and deliberately created. In the first chapter, Chion discusses the relationship as an illusion:

> *"This book is about precisely this phenomenon of audiovisual illusion, an illusion located first and foremost in the heart of the most important of relations between sound and image ...: what we shall call* added value.*"*[1]

[1] p.5, *'Audio-Vision'*[54].

By *added value*, Chion refers to the way in which sound in film enriches and alters how the image is perceived. It does this in such a way that this "added value" appears to naturally be part of the image itself, with the sound appearing to simply duplicate what was already present. This added value is particularly present in the phenomenon of synchresis.

3.2 A Definition of Synchresis

> *"the spontaneous and irresistible weld produced between a particular auditory phenomenon and visual phenomenon when they occur at the same time."*[2]

Essentially describing the work of foley artists in film, synchresis refers to the way that otherwise unconnected sound and visual events can become entwined, and inseparable in the minds of the audience. It refers to the way that foley artists are required to build up the sound environment in films from scratch, recording footstep sounds to match the film of an actor walking, creaking sounds to match a door opening, etc. Although the source of the recorded sounds is completely removed from the source of the filmed visuals, the audience does not perceive sound and visuals as separate - they are perceived as a single, fused audiovisual event.

An interesting example is that of the film punch. In real life, punching someone rarely makes much sound, regardless of how

[2] p.63, Ibid.

much pain is inflicted (and ignoring any vocalisations by the unfortunate victim), yet in film we are accustomed to hearing an exaggerated *'whack'*- or *'thump'*-type sound. This phenomenon is so prevalent in film that seeing a punch filmed naturalistically seems somehow false, or unreal, and lacking in substance. This lack of substance perhaps explains why the practice began - the sound helps to emphasize the physical nature of what's being depicted, and fix it as an impact in the temporal domain.

It could perhaps be argued that this emphasizing effect makes the punch a special case, but as alluded to previously, synchresis is widespread throughout film, and not necessarily used to root the images in a more physical domain. Another example is that of the film of someone walking - Chion argues that in this case, synchresis is unstoppable, due to our expectation that there will be sound associated with the actor's footsteps. As we're expecting this sound, the foley artist is able to substitute almost any sound, and our brain will still form a connection between sound and visual[3]. Based on his observation that, when playing a stream of random audio and visual events, certain combinations will adhere better than others, Chion notes that synchresis *"is also a function of meaning, and is organized according to gestaltist laws and contextual determinations"*[4]. In addition he notes that while cultural habits may have a significant part to play, there is some evidence

[3]The example Chion gives of this is Tati's *'Mon Oncle'*.
[4]p.63, ibid.

that it also has an innate basis. It is this innate basis that I have focussed on in my attempts to link sound and visuals as tightly as possible.

3.3 Motion as the Connection

The longstanding presence of synchresis in film, and particularly the way that the illusion is apparently rarely perceived as such by the audience, suggested a very promising avenue of exploration in my attempts to link sound and visuals. The following is an attempt at examining how the phenomenon works, with a view to defining some basic principles which can then be used to create synchresis in an audiovisual instrument.

I'll take the footsteps example as a starting point. Video 1 shows my then-flatmate walking across our living room floor, with the accompanying soundtrack as it was originally recorded. Looking at the visuals, we see a foot moving towards the floor, colliding with it and lifting off, an action that is repeated four times in the video. On the soundtrack we hear his footsteps as distinct sonic events with a sharp attack and short decay, synchronised with the points of collision we see in the visuals. My hypothesis is that, where synchresis is involved, it is motion that allows our brain to form a connection between sound and visuals. Looking at Video 1 again, the most relevant information we get is from the foot in motion, and particularly its collisions with the floor. If we view motion in sound

as being the change of sonic attributes over time, we can hear the footstep sound as being comprised of a rapid rise in the amplitude envelope followed by a short decay, as well as the change of spectral content from a complex noise-based spectrum to a slightly simpler spectrum based on the resonant properties of the wooden floor (supplemented of course by the natural reverberation of the room it was recorded in). Looking at the two domains together therefore, there is a clear connection between the visual point of collision and the initiation of the complex sonic event that makes up the footstep sound. It is the fact that the motion of the two domains match up in this way that tricks our brain into perceiving the combined audio and visual streams as a single event, rather than two separate sensory streams.

To take this idea further, I've replaced the soundtrack in Video 1 with a series of enveloped sine tones, in Video 2. In this case, much of the context and detail is removed from the original soundtrack, and the only sonic motion present is that of the amplitude envelope applied to the sine tones. Following the previous hypothesis though, visual- and sonic-motion still coincide, and the perception when viewing the video is still of a sequence of fused audiovisual events. Taking it further still, Video 3 sees the visuals replaced by two abstract boxes, moving towards each other and colliding four times. Again, most of the context has been removed, with both domains now reduced to exhibiting little more than the motion of simple attributes, yet synchresis still functions, I would argue, just

as strongly as in the original video. This ability to construct a tight audiovisual connection out of the most abstract elements is key to realising my intentions with Ashitaka, as it suggests that it is therefore possible to build up a complex series of audiovisual connections that the audience (that is, *any* audience) will understand innately.

As noted by Chion, however, the footstep example is a very specific case of synchresis. So, what if we were to apply the same line of reasoning to another kind of motion? Video 4 shows a radio-controlled car being driven across my living room floor, and is intended to demonstrate a kind of linear motion, in contrast to the collision-based motion of the footsteps example. In this case the camera is positioned so that the car moves across its field of view in a straight line, and the sound is simply that of the car's electric motor. Though the effect is perhaps slightly weaker than the footsteps example, the viewer still perceives a fused audiovisual event/stream rather than separate information streams. Using the same process we followed for the footsteps example, Video 5 sees the original soundtrack replaced by a sine tone. In this case the sonic motion was originally focussed solely on the pitch of the tone, though I found that it was necessary to apply an amplitude envelope also to mimic the way the car moves towards then away from the camera. Without this addition, the sound did not adhere strongly to the visuals[5], my theory being that this enveloping

[5]To go further still a doppler effect could be added to strengthen the connec-

is necessary because the car is not always visible (as it goes outside the camera's field of view). Again, though the effect is slightly weaker than in the footsteps example, synchresis still functions. As before, Video 6 then replaces the visuals with a simple abstract box, moving in a straight line across the camera's field of view, and again the viewer should still perceive sound and visuals as intimately connected. To investigate the apparent requirement of an amplitude envelope, I created a further two (purely abstract) videos. Video 7 is the same as Video 6 without the amplitude envelope, and Video 8 shows the box moving in straight lines, but not leaving the camera's field of view, and with no amplitude enveloping. Comparing Videos 7 and 6, 7 does seem to have a weaker audiovisual connection, while the synchresis in Video 8 is relatively strong, and does not appear as if it would benefit greatly from the addition of an amplitude envelope. I would speculate that there may be two issues at work here; firstly that when a moving object is not visible synchresis may require extra information leading up to that object becoming visible (or vice versa), and secondly that our experience with objects on camera may have a part to play, by encouraging us to expect objects moving towards the camera to get louder as they get nearer.

Experience would seem to be key to the phenomenon of synchresis. Our experience of the physical world tells us that when we can see an object is in motion, there will tend to be some kind of sound

tion, though I did not feel it absolutely necessary.

associated with it, and the converse is also true - if we can hear a sound, we tend to expect there is some kind of visual movement associated with it[6]. In the physical world, motion gives rise to visual and auditory stimuli, and we have naturally developed senses to detect these stimuli (and thus, the underlying motion). While it is not true that every visual movement in the physical world is accompanied by an audible sound (see for example someone waving their hand), we have enough experience of visual movements being accompanied by some kind of auditory 'movement' that our brain will form a connection even when this related movement has been falsified, as in synchresis.

It is perhaps obvious, but it is important to clarify that we are talking about *perceivable* motion here. A sine tone could be considered as possessing a kind of motion, in that it relies on the vibration of particles in the air to be heard. To the human ear though a sine tone appears to be fundamentally static - assuming amplitude is held constant - and is perceived as a single tone which does not possess any motion of its own. It follows therefore that if the audience cannot perceive some kind of related motion in sound and visuals, synchresis cannot function, as there is no longer any perceivable connection between the two domains.

[6]Though it could be argued here that our relatively recent history of listening to music on radios, stereos, iPods etc. has severed this (presumably automatic) connection somewhat.

3.4 A Psychological Perspective

While there have been a number of psychological studies conducted into specific audiovisual illusions, there do not appear to have been any studies which focus explicitly on synchresis. Despite this deficit, I would tentatively suggest that the following audiovisual illusion studies do appear to support my hypothesis that when sound and visual streams share related motion, the brain will perceive the two as one.

A well-known audiovisual illusion is the McGurk Effect[83], named after one of the psychologists who first identified it. This illusion explicitly concerns speech perception, and how a person's speech may be perceived differently if coupled with a false video stimulus. Specifically, McGurk and MacDonald found that if the sound of someone saying /ba/ is accompanied by video of someone saying /ga/, the audience will actually perceive the sound as /da/.

A second illusion is described in the paper *'What you see is what you hear'*[100], where the authors discuss an illusion in which auditory stimuli appear to influence visual perception. In this case (known as the Illusory Flash illusion), the authors displayed a single visual flash accompanied by multiple auditory beeps. They found that, if two beeps accompanied a single flash, the audience would perceive two flashes, matching up with the two beeps. This illusion persisted even among participants who were aware of the

illusion beforehand. The authors found that the threshold for this illusion to occur is on the order of 100ms - if the beeps were spaced further apart than that, the audience would perceive only a single flash.

Perhaps the most well-known audiovisual illusion is that of ventriloquism. First discussed in [71][7], ventriloquism revolves around a sound appearing to emanate from a spatial position different to its real spatial position. When a particular visual stimulus is presented to accompany an aural stimulus which has a different spatial position to the visual one, the audience will perceive the sound as emanating from the visual stimuli's spatial position.

Most promising are the conclusions reached by Soto-Franco and Kingstone in the *Multisensory Integration of Dynamic Information*[103] chapter of *The Handbook of Multisensory Processes*[52]. The chapter consists of an overview of previous studies conducted into the perception of motion, and a discussion of the authors' own studies in this area. A number of different experiments are presented, starting with experiments which demonstrate a degree of integration between domains (sight and sound) when static stimuli are involved (e.g. ventriloquism), then with dynamic stimuli. One of the first results of note from these experiments is that the degree of integration appears to be far higher when dynamic stimuli are involved than with static stimuli. Indeed, the authors conclude the

[7]And also discussed by Michel Chion in *Audio-Vision*[54].

first part of the chapter with the following statement:

> *"Considered together, the data point to the conclusion that the experience of motion is critical for cross-modal dynamic capture to occur, and therefore this illusion reflects the integration of dynamic information."*[8]

While these experiments do not specifically look at the phenomenon of synchresis, they do demonstrate that the brain makes some intriguing (and perhaps unexpected) connections between information received from the visual and aural senses. In addition, the previous quote appears to support my hypothesis that the key factor in convincing the brain to form a fused audiovisual connection is motion.

[8] p.57, [103].

Chapter 4

Mappings

4.1 A Film-Based Approach

My first attempt at developing mappings based on synchresis was derived from an examination of the motion involved in filmed moments of synchresis[1]. This approach began with the classification of different types of motion, with the aim being to build up a vocabulary with which the two domains (sight and sound) could interact with each other.

To start with, I made a distinction between *Forms of Motion* and *Domains in Which Motion May Occur*. *Forms of Motion* refers to the way in which something (i.e. an aspect of either the aural or visual domain) moves, and this is assumed to be a general function which is equally applicable across both domains. So for example, the simplest *Form of Motion* would be a constant velocity motion,

[1] i.e. no attention was paid to a performer's interactive input - I solely considered the 'non-realtime' construction of fused audiovisual relationships

where 'something' moves in a straight line, from one point to another. *Domains in Which Motion May Occur*, on the other hand, refers to the 'something' that moves, and will tend to encompass various domain-specific attributes. A simple example of a visual *Domain* would be the spatial position of an object, while an aural *Domain* could be the pitch of a continuous tone. To form a complete (synchretic) audiovisual connection therefore, an author would have to connect an audio *Domain* and a visual *Domain* with a particular *Form of Motion*. The theory is that this should be sufficient to create an instance of synchresis, and fuse sound and visuals together into a single (perceived) entity.

4.1.1 Forms of Motion

Table 4.1: Forms of Motion

Constant Velocity
Collision-Based Motion
Periodic Motion
Gravity-Based Motion
Discontinuous Motion

Table 4.1 demonstrates some example *Forms of Motion*. A brief explanation is perhaps required:

- **Constant Velocity:** This should be fairly self-explanatory. Compared to the other forms of motion, this could perhaps be seen as providing a weaker connection between audio and visuals, since there are no discrete temporal events. This does

not mean it cannot prove useful in certain situations, however. We have already seen an example (the radio-controlled car) of constant velocity motion in the Synchresis chapter.

- **Collision-Based Motion:** This is primarily derived from the footstep example in the Synchresis chapter. In the visual realm, it refers to objects colliding with each other and then reacting. In the audio realm, however, it refers to the kind of sound associated with collisions, referring to the way that, while the visuals are in motion before and after the collision, sound will only be instigated at the point of collision (assuming it is not already in motion from a previous collision).

- **Periodic Motion:** Again this should be fairly self explanatory, referring to motion that repeats itself in a perceivable fashion.

- **Gravity-Based Motion:** Related to collision-based motion in that it is based on physical phenomena, this essentially refers to attraction/repulsion forces such as gravity. This is probably most easily perceived visually, though aurally it could refer to motion that gradually decreases or increases in range.

- **Discontinuous Motion:** This refers to sudden discrete jumps, as opposed to the mostly smooth motions described previously. This form of motion is most obviously visible in an artform such as music video.

4.1.2 Domains in Which Motion May Occur

Table 4.2: Domains in Which Motion May Occur

Visual		Aural
Position (of an object)		Instantaneous Amplitude
Size (of an object)		Pitch
Rotation		Brightness
Smoothness		Energy Content
Articulation (of an object)		Spatial Position
Pattern		Noise-Pitch

Table 4.2 provides some example *Domains in Which Motion May Occur*. Rather than go through each entry, I'll just provide a brief explanation of some of the less obvious entries:

- **Smoothness:** This refers to how smooth or coarse a particular part of the visual is. That part could be the shape of an object, or a more general impression of how colours (and particularly patterns) contrast with each other.

- **Articulation (of an object):** This refers to particular visual objects which may articulate their shape, in much the same way as humans and animals do with their arms, legs etc.

- **Pattern:** Refers to a visual motif which is recognisably periodic, whether it is viewed statically or in motion. This is particularly relevant to John Whitney's notion of Visual Harmony [111].

- **Brightness:** Refers to how perceptually 'bright' the sound is, something which is primarily defined by the high frequency

content of the sound.

- **Energy Content:** Refers to the perceived energy content of the sound, something which is perceptually similar to the Root-Mean-Squared amplitude of the sound.
- **Noise-Pitch:** Refers to where a sound is on the continuum between pure noise, and a pure (sine) tone.

4.1.3 Example Mappings

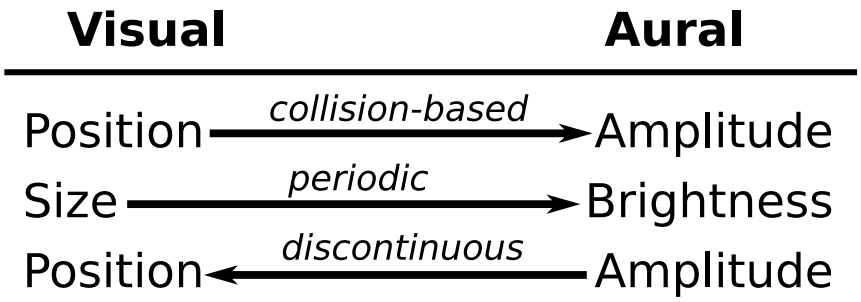


Figure 4.1: Some Simple Example Mappings

Figure 4.1 demonstrates how the *Domains* may be connected via *Forms*, with some basic example mappings.

The first mapping here is based on the footsteps example from chapter three. The position of a visual object is mapped to the amplitude of a sound (say, a sine tone, for simplicity), via a collision-based *Form*. As such, when the visual object collides with another visual object, the amplitude of the sine tone (which has until now

been silent) experiences a sudden rise, followed by a shallower decay.

The second mapping (see Video 9) describes a visual object which pulsates periodically. This motion is then mapped to the brightness of a sound (in this case we just modulate the cut off frequency of a low pass filter, though a more sophisticated approach would probably provide more satisfying results).

The final mapping (see Video 10) describes a situation which is relatively common in music visualisers and, to some extent, music videos. The amplitude of the audio is followed so that a visual object is jumped to a new, random position on the detection of a transient. This provides an obvious visual correlate to the transients in the audio, and would tend to be more successful with sounds which contain a high number of percussive impacts.

The intention of these examples is to demonstrate how this system of *Domains* and *Forms* can work. The next step is to then build up a number of these mappings to create a complete instrument, where the sound and visual outputs are perceived as intimately entwined, and not as two separate, distinct entities. This would hopefully provide a more satisyfing and interesting connection than is visible in the simple examples presented here.

4.1.4 Evaluation

This discussion of *Forms* and *Domains* was my first attempt at laying out a formal system by which I could construct an audiovisual instrument. As my initial inspiration was derived from Michel Chion's film theory, it naturally functions relatively well when it comes to describing the process of synchresis in film. What it does not take into account, however, is any real concept of interactivity in the process. The previous example mappings describe the audiovisual connection fairly well, but have no conception of a performer's role in the system. From simply looking at the previous examples, it is unclear how a performer would manipulate and interact with these audiovisual objects. This is a problem which became more and more apparent as I developed a number of experimental audiovisual instruments (more of which in Chapter 7) based on this approach, prior to the development of the Ashitaka instrument.

4.2 Audiovisual Parameters

In order to tackle the problem of the performer's role in the structure of an audiovisual instrument, I developed a second approach based on the idea of 'audiovisual parameters'. This idea is derived from my own conviction that an ideal audiovisual instrument would, first, have audio and visual outputs which are perceived as

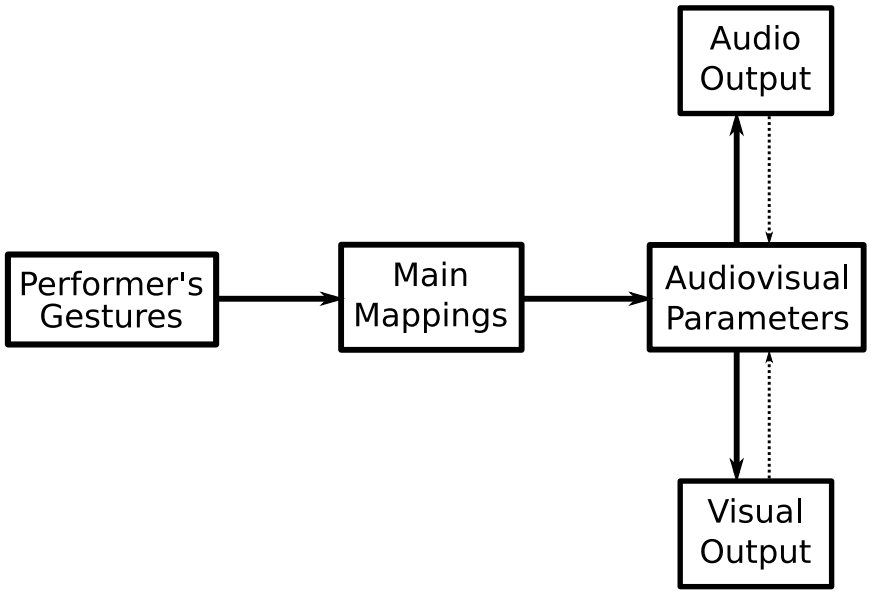


Figure 4.2: An Overview of the Instrument's Mappings

inextricably linked. Secondly, it would have a number of audio-visual 'facets', where an observer examining the instrument could say, "this particular visual part is fused to that particular sonic part", and so on. In this mapping scheme these 'facets' are collections of explicit mappings called *Audiovisual Parameters*.

Figure 4.2 portrays an overall view of the mapping scheme for an instrument based on this approach. As is perhaps apparent, it has more in common with existing research conducted into mappings for musical instruments than the previous film-based approach does. Before discussing the overall view however, it is necessary to first define what an Audiovisual Parameter is.

Figure 4.3 is a schematic of a single Audiovisual Parameter. As

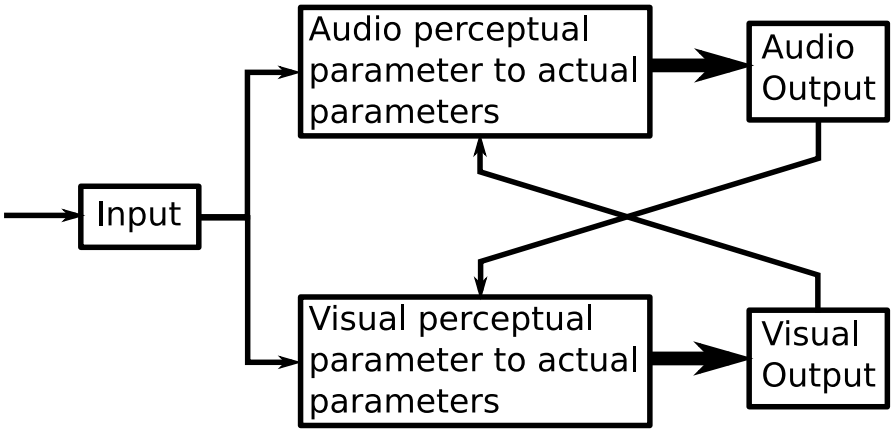


Figure 4.3: An Audiovisual Parameter

mentioned already, an Audiovisual Parameter is a mapping stage in itself. Partly deriving from the 'perceptual spaces' approach discussed in the *'Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces'*[46] paper, an Audiovisual Parameter primarily consists of a single perceptual audio parameter linked to a single perceptual visual parameter. The idea here is to link perceptual parameters[2] that 'make sense' according to our experience of the physical world, so visual size could be linked to pitch, etc. A perceptual parameter here is essentially a one-to-many mapping, with a single input (say, aural brightness)

[2]By 'perceptual parameter' I mean an aspect of a sound or moving image which can be defined in perceptual terms - in how we perceive it. e.g. listening to a violin we can say it sounds particularly bright or dull, it is smooth or rasping etc. While these kind of descriptions would be very informative to most listeners, they do not necessarily map directly to the low level technical parameters of the violin's sound generation mechanism. Such technical parameters would be things like bow force and velocity, which are not necessarily perceivable to an untrained ear. The aim is to design an instrument by describing it in general terms, easily understood without intimate knowledge of the synthesis methods it uses.

potentially mapped to a number of low level parameters in the associated synthesizer. The use of this perceptual intermediary stage is necessary because most audio (and visual) synthesis algorithms make use of parameters which are not necessarily easily understood simply by listening to (or watching) the output.

The two perceptual parameters are both driven by the same input, which serves to form the first part of the audiovisual connection. The second part involves feedback from the opposing domain being returned to the perceptual parameters. The reason for this additional feedback can be seen if we look at the example of an instrument using a physical model for the audio output. With most audiovisual instruments, it would seem natural to want to map perceived audio amplitude to some visual parameter. Physical models, however, do not provide the performer with complete control over their output amplitude, as most models will tend to resonate for some time after the performer has ceased to excite them. As such, if we map audio amplitude to some visual parameter without any feedback, there is likely to be a disconnect when the performer stops exciting the model, as the model will continue to resonate, but the visual synthesizer will stop moving. Therefore, some degree of feedback between the two domains is necessary (though it is likely that different combinations of sound and visual parameters will require different amounts and types of feedback).

To return to Figure 4.2 then, an audiovisual instrument based on this scheme would have a number of Audiovisual Parameters,

mapped to the performer's gestures via the Main Mappings block. This block is intended to describe a more traditional set of musical instrument mappings. Indeed, it should be possible to replace the Audiovisual Parameters with simple perceptual audio parameters, to create an entirely sonic instrument. The aim with the Main Mappings block is to focus on the playability of the instrument, on how easy it is for the performer to articulate complex gestures, and how enjoyable it is to play. As such, when designing an instrument using this scheme, the designer would first create a set of Audiovisual Parameters, then, when satisfied, move to the Main Mappings block to define how the performer interacts with these parameters, presumably using some combination of one-to-many and many-to-one mappings.

4.3 Gesture Recognition

The preceding sections have focused on the low level, detailed interactions between performer, sound and visuals, but Digital Musical Instruments (and audiovisual instruments, as discussed here) also have the potential for mappings which work at a higher level. This higher level could take the form of gesture detection, whereby the software detects when the performer articulates particular gestures, and triggers some action as a result. Such a strategy could involve changing the instrument's output in some fashion, changing the way it reacts to the performer's gestures, or triggering spe-

cific audiovisual events. Significantly, this approach could be used to strengthen the connection between the performer and the audience, if the particular gestures are different enough from the performer's usual gestures with the instrument. By triggering specific actions on the completion of certain gestures, the performer's role in the performance may be made clearer to the audience. This is because a clear chain of cause and effect is set up, which may not always be so obvious to an audience with respect to the lower level Audiovisual Parameters. This approach could perhaps be related to a form of spellcasting, as can be seen in various computer games, from Black & White[29] (PC) onwards.

Having said this, it is clear that not all interfaces will allow the performer to make gestures which are particularly noticable to an untrained audience. For such a purpose it is perhaps necessary to restrict the kinds of interface we are referring to in this section, to those which allow the performer to make big, self-evident gestures.

The recognition of human gestures is a large and complex subject, but since we are dealing with interfaces with a strictly defined set of variables our task can be somewhat simplified. As alluded to previously, basic gesture detection for mouse-driven gestures has been available in a number of applications (from Black & White to the Mouse Gestures plugin for Firefox, and of course the Nintendo Wii's accelerometer-driven games) for some time now. As such, I believe a good approach would be to borrow the method used in such applications, and simply extrapolate if more than two dimen-

sions (or variables) are required. The online article *'Recognition of Handwritten Gestures'*[60] explains how this form of gesture recognition is implemented. Acting on a complete gesture (stored as a sequence of points in 2D space) performed by the user, the process is as follows:

1. Scale the gesture to a predetermined size.
2. Space out the points in the gesture so that they are all an equal distance from their nearest neighbours. This is so that the speed at which the gesture was drawn does not affect the detection result.
3. Centre the gesture around the origin.
4. For every stored gesture we want to test against, take the dot product of the stored gesture and the newly modified gesture. A value of 1 means there is a perfect match, slightly less than 1 is a good match, and a low or negative number means there is no match.

As it relies on a dot product operation this method can easily scale to more than two dimensions for interfaces with a large number of variables. The only slight issue is that it is intended for complete gestures (started when the user presses the mouse button, stopped when they release it), and musical interfaces, by way of contrast, will usually output a constant stream of data. This can

likely be overcome however by simply testing the performer's input at periodic intervals.

Chapter 5

An Extended Mission Statement

This chapter outlines the various issues which this thesis attempts to (at least partly) resolve. The approaches taken to resolve to these issues will be discussed at length in the following chapters.

5.1 Artistic Issues

5.1.1 Sound and Visuals in an Instrument

Though comparatively small with respect to that of acoustic instruments, there is somewhat of a tradition of instruments which combine sound with some kind of explicit visual stimulus. The colour organs developed by various individuals and Levin's AVES all tend to fall within this area. Generally, however, the connec-

tion between sound and visuals in these instruments is either not discussed in great detail, or relies on certain assumptions or rules that I do not feel comfortable following. Levin's work, for example, seems to have been primarily concerned with developing a satisfying interface metaphor, rather than looking at how sound and visuals may be connected. The colour organs, on the other hand, are united by a belief that specific colours can be linked to specific pitches. While this belief has precedent in the medical condition of synaesthesia, it is not easily perceived or understood by an untrained eye, and often seems to rely on entirely subjective ideas of what particular pitches 'look like' in the visual realm.

This subjectivity, or opacity, in the relationship between an instrument's audio and visual output is something which I aim to avoid with Ashitaka. The reason for this is that there may be substantial scope for confusion or misunderstanding (on the part of the audience) with an instrument like this. It seems all too easy to create audiovisual works in which there is no apparent connection between sound and visuals, where the audience need detailed programme notes to understand the various relationships involved. While this is not necessarily a bad thing, my own intention is to create a connection which can be easily intuited by an audience who do not have prior knowledge of the instrument. Essentially, there should be some kind of 'way in' to the instrument's audiovisual world for the audience, inherent to the instrument's design. The basic principles of its operation should be easily understood,

in the same way that you can understand the basic principles of a guitar whether or not you can actually play it. Synchresis provides a potential solution here, in that it creates a very strong, perceivable connection between sound and visuals. Its longstanding existence in film can also be seen as 'proof' that it can be considered a more or less universal phenomenon, in that is perceived in much the same way by the vast majority of cinema-goers (if not the whole of humanity). This hopefully guards against the accusations of arbitrary or subjective audiovisual connections that are sometimes levelled at artists working with synaesthesia.

A second possible criticism of my own approach (of a fused audiovisual medium) is outlined in Nick Cook's *'Analysing Musical Multimedia'*[56]. He quotes Bertolt Brecht:

> *" 'So long as the expression 'Gesamtkunstwerk' (or 'integrated work of art') means that the integration is a muddle, so long as the arts are supposed to be 'fused' together, the various elements will all be equally degraded, and each will act as a mere 'feed' to the rest.' "*[1]

And goes on to say:

> *"[David] Kershaw echoes such language when he condemns the 'fawning dependency of visual image on sound' that results from too tight a co-ordination of the two; any such 'simplistic one-to-one relationship', he says, results in 'the*

[1] p.126, [56].

one becoming a mere embellishment, a condiment to the other'. "[2]

It should be noted that Cook is somewhat critical of these ideas, but he does also appear somewhat dismissive about art which uses the kind of 'one-to-one relationship' this thesis is advocating. In Chapter 3 of *'Analysing Musical Multimedia'* Cook outlines three models of multimedia; Conformance (both media say the same thing in the same way), Complementation (both media say the same thing in different ways), and Contest (both media actively contradict each other). Figure 5.1 (taken from *'Analysing Musical Multimedia'*[3]) portrays these three models together with the two tests which are used to distinguish between them.

According to Cook then, I believe the use of synchresis as a connection between sound and visuals belongs in the Conformance category, in that the two domains are intended to be fused together, and appear as one medium. Cook spends few words investigating this Conformant model of multimedia however, claiming that *"conformance is a much less frequently encountered category of IMM [an Instance of MultiMedia] than the existing literature might lead one to suppose"*[4] and that *"Conformance ... is hardly a model of multimedia at all, if (as I maintain) multimedia is to be defined in terms of the perceived interaction of different media."*[5]. As such, much of the

[2]p.126, Ibid.
[3]Figure 3.1, p.99, Ibid.
[4]p.102, Ibid.
[5]p.106, Ibid.

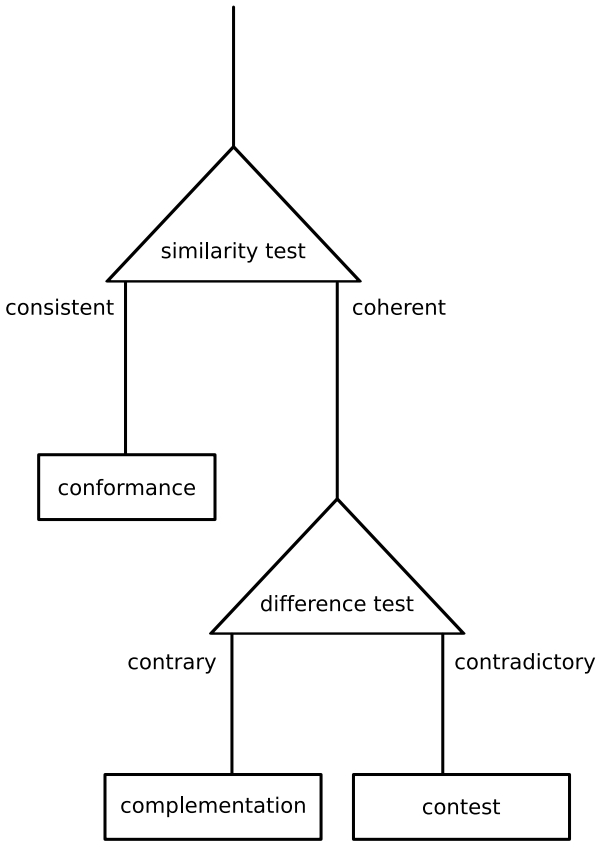


Figure 5.1: Nick Cook's three models of multimedia

book is spent discussing the other two models and looking at how metaphor may be used to construct complex relationships between different media.

The question which follows then is; if both sound and visuals are essentially doing the same thing, why bother with two media at all? Would it not be just as effective to work in a single medium, and discard the visual (or aural) component? This question, however, is predicated on the assumption that sound and visuals are

entirely separate entities, and that combining them in this way means one will always end up following the other. My own work though, revolves around the idea that it is possible to create a completely fused, audiovisual medium that is a single entity in itself, as opposed to the combination of two separate and distinct entities. Essentially the aim is to create an audiovisual material that works in the same way that, say, two rocks colliding in the physical world can be said to create an audiovisual event (we see them moving towards each other, and hear a noise as we see them collide). The use of computers means we can potentially create a far wider range of possibilities than we ever could if we limited ourselves to the manipulation of physical objects.

A second question raised by Cook's claim that conformance *"is hardly a model of multimedia at all"*, is; if it is not multimedia, what is this medium? One way to look at it is from the perspective of artists such as Kandinsky, and those who worked within the Visual Music field. From this perspective, music is seen as the play of abstract forms over time, and can thus be easily extended into the visual realm. Given that my own work for this thesis will be solely concerned with abstract visual forms and synthesized sonic materials, it seems reasonable to then conclude that the medium I am working in is, simply, music. This definition of music, however, could be thought of as quite limited, given that it ignores a substantial part of electroacoustic musical tradition. The use of sampling and field recordings (or similar forms) is not accounted for in any

way, as they involve different processes (related to the audience' recognition of familiar sounds, images, etc.) to those involved in the manipulation of sonic materials generated for specifically 'musical' (in the Western, pre-electroacoustic sense) purposes[6].

This issue does not necessarily present a problem, however. One of the aims of this thesis is to present a starting point for work in a specific audiovisual medium, and I believe that limiting ourselves to abstract forms is a reasonable foundation for such work. This does not prohibit such a medium from being used in conjunction with more representational forms at a later stage, simply that such an undertaking would involve various issues and processes not discussed in this thesis. One issue which crops up in a number of analyses of multimedia artforms[7] is that, to properly understand how the different elements in such a work interact, the first step is to look at each element alone, and understand how it works with relation to itself. The 'audiovisual music' proposed by this thesis could be considered as a single element in such an analysis, and thus should first be understood on its own merits, before any attempt is made to combine it with other media.

[6]By this I mean sound generated by physical instruments, synthesis algorithms etc. Sound does not quite fit into the standard visual dichotomy of the representational and the abstract (the sound of a violin is always representational of a violin), but it seems to me there is a similar distinction to be made in music. In classical music, the sound/timbre of the violin is secondary to the notes being played, which I would argue are a more or less abstract quantity.

[7]For example, [107] and [56].

5.1.2 Performance Context

A question which has not been raised yet is in what context the instrument developed for this thesis will be performed. The traditional context for musical performance is one where the performer(s) is/are situated on a stage, facing a passive audience. Expanding upon this, there is a marked difference between a classical concert and that of artists working within the popular music field[8]. As discussed in the Background chapter, popular music has always had a strong visual component, that is a significant part of the associated performance context. Whereas classical performances tend to involve relatively static performers (notwithstanding the conductor, perhaps) and very little additional visual material (i.e. beyond the motion involved in playing the instruments or conducting), it is common for performers working within the popular music sphere to incorporate a lot of movement, elaborate set designs, and visual projections into the performance.

There are of course other performance contexts with similar characteristics (e.g. dance, opera etc.), but what these contexts share is a typically passive audience. A tradition involving an active

[8]'Popular music' probably needs a definition here. I use the term not to refer to music which is necessarily popular, but rather to refer to music which is derived from popular sources such as rock and roll, tin pan alley etc. I view this as separate from music made in the Western classical and electroacoustic traditions. A recent example of music derived from popular sources (rock, folk) which is not necessarily popular itself would be the noise music of artists such as Wolf Eyes, Burning Star Core etc. Essentially, by 'popular music', I mean 'Western music which is not part of the classical or electroacoustic, high art tradition'.

audience can, however, be seen in the singing of hymns in church, where each member of the congregation has a part to play in the musical performance. Numerous installations in art galleries could also be said to involve an active audience, making use of sensors or specially-built interface devices to guide sonic developments.

Generalising, there are two settings involved so far; the concert hall, which is designed specifically for a traditional form of musical performance with a passive audience, and the art gallery, which presents a far freer environment, less constrained by musical tradition. The pervasiveness of the internet, however, does present some alternative possibilities. For a start, it enables performers from across the world to perform together, using software such as NINJAM[73]. It also enables the creation of performances in virtual world settings, such as Second Life[22], which has attracted a sizable community of artists since its inception.

5.1.3 The Form of the Instrument

There are a number of factors which will influence the final form of the instrument. As well as the intention of linking sound and visuals in an intuitive, immediate fashion (which will naturally impact on the form of the instrument's output), there is also the desire to create an instrument which is capable of a wide expressive range, and which is enjoyable to play. The following is a list of attributes which may be desirable in the creation of an instrument that fulfils

these criteria.

- **Possibility for exploration:** The possibility for exploration is key in encouraging the performer to continue playing with the instrument. If the instrument's entire expressive range can be immediately grasped, the performer is unlikely to have any motivation to continue performing with it. This also ties in with my Masters research[89], which was concerned with developing music software that stimulates the user's creativity. For that project, encouraging exploration was one of the methods I used in my attempt to stimulate the user's creativity.

- **Unpredictability:** Related to the above, a degree of unpredictability is also desirable, in that it presents a challenge to the performer. It makes the instrument a non-trivial system which must be learned before the performer can gain mastery over it. Unpredictability also raises the possibility of danger - of the instrument suddenly getting too loud, or going entirely silent - which raises the stakes for the performer, and makes it more important (and thus more satisfying) to accurately articulate particular gestures.

- **Possibility of dynamic structural alteration:** This element is somewhat related to both the previous ones. Algorithms for generating sound and visuals tend to have a very strictly-defined range of potential outputs, with the result that an instrument designed around an existing algorithm may prove

quite predictable to a musician already familiar with that algorithm. Instruments which are able to dynamically re-structure themselves, however, add a degree of unpredictability, and create wider scope for alteration. Essentially, the model I am proposing is one consisting of multiple small units with limited ranges of behaviour, which interact with each other in some way. This interaction between limited, discrete units can then give rise to more complex, interesting behaviour.

5.2 Technical Issues

5.2.1 Realising a Fused Audiovisual Connection

I have settled on synchresis as the mechanism by which I will try and link sound and visuals in an instrument. The next question is how best to implement this mechanism. As discussed in Chapter 3, I believe our perception of synchresis is based on our experience of the physical world. When we see objects in motion, part of our brain expects to hear a corresponding sound. In the construction of synchresis, we are essentially trying to mimic the way in which sound and visuals are linked in the physical world, with the difference being that we are extending the possible combinations between sound and visuals. It seems logical to conclude then that we are crossing into the domain of virtual worlds - we want to create something that somewhat resembles the real, physical world,

but which lets us do things not strictly possible in the real world.

At the time of writing, two of the most popular virtual worlds are Second Life and World of Warcraft[43]. Neither environment, however, is particularly suitable for our purposes, as they were not designed to enable musical performance. Their proprietary nature also means they are not easily modifiable to suit our needs. A more suitable environment can be found in the X3D specification[28], an open standard for representing and communicating with 3D scenes and objects. The successor to VRML[40], X3D is designed to cover a huge range of possible applications. Implementing just a small subset of the specification would provide a substantial base for the instrument's development. The specification is somewhat lacking with respect to sound (essentially limited to playback of pre-recorded sound files), but the open nature of the standard means this problem can be easily overcome with simple extensions to the specification.

X3D only requires browsers to implement stereo panning of sounds, but considering the 3D nature of an X3D scene (and of the audiovisual instrument this thesis is concerned with), stereo panning seems somewhat limited. A more flexible and useful solution would be to implement Ambisonic[58] sound spatialisation. With Ambisonics, any sound placed in an X3D scene can have its 3D position encoded together with its output, so that the sound may be placed correctly in the listener's soundfield, regardless of the number, or configuration, of speakers used. Given enough speakers,

it is possible to generate a full 3D soundfield using Ambisonics. Also, because Ambisonic sound is encoded together with its 3D position, there is no need to individually adjust the speaker feeds for each sound source, as is necessary with surround panning when working with 5.1 sound.

5.2.2 Interfacing Performer and Instrument

How the performer interacts with it is a key element in determining an instrument's character, and defining its limitations or constraints. As we are dealing with a computer-based instrument (as opposed to an acoustic one, where the performer is in direct contact with the sound generation mechanism), we can define two categories of physical interface. The first refers to interfaces which are physical objects consisting of various sensors, to detect particular gestures and transmit them to the sound generation device. The second is a more intangible interface, relying on direct transmission of the performer's gestures to the sound generation device without requiring the performer to hold or manipulate any physical object. To the best of my knowledge the earliest example of this kind of interface is the Theremin, with more recent examples making use of technologies such as video tracking.

The advantage of the physical object approach to interface design is that it presents the performer with a specific set of constraints to work within, or against. These constraints make it eas-

ier for the performer to judge their gestures with the interface, as they are working within a set of explicit physical limits. By contrast, the intangible approach presents the performer with no such limits, and offers no guidelines on how to produce a particular gesture accurately. It could perhaps be argued that having a physical interface provides the performer with a more direct connection to the sound generation mechanism, as they are able to hear exactly what the manipulation of one parameter does to the sound. With the intangible approach, it is surely far harder to determine the parameters of the interface, as human gestures are made up of numerous different parameters, any of which could manipulate the sound. In the paper *'Gesture and Musical Interaction: Interactive Engagement Through Dynamic Morphology'*[94], however, Garth Paine argues that such an approach essentially switches the performer's focus away from how to interact with the physical object, and towards their experience of their environment:

> *"The absence of instrument, that is a metal/wood/skin/gut construct, places the instrument, through gesture, in the mind. It is about experience and not the techniques of addressing the object, instrument."*[9]

[9] p.3, [94].

5.2.3 Software Requirements

This section outlines the main requirements of the software environment to be developed for the instrument.

- **Wide range of possible uses:** Beyond just implementing an audiovisual instrument, the software should aim to support as wide a range of uses as possible. While focused on audiovisual performance, the software should, as much as possible, not dictate how the user is supposed to use it. The aim is for the software to be used by a wide range of people, and for it to be capable of being used in ways I had not envisioned. Four basic uses are; *audiovisual performance*, including the possibility of collaborative performance (for which the Open Sound Control[112] (OSC) protocol would be particularly useful); *spatialised (3D) sonic art*, as the environment, including sound, is 3D; *a virtual world*, such as Second Life. This is probably not possible, given the timespan of this PhD; and *3D model viewing*, where the user can view and manipulate a 3D graphical model in realtime.

- **3D Sound Spatialisation:** As discussed earlier, the software should be able to generate a full 3-dimensional soundfield, making use of Ambisonic encoding to provide the most accurate soundfield possible with any given set of speakers.

- **Extendable:** Tying in to the requirement for the software to

support as wide a range of uses as possible, the software should also provide the user with means to extend its capabilities. The X3D specification provides support for scripting which allows for a substantial degree of extension, but the user should also be able to add their own node types as shared libraries. The implementation of shared library support will also mean I can make my audiovisual instrument part of an external library which the user does not have to load if they only require support for the official X3D specification. Implementation of the OSC protocol will also enable users to interface the software with other OSC-capable software and hardware, further extending the software's capabilities.

- **Capable of recording performances:** While it is intended for realtime audiovisual performances, the software should also be able to record such performances to be viewed again at a later date. With the rise of sites like YouTube[45] and Vimeo[25] it has become extremely easy to share videos, and the software should provide some facility to enable the user to make videos of their performances.

- **Scalable to Multi-Processor Architectures:** The development of computer hardware in recent years has been focusing more and more on multiple processor, or multiple core, architectures. The software should therefore look to leverage such architectures' multithreading capabilities in order to make best

use of the hardware.

Chapter 6

The Heilan X3D Software Environment

The Ashitaka audiovisual instrument is a single object within a wider software environment called Heilan. This chapter will describe the environment together with the structure of the underlying code.

6.1 An Introduction to Heilan

Heilan is an X3D browser written in C++ using the SDL[23], TinyXML[24], and PortAudio[19] libraries[1]. X3D[28] is the successor to the VRML[27] file format for modelling 3D graphics, often used to describe so-called 'virtual worlds'. X3D is an XML-based file format (though the specifications do allow for other encodings, such as a backwards-

[1] In addition, at the time of writing SDL_image[78], libsndfile[12], JACK[57], FFTW[5], libjpeg[9] and libvorbisfile[13] are also used.

compatible VRML encoding[2]). An X3D browser is an application which interprets and displays X3D files, and allows the user to interact with the resultant X3D scene to some degree. The X3D specification uses a concept of 'Profiles', which describe a set of rules and functionalities that an X3D browser must support if it is to conform to a particular profile. As such, a browser need not support the entire X3D specification in order to be a compliant browser. At the time of writing Heilan supports the Interchange profile, with the addition of various other nodes such as Sound and AudioClip. A node in X3D parlance refers to an entity or object type which may be inserted into an X3D scene (for example a box, or an image to be used as a texture).

Heilan was originally developed using the X3DToolKit[79] library, which provides the ability to parse, display and transform X3D content. By the end of the first year of my PhD, however, it became clear that the library was no longer actively developed. This factor, together with the difficulty of integrating sound and visuals within the library's code framework, led to a new version of Heilan being developed from scratch, which does not rely on the X3DToolKit. While I do not believe any code was directly taken from the original project, it did obviously influence the current version of Heilan, with the result that there may be certain areas where this influence is visible.

One of the motivations in developing Heilan is the fact that there

[2]Note that Heilan only supports the XML encoding.

were no X3D browsers available which offered a low latency audio engine. Naturally, low latency is a requirement for any musical instrument, so developing Ashitaka in an existing browser would have presented considerable problems. Heilan uses the cross-platform PortAudio[19] API to provide low latency audio on all the platforms it runs on[3].

A second factor which distinguishes Heilan from existing X3D browsers as far as audio is concerned is the use of Ambisonic spatialisation for all sound-producing objects in the environment. Ambisonics is a method of encoding audio so that it may be decoded to almost any number of speakers in almost any configuration, while retaining the three-dimensional (or two-dimensional) spatial information it was encoded with[4]. As such, Heilan is equipped to provide a fully three-dimensional soundfield corresponding to the spatial positions of sound-emitting objects within the software environment.

A further audio-related distinguishing factor is Heilan's support for Open Sound Control (OSC). When this project was begun, the only interactivity X3D browsers offered was in the shape of mouse,

[3]ASIO on Windows, JACK on Linux, and CoreAudio on OSX.

[4]The article *'Spatial Hearing Mechanisms and Sound Reproduction'*[58] presents a good overview and explanation of Ambisonics. Heilan's Ambisonics code is derived from this article. The article *'Surround-sound psychoacoustics'*[67], by the inventor of Ambisonics, Michael Gerzon, discusses the psychoacoustic issues of 2D sound reproduction which Ambisonics attempts to circumvent. The webpage *'First and Second Order Ambisonic Decoding Equations'*[66] lists a number of decoding matrices which can be used with specific loudspeaker configurations. Heilan implements these matrices as speaker configuration presets.

joystick, or in some cases data glove interfaces, primarily controlling avatars within a 3D scene. The ability to directly control aspects of a scene with the degree of control a musician would expect was simply not available. Heilan solves this problem by acting as an OSC server, where any object in a scene may be manipulated via OSC messages once it has been assigned a name via X3D's *DEF* keyword. As such, Heilan can potentially provide a multiple user, collaborative 3D audiovisual environment. It should be noted that since Heilan's first release, the FreeWRL[7] browser has added ReWire[21] support which offers some of the same functionality, but since ReWire is based on MIDI, it does not allow for the same flexibility with regards multiple performers, and the data resolution is extremely limited. The FreeWRL implementation also requires browser-specific X3D nodes which are not compatible with other browsers, while Heilan's implementation is based entirely on existing X3D mechanisms (meaning an X3D file written to take advantage of Heilan's OSC support will still render without errors in other browsers). An additional drawback with regard to the ReWire protocol is that it is a proprietary standard, only available to commercial software developers.

In addition to these features Heilan also offers the option of using a multi-threaded audio engine. This is designed to make more efficient use of computers with multiple processors or multiple cores, and process the audio for separate objects in a 3D scene in separate threads.

Finally, Heilan also supports the use of libraries to extend the application with additional node type definitions, navigation types, and sound file loaders. This means that certain deficiencies in the main program such as not supporting all the nodes defined in the X3D specification may be rectified at a later stage without recompiling the entire program. There is currently one library available for Heilan; libheilanextras, which contains all the extra nodes I've created which are not part of the X3D specification (including two different versions of the Ashitaka instrument/node).

6.2 Code Structure

6.2.1 Overview

Heilan is oriented around the scene graph - the structure which represents the current X3D scene. In Heilan this takes the form of a simple tree structure, analogous to the structure of an X3D file. Every X3D node which Heilan supports has a corresponding C++ class, derived from the AbstractNode base class. In Heilan, a node may represent both a data structure, and a way of transforming, representing, or manipulating data. As such, the AbstractNode base class presents interfaces for accessing such data, and for manipulating it, or outputting it to the screen or the soundcard. This mirrors the X3D specification, in that an X3D node may simultaneously possess information, and represent a way of transforming

it (e.g. the Transform node). Though this approach has certain disadvantages[5] it does mean that the code for a particular node type is kept to a single class, and there is very little communications overhead compared with a design where data is kept separate from functionality. The version of Heilan which used the X3D ToolKit library, for example, had to maintain separate scene graphs for the X3D data, the OpenGL output, and the Sound output. This sometimes resulted in a single node implementation requiring up to six separate C++ classes (the three scene graph nodes, plus 'visitor' classes for each scene graph, used to instantiate and traverse those nodes[6]). There is also a conceptual reason Heilan integrates everything into a single class, and that is this thesis' goal of creating a fused audiovisual material. By having a single class responsible for both sound and visual output I aimed to reduce the structural barriers between the two domains.

6.2.2 The AbstractNode Class

The primary building block for a scene in Heilan is the AbstractNode class. This is the base class for all X3D nodes and provides the interfaces necessary for nodes to be drawn to screen and generate audio. It also provides some implementation methods which are

[5]The AbstractNode class is somewhat monolithic, for example, as it has to provide interfaces for a wide range of possible actions.

[6]The current version of Heilan incorporates this traversal into the AbstractNode class.

used by the majority of the nodes in the program[7].

As far as Heilan is concerned, a node has up to two primary roles - it may be responsible for drawing 3D geometry to screen, and it may generate spatialised audio. Starting with the visuals, AbstractNode provides a very simple interface. All drawing is done in the `graphicDoStuff()` pure virtual method, using the OpenGL API[18]. OpenGL's state machine operation makes the implementation of nodes such as Transform trivial, though it does present a slight problem with respect to Heilan's X3D file parser when, for example, it encounters a file with an Appearance node appearing after the related geometry node. The problem occurs because Heilan's parser is very simple, and creates a scene graph identical to the heirarchy of nodes in the X3D files it reads. As such, an Appearance node would be processed after the associated geometry node, resulting in its OpenGL commands being essentially ignored. To get round this, AbstractNode provides a `shouldBeStart` variable which, if set to `true` by a subclass, is used to indicate to the parser that this node should appear at the top of its parent's list of child nodes. Though not an ideal solution, this mechanism should be enough to prevent the worst problems associated with Heilan's simple parser design. In order to slightly optimise drawing, AbstractNode also provides a `visual` variable which is used by ParentNode to determine whether to call a child's `graphicDoStuff()`

[7]For example, `getAmbisonicAudio()`.

method or not (as some nodes do not provide any visual operations).

Audio in Heilan relies on the `AmbisonicData` structure, which represents a single sample of audio in Ambisonic B format. It has four member variables representing the four Ambisonic components (W, X, Y, and Z), and various convenience methods for adding and subtracting (etc.) two AmbisonicData structures together. All the convenience methods are in the form of overloaded operators, including an assignment operator that takes in a `ThreeFloats` structure (the equivalent of X3D's SFVec3f) representing a position in the 3D scene, and converts it to the appropriate Ambisonic coefficients to place a sound at that position in the soundfield. An example of how this works follows:

```
float monoInputAudio = 1.0f;
ThreeFloats position(2.0f, 1.0f, 0.0f);
AmbisonicData ambiCoeff;
AmbisonicData ambiOutputAudio;

ambiCoeff = position;
ambiOutputAudio = ambiCoeff * monoInputAudio;
```

The last line essentially encodes the `monoInputAudio` audio sample to Ambisonic B format by multiplying it with the Ambisonic coefficients generated by assigning `position` to `ambiCoeff`. As operator overloading is used, all the hard work goes on behind the scenes, keeping the code relatively clean. It should be noted though that AmbisonicData's ThreeFloats assignment operator is relatively

CPU-intensive[8] and it is not advisable to call it for every sample.

AbstractNode provides two main audio generation methods; `getAmbisonicAudio()` and `getMonoAudio()`. For the most part, subclasses will only have to override `getMonoAudio()`, in which a block of audio samples is written to the W components of an AmbisonicData buffer. `getAmbisonicAudio()` is essentially a wrapper round `getMonoAudio()` that takes the monophonic audio generated there and encodes it to Ambisonic B format. To get around the CPU-intensive operations in `AmbisonicData::operator=(const ThreeFloats&)`, `getAmbisonicAudio()` only calculates the coefficients at the node's current position and those at the position it will be by the end of the audio processing block, and uses linear interpolation between those two coefficients during the processing block. Though this will result in some inaccuracies in the soundfield when nodes are in motion, they are unlikely to be noticable. The interpolation also reduces any zipper noise that would be present when a node moves a large distance in a short period of time.

Similar to the visual case, AbstractNode provides an `aural` variable which is used to determine whether to call a node's `getAmbisonicAudio()` method.

All non-abstract subclasses of AbstractNode are required to specify a string-based type, by calling `setType()` in their constructors

[8]It makes use of 3 `sqrtf()`, 2 `sinf()`, 1 `asinf()`, and 1 `cosf()` calls to calculate the coefficients.

[9]. This is so that the X3D file parser can easily match a node specified in an X3D file to the correct AbstractNode subclass.

The X3D specification states that nodes may be assigned a name with the `DEF` keyword. This is used to set up event routing, among other things. Once a node in a scene is assigned a `DEF` name, events may be routed from that node to another named node with the use of a `ROUTE` statement. Most of the event handling code resides in AbstractNode, so that when the file parser comes across a `ROUTE` statement, it calls `addRoute()` on the transmitting node. This informs the transmitting node of three things; the event to be transmitted (`fromField` in an X3D file), the node to send to (`toNode`), and the attribute of the node to send events to (`toField`). The transmitting node then adds this information to a Standard Template Library (STL) multimap which will contain all the routing data it needs to know. To send events, nodes call the `AbstractNode::sendEvent()` method, which will place the event in an STL `queue` until it can be dispatched when `AbstractNode::handleEvents()` is called. `handleEvents()` dispatches all pending output events for its node, and acts on any pending input events. The output events are dispatched via a call to the receiving node's `addEvent()` method. `handleEvents()` is called for all '`ROUTE`d' nodes by the main `Browser` class at the end of each paint cycle. The following describes the process more suc-

[9] i.e. the Box class calls `setType(''Box'');` in its constructor.

cinctly:

On the file parser coming across a ROUTE statement:

1. Call `sceneRoot->findNode()` to find the named transmitting node.
2. Call `sceneRoot->findNode()` to find the named receiving node.
3. If both nodes exist, call `AbstractNode::addRoute()` on the transmitting node.

When a node needs to send an event:

- Call `AbstractNode::sendEvent()` with the value and name of the attribute to be sent. This can happen in any thread, and at any time.

At the end of each paint cycle:

- For every `ROUTE`d node call `AbstractNode::handleEvents()`. This does the following:
    1. Apply any pending input events using the `AbstractNode::setAttribute()` method, clear the input queue.
    2. Dispatch any pending output events from this node by calling `AbstractNode::addEvent()` on the receiving node, clear the output queue.

The `AbstractNode::findNode()` method is essentially used to traverse the scene graph, returning a pointer to the named node if it exists, or 0 otherwise. It is overridden in the `ParentNode` class to enable this traversal.

A final point to note about AbstractNode is that it uses reference counting, to enable users to make use of the X3D `USE` keyword. This keyword is used to enable the use of a single instance of a node in multiple places in the scene. As such, a `USE`d node will have multiple `ParentNode`s, and reference counting is used to avoid deleting it multiple times (as node deletion is handled by a node's parent for all nodes barring the root node of the scene).

6.2.3 ParentNode

The second most important node type in Heilan is the `ParentNode` class. This is an abstract class derived from `AbstractNode` which is subclassed by any node which may contain child nodes.

One of the intentions behind the design of `ParentNode` was that it would keep track of its own child nodes, and that it would provide methods for traversing the scene graph. As such, the `Browser` class would only need to keep track of the single root node in the scene, as all child operations would be called in turn by their parents. The following demonstrates how `ParentNode` overrides `graphicDoStuff()` to draw all its children to screen:

Pseudocode for ParentNode::graphicDoStuff():

1. Call `Parentnode::preChildDraw()` virtual method.
2. For each child node, call its `AbstractNode::graphicDoStuff()` method.
3. Call `Parentnode::postChildDraw()` virtual method.

As can be seen from the above, by calling the scene's root node's[10] `graphicDoStuff()` method, the entire scene is rendered as each `ParentNode` iterates through its children (which may themselves be `ParentNodes`). Because `ParentNode` overrides the `graphicDoStuff()` method, the virtual `preChildDraw()` and `postChildDraw()` methods are provided so that subclasses may still draw to the screen without disturbing the existing iteration code. As such, subclasses of `ParentNode` should not ever override `graphicDoStuff()`[11], but use the provided pre- and post-draw methods.

In a similar fashion, `ParentNode` overrides `AbstractNode`'s `getAmbisonicAudio()` method in order to call its children's `getAmbisonicAudio()` methods. The audio returned by its children is summed together, along with the (Ambisonic-encoded) results from its own `getMonoAudio()` method. Unlike the graphical methods, `ParentNode` does not introduce any extra audio methods, as it is assumed that subclasses will rarely have to override or

[10]Bear in mind the root node of any X3D scene should always be a `Scene` node, which is a subclass of `ParentNode`.

[11]An exception is the `MultiTexture` node.

even interact with `ParentNode`'s `getAmbisonicAudio()` method, instead just doing all their audio calculation in `getMonoAudio()` (which `ParentNode` does not override).

In the case of `preChildDraw()` and `postChildDraw()`, both methods are declared `protected`, in order to prevent them from being called by outside actors. Generally speaking, very few methods in any of Heilan's classes are declared `protected`, and no variables are declared as such, following Scott Meyers' advice in [84].

The `ParentNode` class keeps track of its children in an STL `vector`. As alluded to before, `ParentNode` is responsible for deleting all its children, which is done via `AbstractNode`'s reference counting mechanism. This happens in `ParentNode`'s destructor, meaning that when the root node in a scene is deleted, all the nodes in the scene are subsequently deleted. Child nodes are added to `ParentNode`s by the file parser on reading an X3D file, using `ParentNode`'s `addNode()` method.

6.2.4 Node Construction

To construct nodes, a factory design pattern is used. Every node class has a static `NodeConstructor` member which is used to construct and return an instance of the node. `NodeConstructor` is a template class derived from the pure virtual class `NodeConstructorBase`. The following demonstrates how it would be implemented in an `AbstractNode` subclass `ExampleNode`:

```
class ExampleNode : public AbstractNode
{
  public:
    ...
    static NodeConstructor<ExampleNode> constructor;
};
```

There is a `NodeFactory` singleton class which keeps track of all the supported node types and their associated `NodeConstructor` members (which it stores internally as an STL `map` of `NodeConstructorBase`s). The X3D file parser then calls the `NodeFactory` instance whenever it comes across a node in a file, in order to construct that node via its `NodeConstructor`. Before any files are parsed, all the supported nodes must first have their `NodeConstructor`s registered with `NodeFactory` via its `registerNode()` method. This approach was derived from the final example given in [4].

6.2.5 Threads

Heilan runs four main threads; a graphics thread, an audio thread, an idle thread and an OSC thread. The graphics thread is the primary thread, and is responsible for drawing the scene to the screen and handling any keyboard and mouse input, as well as certain system events (such as the window being resized). The audio thread is responsible for generating audio from nodes and sending it to the soundcard. As Heilan is concerned with low latency audio operation, this thread is high priority and its function

is largely determined by the soundcard, which frequently requests more audio to output. The idle thread is used primarily to allow for streaming certain file types from disk, such as audio and video files. These files types are streamed from disk rather than read entirely into RAM because the files used may be of any size and thus could easily consume the entirety of the user's available memory if read from in this way. The streaming is done in a separate thread because (particularly in the audio thread) file operations could lead to glitches due to the relatively slow speed of disk reads (compared to that of RAM reads). The idle thread spends most of its time asleep, waking up every 62ms[12] to call the scene's root node's `idle()` method (which in turn calls the `idle()` method of every node in the scene). The OSC thread is used to listen for OSC messages on the user's defined port, and pass them on to the relevant nodes. Like the idle thread, the OSC thread spends most of its time asleep, only waking up when it receives an OSC message (or bundle). Generally speaking, any operation that requires cross-thread communication is protected by a mutex.

As mentioned previously, in addition to the four main threads, Heilan has the ability to split off the generation of audio by nodes into separate threads. This ability is built into the `ParentNode` class, so that any node derived from `ParentNode` is capable of running its childrens' audio processing in parallel. This function is en-

[12]62ms being 1/4 of the time it should take before a single audio block in the AudioClip node needs to be reloaded with the next block of data from the audio file.

abled by giving the desired `ParentNode` a `MetadataString` child with a `name` of *heilan_parallelChildren*, and a `value` of *'TRUE'*. The following X3D file thus processes the audio from two `Sound` nodes in parallel:

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE X3D>
<X3D profile='Full'>
  <Scene>
    <MetadataString name='heilan_parallelChildren' value='TRUE'/>

    <NavigationInfo type='ANY'/>

    <Transform translation='-4 0 -8'>
      <Sound>
        <AudioClip url='leftSound.wav'/>
      </Sound>
    </Transform>

    <Transform translation='4 0 -8'>
      <Sound>
        <AudioClip url='rightSound.wav'/>
      </Sound>
    </Transform>
  </Scene>
</X3D>
```

To accomplish this, the required `ParentNode` constructs the requisite number of threads (which immediately sleep until further notice) when their X3D file is parsed. In its `getAmbisonicAudio()` method, `ParentNode` then signals the child threads via a semaphore that they should process a block of audio. On completion of this task, each thread returns the generated audio to the `ParentNode`, which blocks the audio thread (via a conditional variable) until all

of its children have returned. The reasoning behind parallel audio processing as opposed to parallel graphical processing is simply that modern graphics programming is almost entirely done on the graphics card and thus parallelising the visual operations is unlikely to provide any significant performance benefit. Audio processing, however, (particularly sound synthesis algorithms) can be very CPU intensive, and thus should benefit most from being split into separate threads on multi-processor/multi-core machines.

6.2.6 Support for Shared Libraries

As mentioned previously, Heilan is capable of loading extra node type definitions from shared libraries. The interface for this is very simple; a Heilan library simply has to implement a `libraryInitialise()` function. In this function, any new node types (or navigation types, or sound file loaders) register themselves with the `NodeFactory` singleton, so that they may be instantiated in a scene. The following code demonstrates a `libraryInitialise()` function for a library with a single node type definition:

```
#include "NodeFactory.h"

#include "Example.h"

using namespace HeilanX3D;

extern "C"
{

void libraryInitialise()
{
    NodeFactory::getInstance().registerNode(&Example::constructor);
}

}
```

Chapter 7

Experimental Audiovisual Instruments

In the process of developing Ashitaka, and the Audiovisual Parameters mapping strategy, I developed eight smaller, experimental audiovisual instruments. This chapter presents each instrument, in chronological order, together with a (subjective) examination of what their successes and failures are. During the development process I rated the instruments out of ten in a number of categories, as a way of quantifying their relative successes and failures and identifying possible areas for further exploration. Though the marking scheme itself was essentially constructed from scratch, the categories are largely derived from ideas elaborated upon in papers such as Sergi Jordá's *'Digital Instruments and Players: Part 1 - Efficiency and Apprenticeship'*[74] and David Wessel and Matthew Wright's *'Problems and Prospects for Intimate Musical Control of*

Computers'[110].

In attempting to define what makes a successful musical instrument, these papers see the key factors as; being able to *"strike the right balance between challenge, frustration and boredom"*[74] for a novice performer, providing rich experiences and encouraging exploration and creativity, and predictability. The quote *"low entry fee with no ceiling on virtuosity"* appears in both papers in reference to the authors' desire for instruments which are easy to pick up and play with no prior experience, but which also provide sufficient scope for a performer to really explore the instrument and gain mastery over it. From these principles then, my own categories attempt to mark out the areas I believe an audiovisual instrument should attempt to excel in:

- **Visual and Aural Expressiveness:** These categories are intended to define how wide a range of expression is possible with the instrument in both the visual and aural domains. This is related to the impulse to provide rich experiences to the performer, and encourage them to explore the instrument further.

- **Apparent Strength of Audiovisual Connection:** (in terms of both the audience and the performer) This is the only category that does not have a counterpart in the aforementioned papers, as it is specific to audiovisual instruments and the aims of this thesis. Separate marks are given to both the audience

and the performer as the performer may often feel a stronger connection than the audience does, due to their active role in the process.

- **Amount of Effort Required:** Refers to the amount of effort the performer has to exert to maintain the instrument's output. This relates back to the idea that the instrument should present a challenge to the performer, in order to encourage them to keep playing and become more proficient.
- **Fun to Play:** How enjoyable the instrument is to perform with, again related to the desire to encourage the performer to explore the instrument further.
- **Reproducability:** How easy it is to reproduce a particular phrase or gesture with the instrument. Similar to the aforementioned papers' predictability.

For all categories except Effort, a higher number is better, with a middling score probably being the most useful for the Effort category.

7.1 Yakul

- **Visual Expressiveness:** 4
- **Aural Expressiveness:** 3
- **Apparent Strength of Audiovisual Connection (audience):** 6

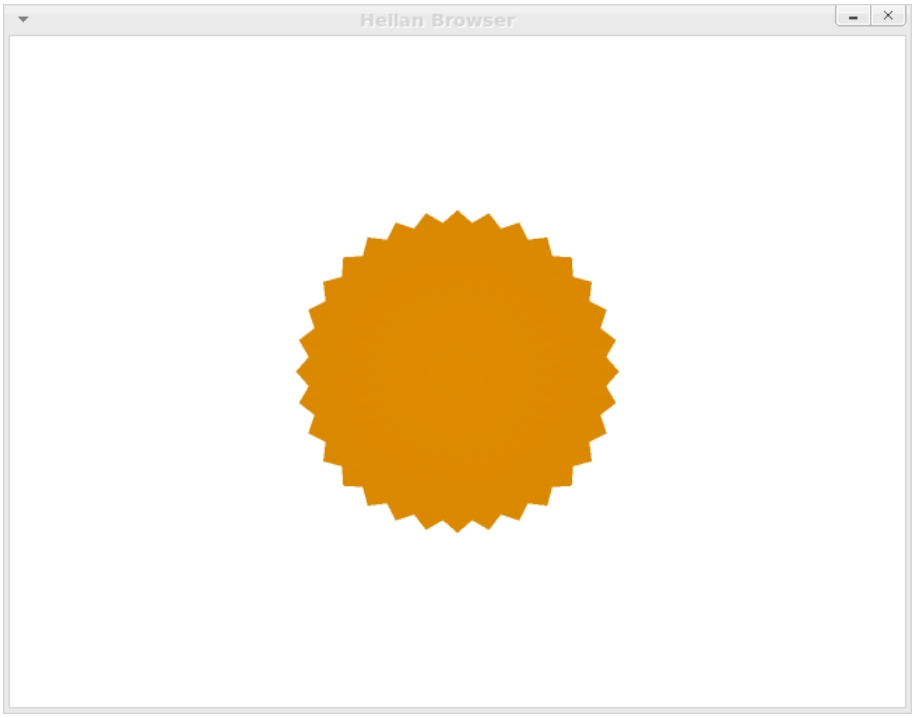


Figure 7.1: Yakul Experimental Instrument

- **Apparent Strength of Audiovisual Connection (performer):** 7
- **Amount of Effort Required:** 4
- **Fun to Play:** 4
- **Reproducability:** 9

Yakul was the first instrument developed for the project, and as such is the simplest instrument examined here. A single mouse-controlled on-screen slider acts as the input interface, with its rate of change controlling a simple visual shape, and an additive synthesizer for the audio. The visuals consist of an orange circle which morphs to a jagged star shape when the input slider has a high rate of change. The slider's rate of change also controls which partials

of the additive synthesizer are audible, with a low rate of change corresponding to the low partials being most prominent, and a high rate corresponding to the high partials being most prominent. When the slider is static, the circle gradually decreases in radius until it disappears, and the audio's amplitude gradually decreases until it is inaudible.

No formal structure concerning the relationship of sound, visuals and the performer's input was defined before developing Yakul. The main impulse behind its design was to try and represent rate of change in an intuitive manner. As such, a high rate of change corresponds to a sharp, high frequency output, in both visuals and audio. This lead to a strong perceivable audiovisual connection for both audience and performer, though the limited range of expression in both the audio and visual domains (both essentially only have a single input) presents the performer with little incentive to continue playing the instrument. The interface of a single GUI slider, however, made for an interesting control method, as it was discovered that, in order to maintain a constant, low rate of change, the best results were actually obtained by moving the mouse in a circle, rather than the linear motion such a GUI element would suggest. As such, the instrument required greater effort to control satisfactorily than originally intended, and some degree of skill was required to maintain a smooth, rounded circle on screen (with the associated low partials in the audio domain).

7.2 Kodama

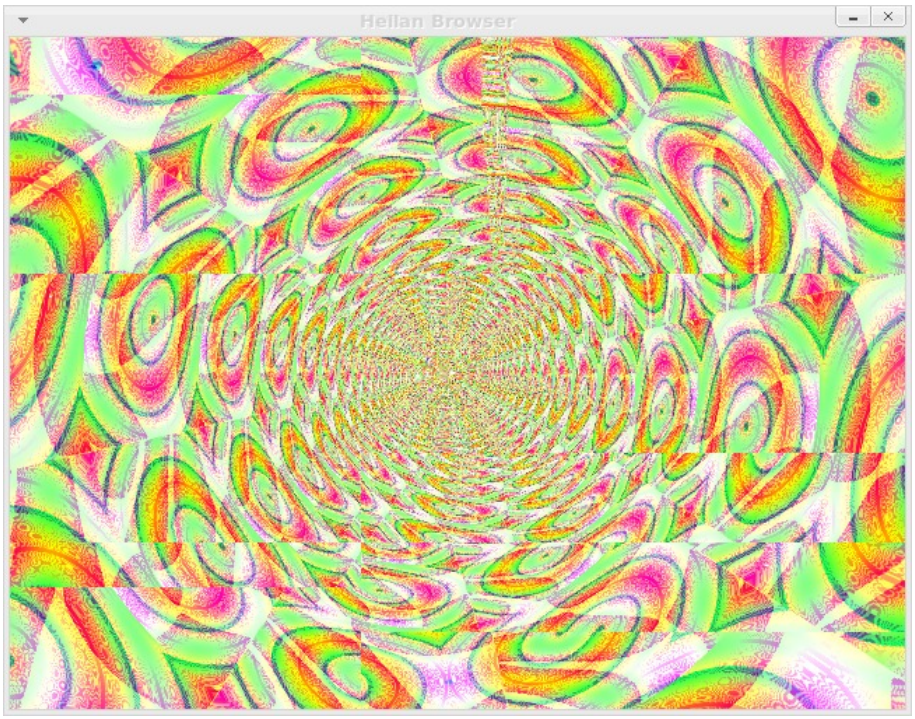


Figure 7.2: Kodama Experimental Instrument

- **Visual Expressiveness:** 4
- **Aural Expressiveness:** 2
- **Apparent Strength of Audiovisual Connection (audience):** 5
- **Apparent Strength of Audiovisual Connection (performer):** 5
- **Amount of Effort Required:** 4
- **Fun to Play:** 2
- **Reproducability:** 9

Inspired by [86], Kodama uses a simple physically-modelled string as an intermediary mapping layer. Both visuals and sound are determined by the positions of individual cells on the string, which is controlled by the performer. The visuals mimic the experience of travelling through a tunnel - the camera is placed at the larger end of a hollow cone, looking towards the cone's point. The walls of the cone are texture mapped to give the appearance of motion (i.e. the textures are constantly panned towards the viewer). The cone is essentially made up of 128 roughly cylindrical sections, each of whose (horizontal) position is controlled by the matching cell on the physically-modelled string (which itself has 128 cells). Audio is generated by an additive synthesizer with 128 partials, each of whose amplitude is controlled by the corresponding cell on the physically-modelled string. The user has one main control - again, a mouse-controlled GUI slider, with the rate of change being the main parameter. When the performer keeps a constant rate of change, the physically-modelled string is bowed, but when a rate of change higher than a specified threshold is detected, the string is plucked and the bowing ceases until the performer resumes a more or less constant rate of change.

While Kodama initially appears quite complex visually, this is primarily a result of the textures used and their spiral panning motion, neither of which the performer has any control over. When the string model is significantly excited, it produces quite violent, almost stroboscopic visuals, though this effect is not really mir-

rored in the audio (though indeed, it's not clear what the aural equivalent would be). When controlled by a physically-modelled string in this manner, the additive synth produces very harsh, almost granular sounds, lacking in nuance or any real expressive range. I believe Kodama fails for two main reasons; the performer's lack of control over the visual output, and a poor choice of mappings between the physically-modelled string and audio and visual synthesizers (particularly the additive synthesizer).

7.3 Nago

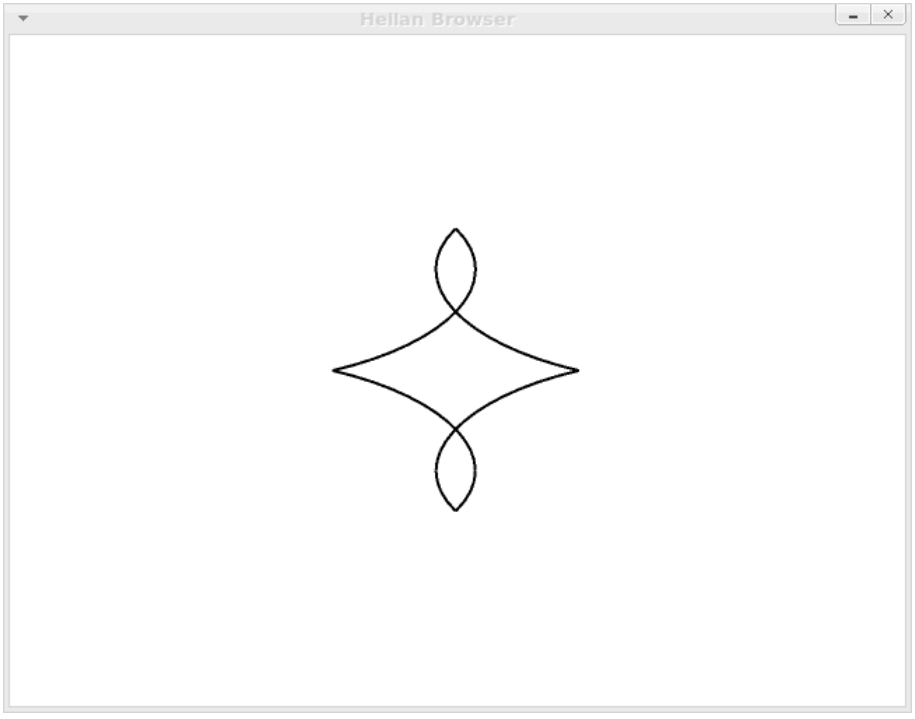


Figure 7.3: Nago Experimental Instrument

- **Visual Expressiveness:** 4
- **Aural Expressiveness:** 5
- **Apparent Strength of Audiovisual Connection (audience):** 4
- **Apparent Strength of Audiovisual Connection (performer):** 6
- **Amount of Effort Required:** 4
- **Fun to Play:** 5
- **Reproducability:** 7

Nago is the first of three instruments which examine my 'Motion as the Connection' hypothesis using the same sound and visual generation methods, and just varying the mappings used. In this case the performer only has direct control over the instrument's audio output. The visuals are then controlled by certain output parameters derived from the audio. The audio synthesizer is a single physically-modelled string, using code derived from the Tao[95] project. The visuals consist of four bezier curves, connected so that they may form a circle when their control points are positioned correctly. The interface to the instrument is a mouse-controlled x-y pad. The x-axis of the pad controls a number of parameters on the string; the degree to which the string is bowed or plucked (again, this is a rate of change parameter - constant rate of change contributes to the amount of bowing, sudden movements pluck the string), and (to a lesser degree) the position of a stop on the string. The y-axis also controls the position of the stop, but to a greater degree than the x-axis.

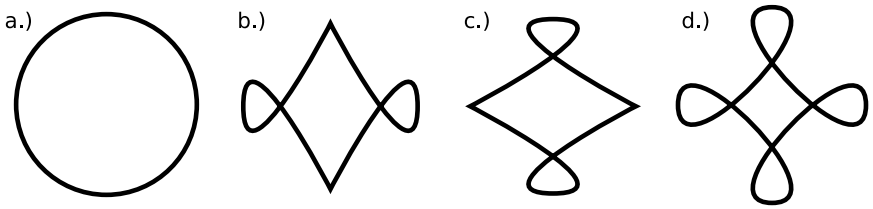


Figure 7.4: Nago Primary Shapes

Figure 7.4 shows the four main shapes which the Nago instrument's curves may form. The amplitude of the audio output controls the radius of the curves from the centre of the screen (low amplitude = small radius...), and the control points of the four curves, which move from shape a.) to b.) with b.) corresponding to a high amplitude. The spectral centroid of the audio also contributes to these control points. The pitch of the audio manipulates the control points towards shape c.). Shape d.) represents the other shape possible when shapes b.) and c.) are combined.

Visually, the way the four curves are manipulated means that Nago has a very limited range of expression. The fact that the performer does not have direct control over the visuals heightens this deficiency, though a more complex method of generating the visuals would likely make for a far more interesting instrument. Aurally, the instrument is more successful, with the two methods of excitation providing a good range of sounds. Due to a bug in the code, the stop does not work as expected and tends to produce a buzzing sound when changed. Additionally, when the stop is not at the end of the string, a high-pitched ringing is produced. While

strictly speaking this is an error, it makes for more complex and unexpected interactions than would be expected with a correctly-functioning string model, and was left 'broken' as a result.

7.4 Okkoto

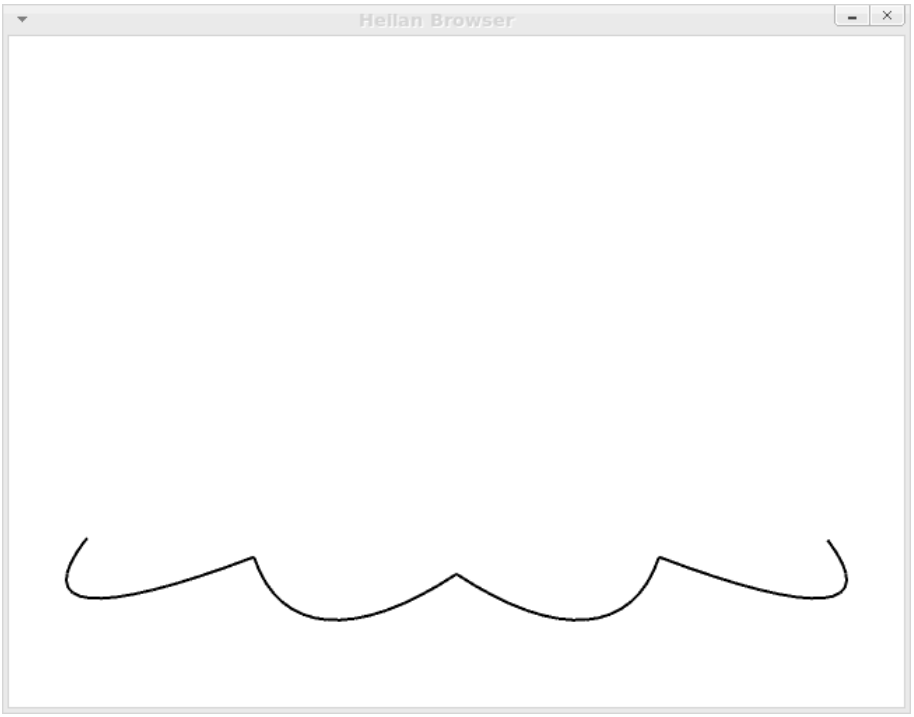


Figure 7.5: Okkoto Experimental Instrument

- **Visual Expressiveness:** 6
- **Aural Expressiveness:** 5.5
- **Apparent Strength of Audiovisual Connection (audience):** 5
- **Apparent Strength of Audiovisual Connection (performer):** 6

- **Amount of Effort Required:** 5
- **Fun to Play:** 7
- **Reproducability:** 7

Okkoto uses the same methods of sound and visual generation as Nago. The interface for this instrument however consists of two sliders on a MIDI fader box. The audio generation in this case is controlled by output parameters determined from the instrument's visuals generator. When the performer is not putting any input into the instrument, the four bezier curves are arranged as a straight line along the lower part of the screen. The average rate of change of slider 1 controls a continuum between this straight line arrangement and a circular-style arrangement, where the curves are arranged end-to-end. This is also affected to a small degree by the average rate of change of slider 2. The difference between the two sliders' rates of change controls the distance between the beziers' control points and their start and end points, while slider 2's position controls how smooth or jagged the beziers' curves are. For the audio the distance between the first and last bezier points controls the stop position. The distance between the beziers' control points and their start and end points controls the bowing frequency, and the inverse of the bowing amplitude. The string is plucked when transients are detected on the continuum of how smooth or jagged the beziers are.

As an instrument, Okkoto was the most successful of the experiments up to this point, in terms of its expressive range and

how enjoyable it is to play. Much of the enjoyment is derived from how the interface, visuals and audio combine with regards to a specific motion. When the first slider is kept moving rapidly, the beziers move to close the 'circle' and the bowing of the string produces a sustained tone. The amount of effort required to do this is relatively high and thus it is extremely satisfying when a(n approximated) circle and sustained tone is achieved. The connection made between a sustained tone and closed shape also seems to be quite apt. The rest of the instrument is not as successful, however. When slider 1 is held stationary and slider 2 is moved in a gentle motion, a satisfying wave-style motion is generated in the curves, though the accompanying audio does not appear to be at all related. As such, this motion is unlikely to be as satisfying from the auience' point of view as it is from the performer's. In general Okkoto is more successful from the performer's point of view than from that of the audience. While the expressive range of the instrument is relatively large, the audiovisual connection has suffered somewhat. Part of the reason for this may be that the performer only has direct control over the visuals, but it could also be argued that the more complex an instrument gets, the harder it is to maintain a robust, intuitive audiovisual connection.

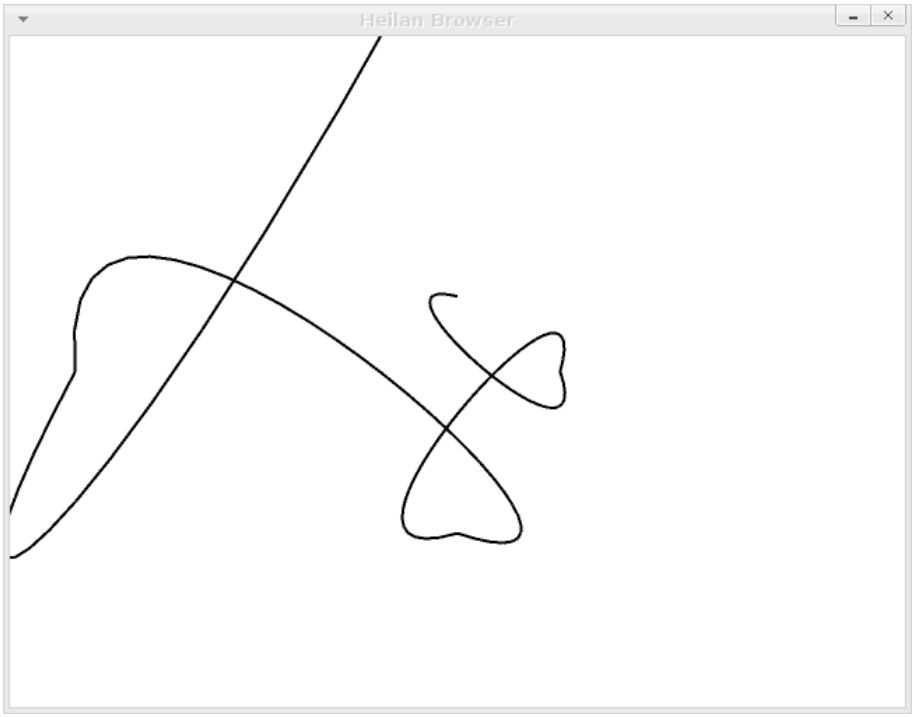


Figure 7.6: Moro Experimental Instrument

7.5 Moro

- **Visual Expressiveness:** 6
- **Aural Expressiveness:** 5
- **Apparent Strength of Audiovisual Connection (audience):** 4
- **Apparent Strength of Audiovisual Connection (performer):** 5
- **Amount of Effort Required:** 2
- **Fun to Play:** 5
- **Reproducability:** 7

Moro uses the same four bezier curves and physically-modelled string generators as Nago and Okkoto, again controlled by two

MIDI sliders. In this case, the performer has direct control over both sound and visuals, with mappings also existing between the two domains. Whereas the previous two instruments essentially treated the bezier curves as 2-dimensional, with Moro the curves also move along the z-axis. Instead of plucking and bowing the string, Moro's string model is excited by a sine tone, and it is the first instrument to do away with the rate of change control method. This was to try and accommodate a more natural way of using the MIDI sliders (as opposed to frantically running them back and forward), and to investigate a more 'linear' method of control. Moro has the most complex set of mappings up to this point, with every audio and visual parameter controlled by two other parameters (in ratios of either 50%:50% or 25%:75%), and there are no one-to-one mappings. There are two performer to visuals mappings, two performer to audio mappings, six audio to visuals mappings, and four visuals to audio mappings.

The decision to go with a linear control scheme possibly hindered this instrument. Despite the large number of mappings, the linear nature of the controls gives the impression that the sliders are mapped to sound and visuals in a one-to-one fashion. The instrument also suffers from a lack of motion in both sound and visuals, with the performer often resorting to 'wobbling' a slider to obtain a more interesting and controllable effect. The large number of mappings between sound and visuals also resulted in a kind of self-reinforcing feedback developing between the two domains, with

the result that even if the performer applied no input, the instrument would still output sound and manipulate the visuals. For the most part the instrument is very static and lacking in expressive range and as a result is not particularly successful.

7.6 Koroku

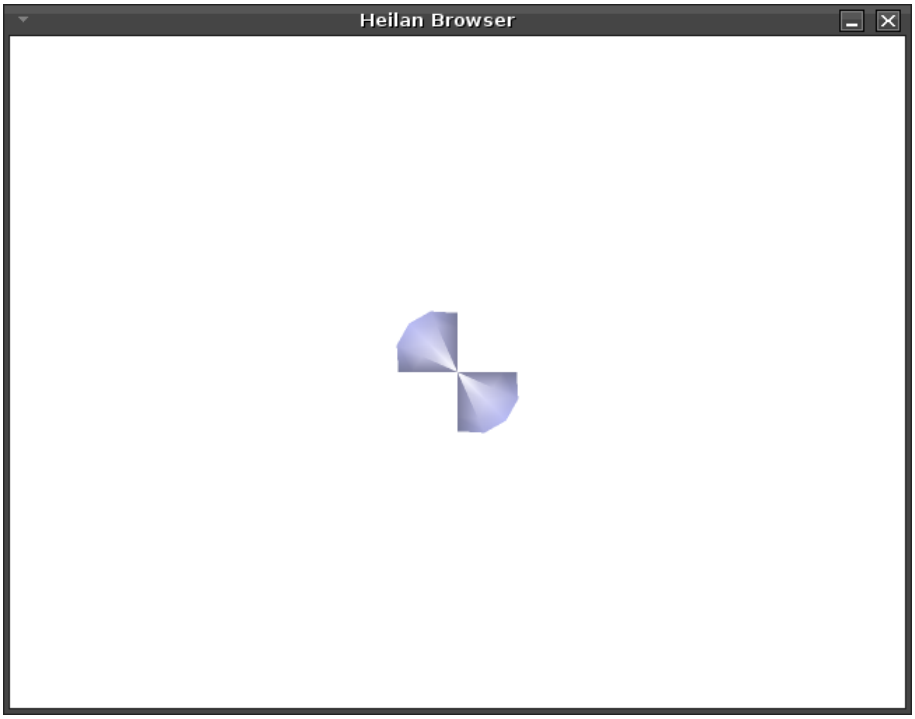


Figure 7.7: Koroku Experimental Instrument

- **Visual Expressiveness:** 2
- **Aural Expressiveness:** 2
- **Apparent Strength of Audiovisual Connection (audience):** 5

- **Apparent Strength of Audiovisual Connection (performer):** 3
- **Amount of Effort Required:** 2
- **Fun to Play:** 1
- **Reproducability:** 1

Koroku is the first of three instruments which uses 3D NURBS (Non Uniform Rational B-Splines) patches as its method of generating visuals. NURBS were chosen because they allow the user to define shapes in significant detail using a relatively small number of input parameters. The control points (and colours) for these patches are interpolated over a number of set positions, to give a wide range of possible forms. Audio is again generated by a Tao-derived string model, this time excited using Tao's own bowing model rather than the more simplistic methods used previously. One of the aims with this instrument was to create a stronger audiovisual connection than present in the previous three (bezier curve, physical model) instruments. To do this, a mapping scheme was chosen whereby audio and visuals are mapped together via one-to-one mappings, while the performer to instrument mappings are all one-to-many, many-to-one or many-to-many. The thinking behind this scheme was that sound and visuals would be closely linked while the performer would interact with the instrument in a more sophisticated, hopefully more rewarding, way. The input interface in this case is four MIDI sliders (1 rate-of-change, 3 linear position), with the rate-of-change input returning as it was determined to be a more satisfying method of control than a simple

linear position-based control. In addition, an attempt was made to introduce stateful behaviour to this instrument - past a certain energy threshold on the rate-of-change slider, the instrument would switch to a different kind of behaviour. For the reasons presented in the following paragraph, however, this stateful behaviour is essentially non-existent. This instrument was also intended to build on the kind of self-reinforcing feedback present in Moro to create an instrument with some kind of life of its own.

Koroku demonstrates the problems involved in creating a complex mapping scheme where multiple domains feed back to each other. In this case, the audiovisual mappings which were supposed to create a system of self-reinforcing feedback simply cancel each other out, and effectively counteract any gesture the performer makes. As such, the instrument as it stands is all but completely unplayable. Left to its own devices it produces fairly static audio and visual outputs, and the performer's input has very little effect. The use of NURBS patches does provide for a wide range of visual expression, but in this case the visuals are constrained to an essentially static form, as seen in Figure 7.7.

7.7 Toki

- **Visual Expressiveness:** 8
- **Aural Expressiveness:** 6

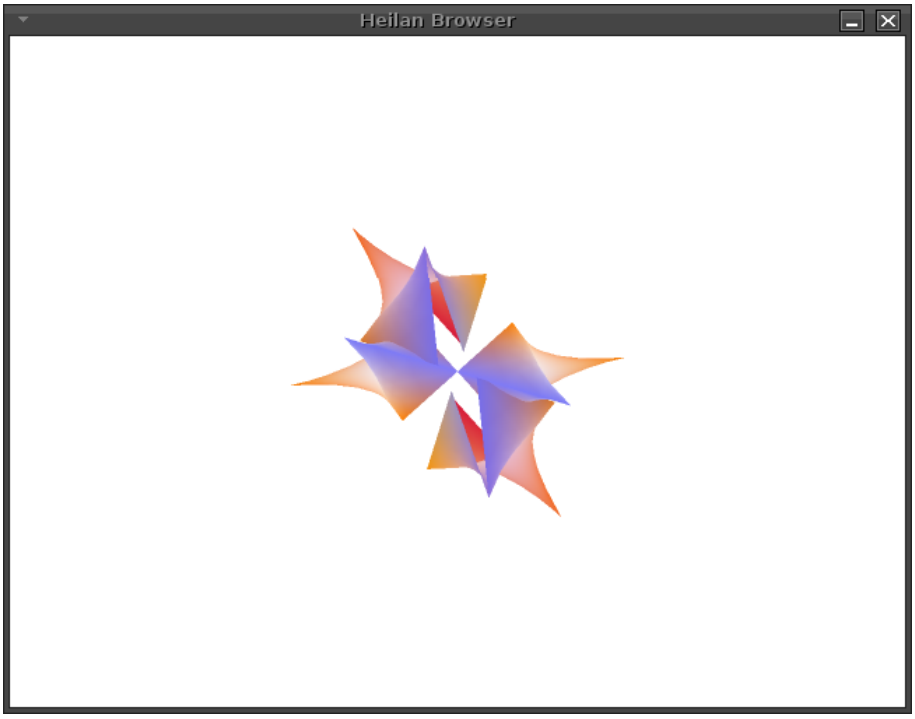


Figure 7.8: Toki Experimental Instrument

- **Apparent Strength of Audiovisual Connection (audience):** 3
- **Apparent Strength of Audiovisual Connection (performer):** 4
- **Amount of Effort Required:** 6
- **Fun to Play:** 6
- **Reproducability:** 6

Toki represents the genesis of the Audiovisual Parameters mapping strategy. After Koroku's failure, I decided a strategy was needed which would prevent such a situation from arising in the first place. As such, five audio and five visual parameters were chosen, and linked to each other. In this case any audiovisual mappings have very limited influence, and sound and visuals are

intended to be primarily linked by virtue of their shared input. The linked a/v parameters are:

Table 7.1: Toki Audiovisual Parameters

Visual	Aural
Scale (size)	Bow force
Colours interpolation position	String frequency
Rotation speed	Feedback amount
Symmetry	Damping amount
Control points interpolation position	Damping position

This approach greatly reduces the number of mappings involved (13 as opposed to 27) and means that the main focus of the mapping work is the mappings between the performer's input and the five audiovisual parameters outlined. In all other respects, the instrument is the same as Koroku, using the same methods of visual and sound generation, and the same input scheme.

As an instrument, Toki is far more successful than Koroku, with a wide range of possible visual expression (though a slightly narrower range in the audio domain). The audiovisual connection is relatively weak however. The reason for this probably the use of low level parameters in the construction of the audiovisual parameters. Taking the Scale/Bow Force parameter as an example, we can see that when Bow Force is decreased to 0, it is possible that the underlying string model will still be resonating. The visuals however will be static at this point, as they are linked to the method of excitation, something which is perceptually harder to discern than the related amplitude of the audio output. It would thus appear that

a perceptual approach to the creation of audiovisual parameters would present a more readily discernable audiovisual connection. The other observation made with this instrument is that the string model could perhaps do with wider range of expression, with percussive sounds in particular being noticable by their absence.

7.8 Jiko

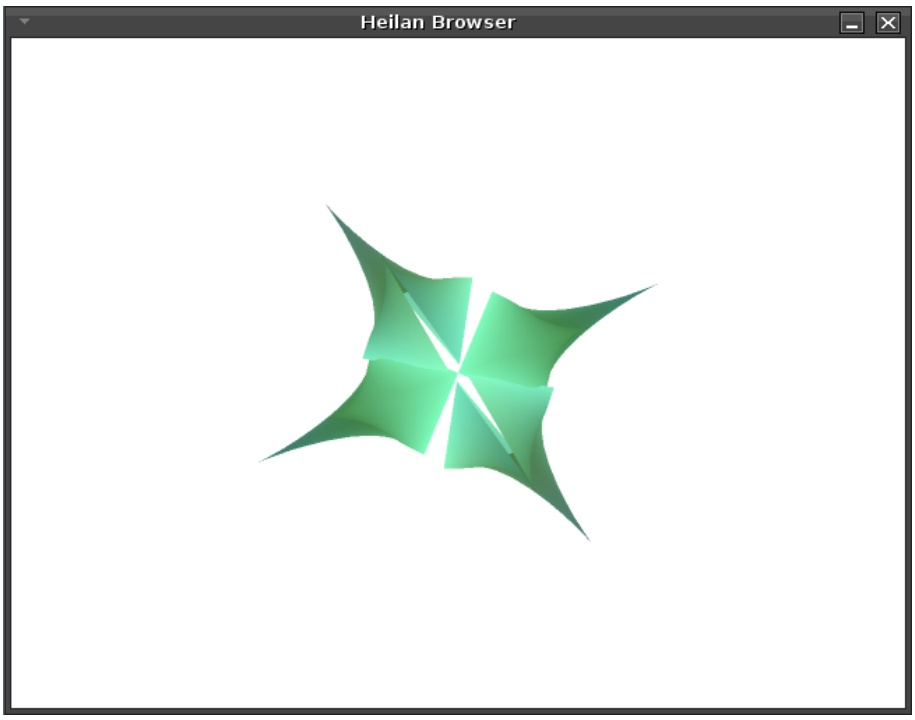


Figure 7.9: Jiko Experimental Instrument

- **Visual Expressiveness:** 4
- **Aural Expressiveness:** 6

- **Apparent Strength of Audiovisual Connection (audience):** 5
- **Apparent Strength of Audiovisual Connection (performer):** 6
- **Amount of Effort Required:** 8
- **Fun to Play:** 6
- **Reproducability:** 7

Jiko is the final experimental instrument developed before I started work on Ashitaka. Sound and visuals are again the same as in the previous two instruments, with three instead of four MIDI sliders used as input (one rate-of-change, two simultaneously rate-of-change and linear position). In this case, the mapping scheme uses perceptual instead of low level parameters in the creation of combined audiovisual parameters. The four audiovisual parameters are:

Table 7.2: Jiko Audiovisual Parameters

Visual	Aural
Size	Energy content
'Pointy-ness'	Pitch
Rotation	Instantaneous energy content
Colour temperature	Decay time

Energy content refers to the perceived energy content present in the sound, and is equivalent to the sound's Root Mean Squared amplitude. Instantaneous energy content refers to sudden amplitude changes in the sound, and is equivalent to the sound's instantaneous amplitude. Colour temperature essentially refers to a continuum between 'cool' colours (blues, pale greens) and 'warm' colours (reds, oranges). Each of these perceptual parameters are

mapped to multiple parameters in the two output generators, and each parameter (barring Colour temperature/Decay time) has some mapping between the audio and visual generators (though these mappings are exclusively one-way, to avoid problems with feedback).

Jiko is probably the most successful of the instruments outlined here. As an instrument it produces a nice warbling vibrato sound, due to the third slider controlling both bow force (rate-of-change) and pitch (linear position). The performer also has a lot of control over the sound, with a slider to control pitch and another controlling the amount of damping. This is significant because the string model can go out of control if pushed too hard, and the damping essentially provides the performer with the ability to bring it back under control. Having said this, the audiovisual connection is still somewhat lacking, something which can perhaps be seen just by looking at Table 6.2. Only the Size/Energy content parameter really has much precedence in the physical world, with the others being somewhat arbitrary. In retrospect, it would perhaps make more sense to match 'Pointy-ness' to Instantaneous energy, and Pitch would perhaps be better matched to Size, though there are not necessarily obvious counterparts to some of the other parameters. Having said this, the approach taken to develop these mappings definitely seems to have the most promise of all the instruments developed here.

7.9 Other Instruments

Though developed separately to my PhD, there are two other instruments which I believe deserve a quick mention here.

7.9.1 Particle Fountain

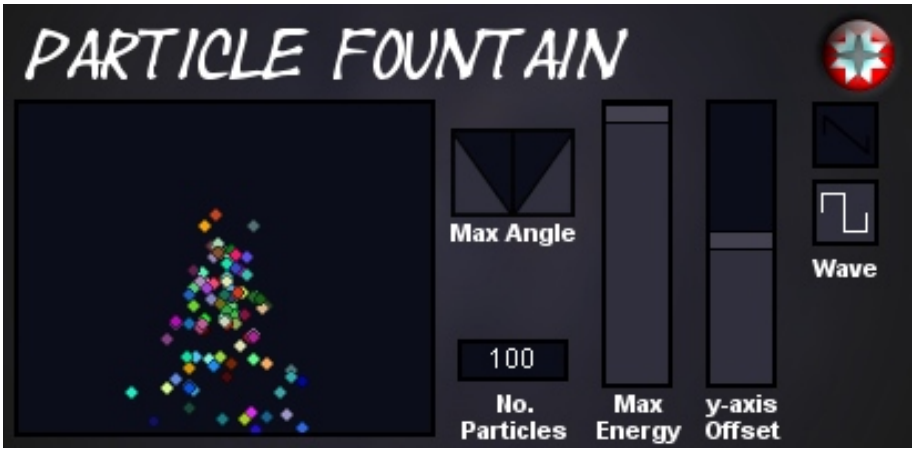


Figure 7.10: Particle Foutain VST Instrument

Particle Fountain[88] is the initial inspiration behind this PhD, originally developed as a simple experiment in linking the motion of a simple (visual) particle fountain to a sound generator. The instrument is a monophonic (monophonic in the sense that only one note can be played at a time) VST[39] plugin controlled via MIDI. When the performer sends the plugin a MIDI note on message, the particle fountain (visible in the plugin's interface as seen above) starts generating particles, and stops when a note off message is received. Each particle controls a simplistic oscillator with three

parameters. The y-axis position of the particle controls the oscillator's amplitude. The x-axis position controls both the degree to which the oscillator is panned left or right (as with almost all VST plugins, the instrument has a stereo audio output), and a pitch offset. The pitch of the oscillator is initially set by the note value of the MIDI message sent to the plugin, with an offset added depending on how far the particle is from the centre of its window (with a negative offset applied when it's to the left of the centre, and a positive offset applied when it's to the right). As such, the sound is essentially controlled by a visual algorithm, which aids the performer in understanding how the instrument works, and generates a relatively complex sound.

Being based on this instrument therefore, my initial proposal for this PhD was aimed at investigating how a visual output may aid the performer in playing an instrument. The more I read on the subject however, the more interested I became in developing an instrument which had a visual output which was of equal importance to its audio output, hence the current direction.

7.9.2 Brush Strokes

Brush Strokes[87] was developed parallel to my PhD in 2007 and, similar to Particle Fountain, was initially a quick sketch based on an idea I had. The instrument is again a monophonic VST plugin, this time with two oscillators which are controlled by 'drawing'

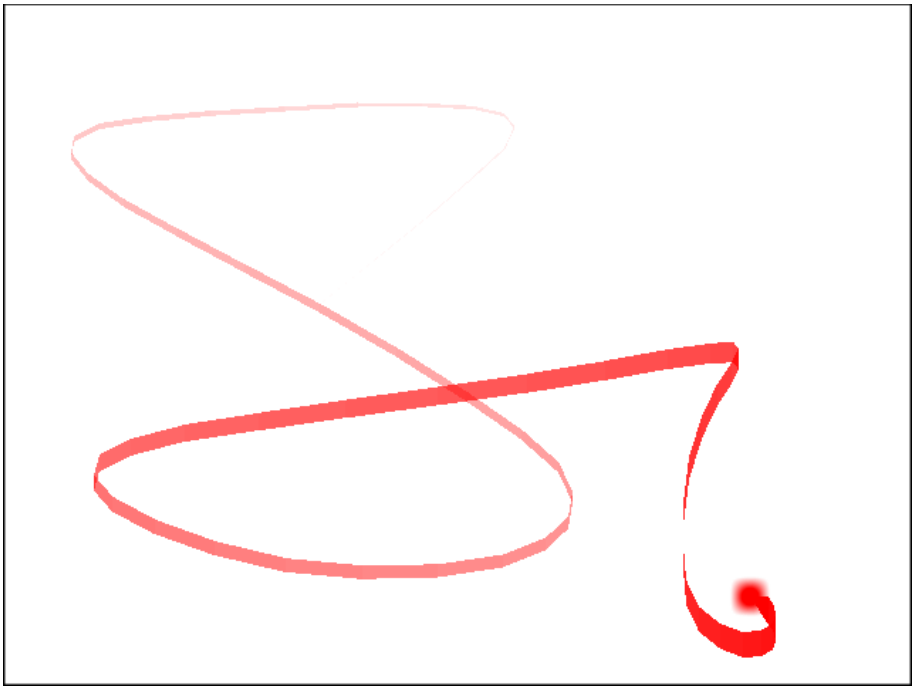


Figure 7.11: Brush Strokes VST Instrument

with the mouse in the plugin's GUI. Visually, any movements the performer makes over the GUI with the mouse are recorded and visualised as a red trail that follows the mouse, gradually decaying away. Sonically, the instrument resembles a form of scanned synthesis, with two wavetable oscillators whose contents are determined by the visual trail's 2 dimensional points. One oscillator determines its wavetable from the trail's x-axis, and the other from its y-axis. As such, if the performer were to draw a horizontal line from right to left, a sawtooth waveform would be generated by the x-axis oscillator, while the y-axis oscillator would be silent. Similarly, drawing a perfect circle would generate two identical sine

tones. Because the trail decays away when the mouse is static, no sound is generated. As such there is a very tight connection between the performer's gestures and the sound and visual output of the instrument.

The immediate connection between the performer's gestures and the instrument's audiovisual output makes Brush Strokes - despite being little more than a sketch - one of the most successful audiovisual instruments I have developed. In addition, the way that the instrument's visual motion is retained (in the form of the trail) makes this motion easier to comprehend. As the human eye is less sensitive to motion than the ear, rapid visual motion can often prove disorientating, but the visualisation of the instrument's visual motion in this manner somewhat counteracts this potential sense of disorientation.

Chapter 8

The Ashitaka Instrument

The main practical focus of this PhD was the Ashitaka audiovisual instrument. The instrument was designed to act as a kind of self-contained ecosystem which the performer can influence and interact with. This ecosystem essentially consists of two types of objects (Quads and GravityObjects) which will be explained in detail later in the chapter. In addition, a physical interface was designed and built for the instrument utilising bluetooth and Open Sound Control technologies.

8.1 The Interface

8.1.1 Hardware

Though it has since been developed in a different direction, Ashitaka was originally conceived as a musical block of clay, to be performed

with using the same gestures a block of clay affords (i.e. stretching, twisting, squeezing...). As such, the interface is designed to be performed using these types of gesture. It has seven sensors; a track potentiometer providing data about lengthwise, stretching gestures, a rotary potentiometer to handle rotational, twisting gestures, four force sensors (the Honeywell FSG15N1A[8]) for squeezing or sculpting gestures, and a 3D accelerometer to provide information about the interface's motion within the performance space. Figure 8.1 shows how these sensors are positioned on the interface.

The other two key components to the interface are a bluetooth module (the Free2Move F2M03AC2 module[6]) to transmit data to a computer, and a PIC microcontroller (part no. PIC16F874A[15]) to arbitrate between the sensors and the bluetooth module. The two potentiometers and four force sensors are all analogue components and so interface with the microcontroller via its onboard 10-bit Analogue to Digital Converter (ADC). The accelerometer (a Kionix KXP74-1050[11]) incorporates a 12-bit ADC on its chip, with an SPI[36] digital interface. This interface is then connected to the SPI interface on the microcontroller.

The bluetooth specification enumerates various different operating profiles which may be used for data transmission. The module in the interface is configured to use the Serial Port Profile[2]), where it is treated as a standard (RS232) serial port. It is interfaced to the microcontroller using the microcontroller's UART interface.

The bluetooth module is set to transmit data at its fastest possible speed, 57.6 kbaud.

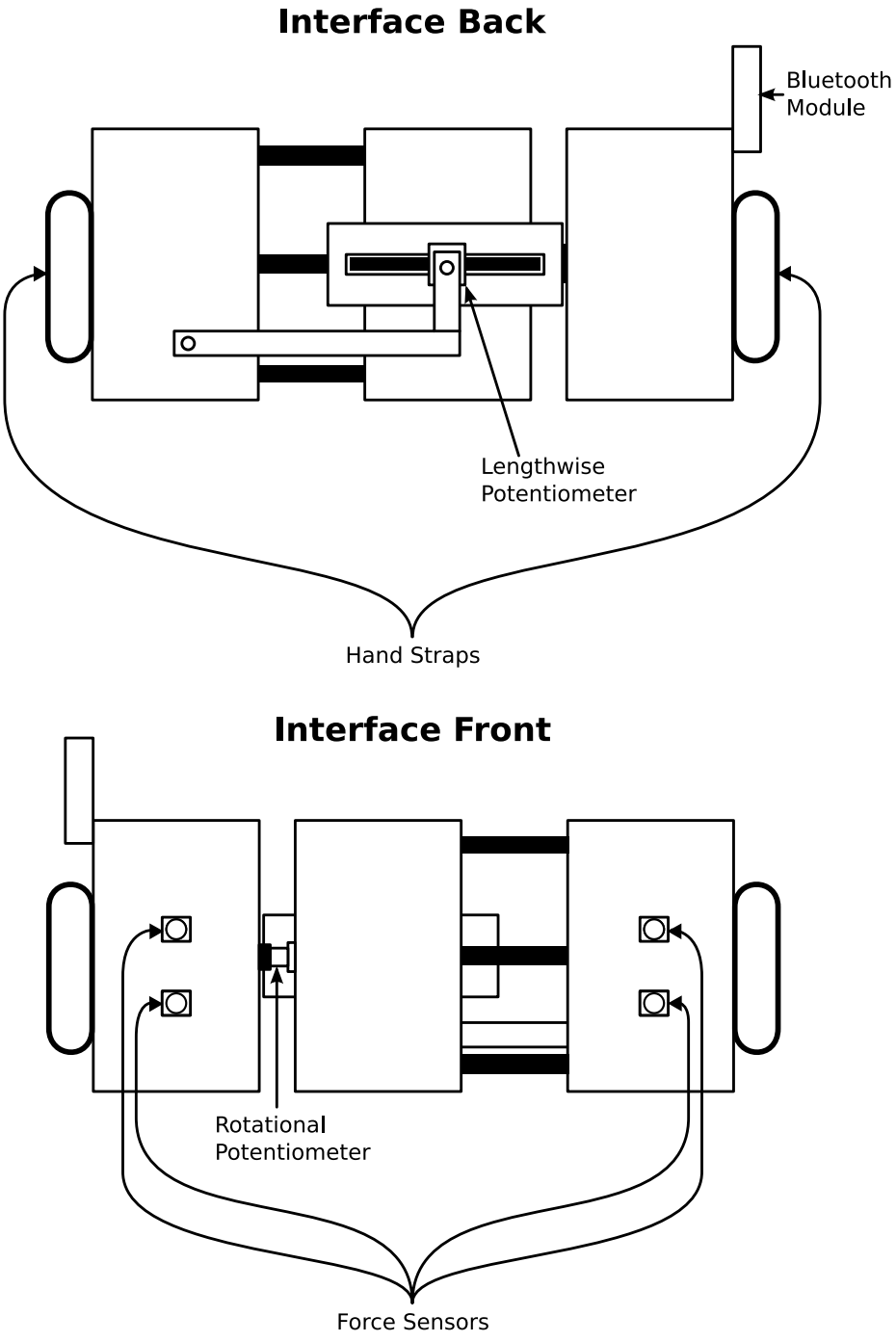


Figure 8.1: Interface Outline

8.1.2 Software (PIC)

A small program runs on the microcontroller which essentially polls the nine inputs (with the three axes of the accelerometer treated as three inputs) and passes the data on to the bluetooth module as fast as possible. A single data sample from an input is represented as two bytes, with the redundant bits (either 6 or 4 bits, depending on the ADC) ignored. The data is sent to the bluetooth module (and from there to the receiving computer) as raw bytes, with a program on the computer converting these bytes to Open Sound Control (OSC) messages to be sent on to Heilan.

In order for the program on the receiving computer to differentiate between the otherwise anonymous stream of bytes it receives, a single 0xFF byte is sent at at the start of each loop through the nine inputs. If one of the bytes obtained from the inputs has a value of 0xFF, then a second 0xFF byte is sent after that byte has been sent, in order for the receiving program to differentiate between a start byte and a data byte.

8.1.3 Software (PC)

A simple program was written to convert the raw data received from the interface into OSC messages to be sent on to Heilan. The justification for not building this functionality directly into Heilan is that the interface produces a very specific pattern of data, and the code to decode it is not easily generalised - a fact that is somewhat

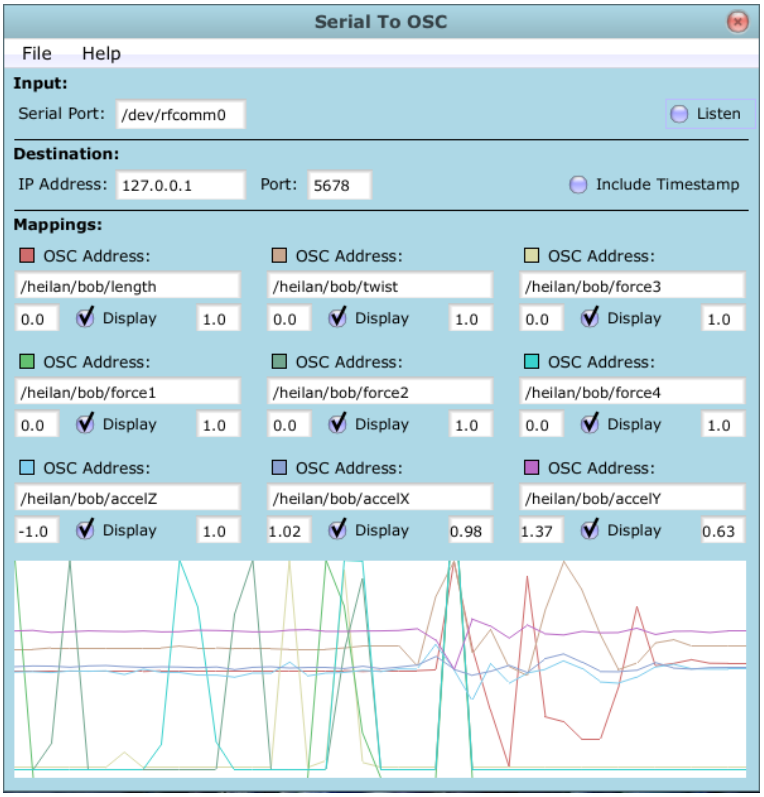


Figure 8.2: Serial To OSC program

at odds with the Heilan's aim to provide a general purpose X3D browser. Splitting the decoding code off like this also means the interface can be used with other OSC-aware applications and it is therefore not tied to Heilan.

As the interface transmits data using the Serial Port Profile, accessing the data on a PC is done with the operating system's standard serial port APIs. The bluetooth dongle on the PC is just treated as an extra serial port, and no bluetooth-specific programming is required. The Serial To OSC program lets the user determine a separate OSC address for each input to send to, as well as the range

each input's data is mapped to. The data from the inputs is sent as floating point values, with the values from all nine inputs sent in a single OSC bundle. This bundle gets transmitted whenever the Serial To OSC program has received a single set of new values for all nine inputs. For debugging purposes, a simple oscilloscope-type view of the received data is included (as can be seen at the bottom of Figure 8.2).

8.1.4 Timings

In *Latency Tolerance for Gesture Controlled Continuous Sound Instrument without Tactile Feedback*[81], Teemu Mäki-Patola and Perttu Hämäläinen present a good overview of the issue of how latency can impact on musical performance. The authors quote the figure of 10ms as being the generally accepted maximum latency acceptable in a musical instrument, but also note that performers' tolerance of latency depends on the instrument and kind of music played. As such, they claim that higher latencies may be acceptable, dependent on context. For the purposes of this thesis however, we will aim for a latency $\leq$10ms.

If we then assume that the latency of the soundcard is 5ms[1], the latency of the phsyical interface should also be no more than 5ms.

There follows a rough estimate of the time taken by the software

[1]This seems a reasonable assumption, given that modern soundcards are capable of latencies of 3ms or less.

running on the PIC to obtain readings from all nine sensors and transmit the data to the bluetooth module:

- No. operations in main loop (estimate): 575
- Time taken by 575 operations given a 20MHz clock: (575*4)/20000000 = 115μs
- PIC ADC acquisition + conversion time: 14.8μs (*6 = 88.8μs)
- Accelerometer ADC acquisition + conversion time: 50μs (*3 = 150μs)
- PIC SPI transmission time: 1.6μs (*9 = 14.4μs)
- Total time: (115 + 88.8 + 150 + 14.4)μs = 408.2μs

From this estimate it is clear we are well within our 5ms goal, but we have not yet accounted for the time taken to transmit our data over the bluetooth connection to the PC. Indeed, the bluetooth connection is the real bottleneck in this system, given that it is restricted to operating at 57.6kbaud. Remembering that we transmit a duplicate 0xFF byte for every 0xFF the PIC generates, we may send anything between 19 and 28 bytes for a single data packet. The following therefore lists the *ideal* maximum latencies incurred by the bluetooth connection:

- 57.6kbaud, 1 byte: 141.1μs
- 19 bytes: 2.68ms

- 28 bytes: 3.95ms

We are now a lot closer to our 10ms limit, but (even in the worst case) still within it. Measurements taken at the PC confirm this. For a sample of 2100 data packets, the average latency between packets is 3.01ms.

8.2 The Ashitaka Ecosystem

Contrary to traditional instrument designs which are more or less monolithic, Ashitaka is designed more as a kind of simplified ecosystem than a single, coherent 'object'. The reason for this is partly that an ecosystem design arguably offers a wider range of visual material than a single object would[2]. It also offers the possibility of unexpected interactions arising between objects, and potentially makes it possible to create a complex system from relatively simple components. Visually it somewhat resembles the particle systems often used in graphics programming to produce such effects as smoke and flames, while aurally it fuses fairly simplistic physical models with what may be considered granular methods of excitation (essentially derived from the particle-based behaviour of the visuals).

There are essentially two reasons for describing the instrument as an ecosystem. Firstly, it refers to the way the instrument will

[2]At least in Heilan, where such an object would be a single 3D model, only capable of changing its shape and colour/texture.

act with or without the performer's intervention. The instrument is made up of a number of individual objects or agents, which have their own set of behaviours and act and interact with each other accordingly. The intention is to give birth to the kind of emergent behaviour which is visible in any ecosystem. Secondly, it refers to the role of the performer as an agent within a wider environment. In order to try and provide a richer experience with the instrument, the performer is never allowed complete control over the ecosystem. They can manipulate and direct certain aspects of the system, but the other agents within the ecosystem always act according to their set behaviour, which will tend to override any direct instructions the performer tries to give them.

The ecosystem consists of two classes of objects; Quads (essentially particles) and GravityObjects (which could perhaps be considered as particle attractors). An explanation of these two classes follows.

8.2.1 Quads

A Quad is a purely visual object, and thus is the main visual stimulus in Ashitaka. A single instance of Ashitaka consists of a number of Quads (128 at time of writing) and - initially at least - a single GravityObject. A Quad has very little behaviour of its own, and is primarily manipulated by GravityObjects, which define how it moves and what colour it is. Visually, a Quad is simply a some-

what transparent cube, with a line trail indicating where it has previously been. Figure 8.3 demonstrates this.

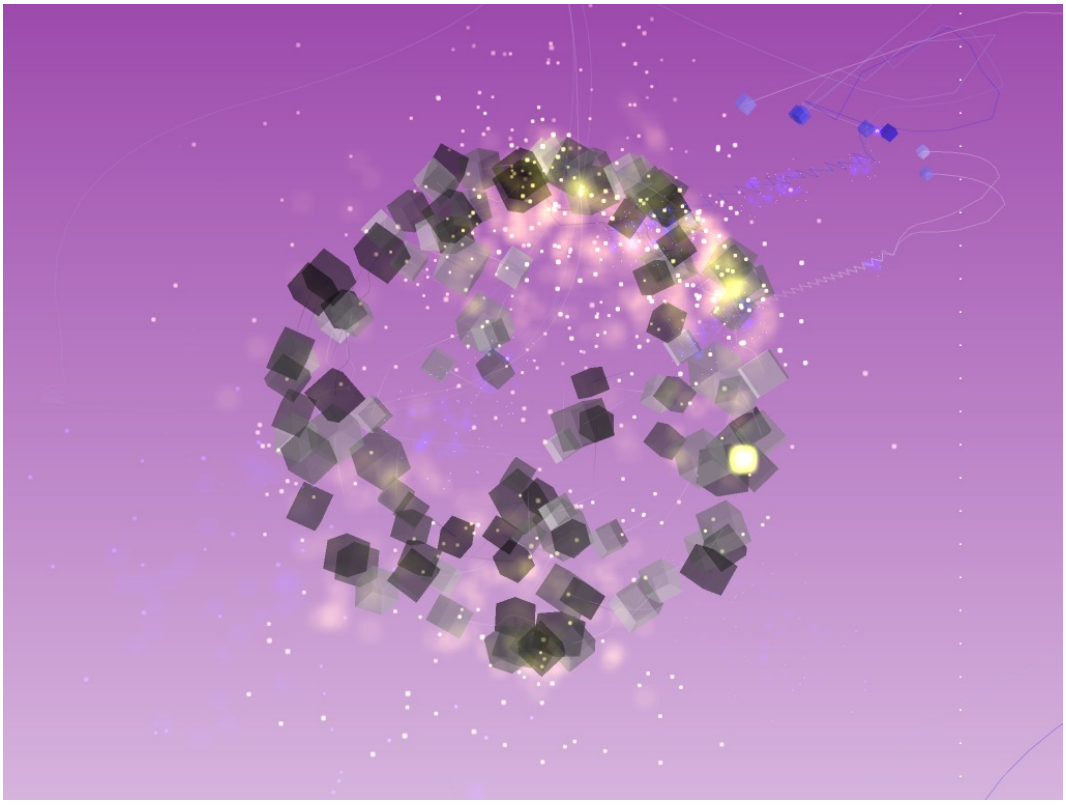


Figure 8.3: Ashitaka static, with one Flocking GravityObject split off

In addition to the Quads and their trails, Figure 8.3 also demonstrates the yellow and blue particle 'explosions' which are generated when Quads collide. When Quads collide, one of these particle explosions is generated, and the two Quads' trajectories are altered. Essentially the two Quads simply swap directions and velocities. Though this approach is simplistic and would not be mistaken for a realistic collision between two physical objects, it

was deemed to be satisfactory for Ashitaka's purposes, given that Ashitaka does not strictly resemble a real-world environment. Also, the main intention behind the inclusion of collisions was to provide discrete temporal events which are easily perceivable to both audience and performer, not to provide a realistic simulated physical environment.

Each Quad is assigned a random 'brightness' value in addition to its GravityObject-derived colour. A brightness of 1 would mean the Quad tends to white, while 0 would mean it tends to its assigned colour. This is to aid visual differentiation between Quads and avoid large blocks of static colour when a number of Quads (owned by the same GravityObject) are on screen.

8.2.2 GravityObjects

A GravityObject is a primarily aural object with no direct visual output, though it has an indirect visual output via its control over the motion and colour of its Quads. GravityObjects are assigned a number of Quads which are their primary responsibility. They also have a sphere of influence, and if the Quads from another GravityObject stray within that sphere, their motion and colour will also be affected (to a lesser degree) by the original GravityObject. This is of course in addition to the influence of their own primary GravityObject.

GravityObjects use a simple physically-modelled string as their

audio synthesizer. Each different type of GravityObject has a different length of string (and thus different pitch), and a different method of exciting the string. In addition, any time two Quads collide, the strings of their primary GravityObjects are plucked. The aim of this approach is to create a soundworld which is capable of generating a wide range of sounds, but is still recognisably coherent, since all the sounds which may be made are generated in the same fundamental way. Figure 8.4 shows a basic diagram of the string model.

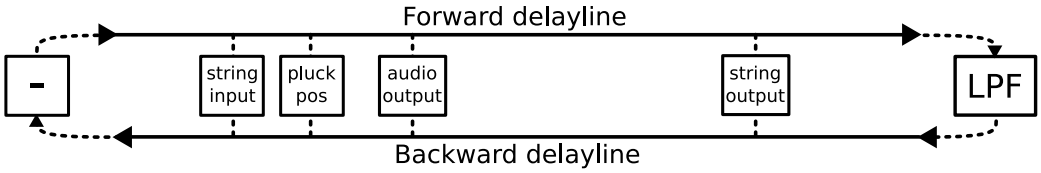


Figure 8.4: The basic string model used throughout Ashitaka

As can be seen in Figure 8.4, the string model is a simple 2-delayline design, based on a model in Julius O. Smith III's *'Physical Audio Signal Processing'*[3]. The two delaylines travel in opposite directions, and are connected to each other at both ends. At one end the value at the end of the backward delayline is inverted and passed onto the start of the forward delayline. At the other end, the value at the end of the forward delayline is passed through a low pass filter onto the start of the backward delayline. The audio output is obtained by summing the values of both delaylines

[3][102], section *Elementary String Instruments*, page *Rigid Terminations*, Figure 4.2: `http://ccrma.stanford.edu/~jos/pasp/Rigid_Terminations.html`

at the *'audio output'* position marked in the figure. The point at which the GravityObjects excite the string is not marked because it may vary at the GravityObjects' whim. Apart from the varying methods of excitation, the only departure from a traditional physically-modelled string algorithm is the application of a tanh-based soft-clipping function to the audio outputs. This is simply to keep outputs of the string within reasonable limits, as the system can be somewhat unpredictable, and may easily become unstable under certain conditions if no limiting is applied.

The other significant feature of Ashitaka's audio engine is that, when two GravityObjects' spheres of influence overlap, their strings become connected. The result is that sympathetic resonances can be set up, further extending the range of possible sounds. As can be seen in Figure 8.4, there are two set positions on the string, from where audio is sampled to be sent onto another string (or strings), and where audio taken from another string (or strings) can be used to excite the string. In order to avoid excessive feedback, the connection between two strings can only ever be a one way connection. Having said this, however, there is no limit to the number of strings one string may be connected to, and the system allows for circular connections, as shown in Figure 8.5. This makes possible a degree of indirect feedback and the creation of complex resonances as the multiple strings simultaneously vibrate in sympathy, while at the same time introducing new excitations into the chain. With enough GravityObjects in close proximity it should be possible to

create even more complex arrangements of strings, one possibility being the development of multiple, chained figure-eight patterns. Any connections between strings are (at the time of writing) visualised by a thick red line drawn between the two GravityObjects.

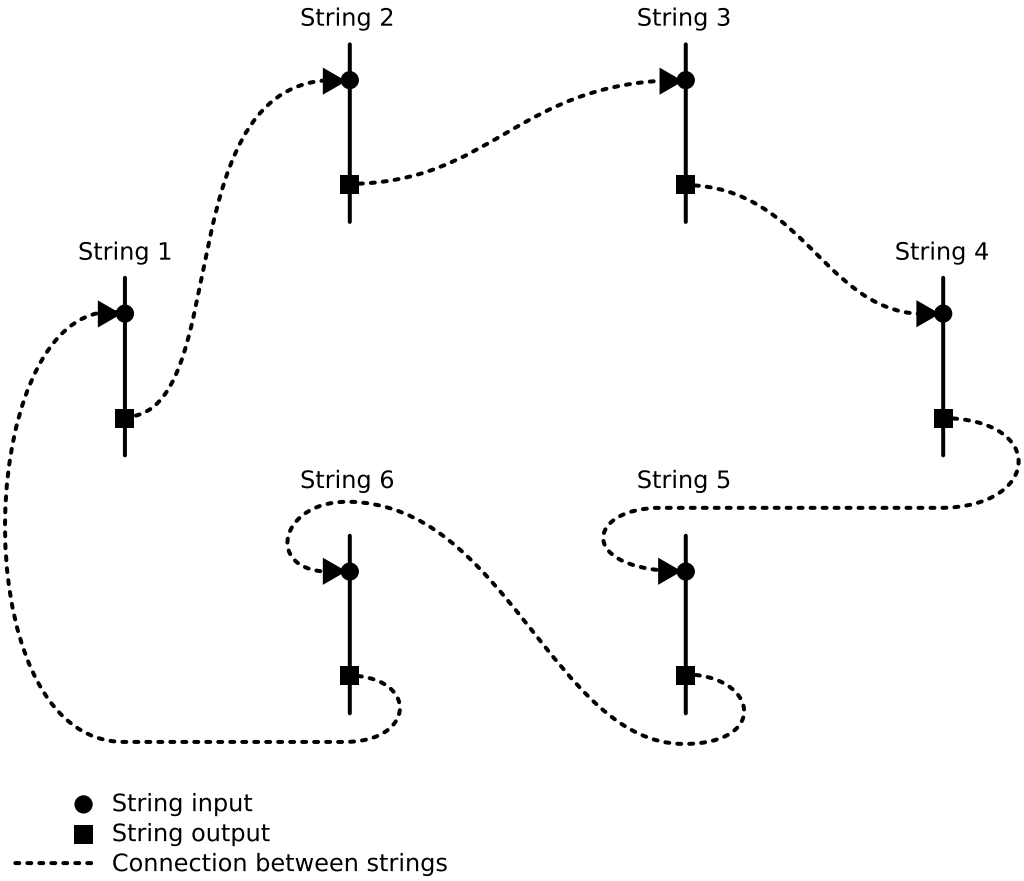


Figure 8.5: Six GravityObject strings connected in a circle

To interact with the Ashitaka ecosystem, the performer has direct control over only a single GravityObject, with the ability to move it within the 3D space, and manipulate the behaviour of its Quads. The performer also, however, has the ability to create new

GravityObjects and 'split off' Quads from the main GravityObject to be assigned to these new objects. These other GravityObjects essentially exist autonomously and behave according to their own specific set of rules. As can be inferred from the above however, the performer is able to indirectly interact with them by moving the main GravityObject within their spheres of influence, setting up connections between strings, and influencing and colliding other Quads. This ability to create and interact with a complex ecosystem which can never be entirely controlled by the performer is the key feature in my attempt to create an instrument which is complex and capable of a degree of unpredictability sufficient to entice the performer to keep playing, and exploring it.

All of the 'secondary' GravityObjects (i.e. those that are not the main, performer-controlled GravityObject) also possess a life force, which decreases over time. When a GravityObject's life force reaches zero, that GravityObject is considered 'dead', and its Quads are released and assigned to the nearest 'live' GravityObject. Again, this is to try and create a complex and dynamic environment for the performer to interact with.

Ashitaka uses the gesture recognition method described in Chapter 4.3 as a way of triggering the creation of new GravityObjects. Only the output from the interface's accelerometer is used for this, as any gestures made for its benefit will be quite large and clearly visible to the audience. This visibility then helps the audience see the link between the performer's gestures and their audiovi-

sual consequences. A particular gesture is associated with each of the three secondary GravityObjects, and when the gesture detector recognises the performer 'drawing' one of these gestures, the corresponding GravityObject is created.

8.2.3 Types of GravityObject

Planet

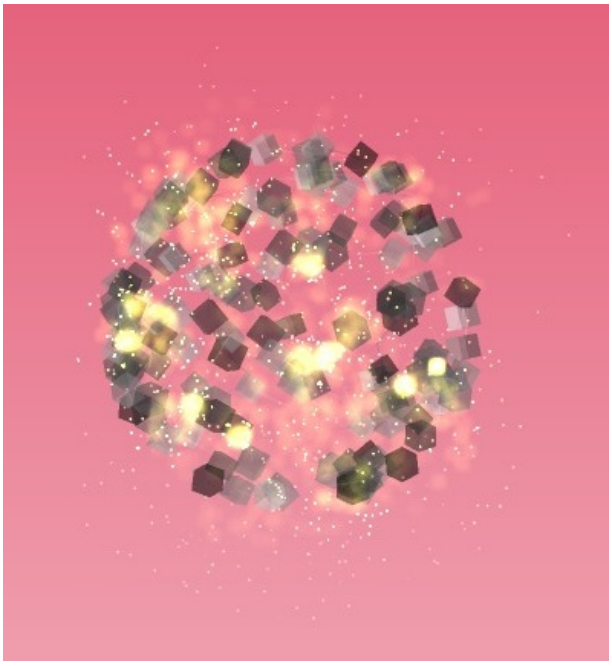


Figure 8.6: Planet GravityObject type

The Planet GravityObject type is unique in that there will only ever be one of them in the ecosystem, and it is directly controlled by the performer. The initial inspiration for this GravityObject was to roughly emulate the motion of clouds around a planet, hence

the name. As such, the GravityObject's Quads are made to continually move across the surface of a sphere, centred around the GravityObject's 3D position within the ecosystem. As can be seen in Figure 8.6, the Quads' close proximity to each other results in a more or less continuous stream of collisions.

Because the Planet GravityObject is controlled by the performer, it is designed around the functionality of Ashitaka's hardware interface. As such, its Quads are divided into four groups, each controlled by one of the four force sensors on the interface. The amount of force applied to one of the sensors determines the degree to which the attached group of Quads gravitates to the surface of the Planet's sphere. When the maximum amount of force is applied, the Quads follow the sphere's surface closely. As the amount of force decreases, however, the 'pull' of the sphere decreases, and the Quads tend towards a simplistic momentum-based motion. i.e. if no force is applied, the Quads will travel along the direction they are already headed with their current velocity, which gradually decreases (though never to absolute zero).

Each type of GravityObject has a different length of string, in order to establish some distinguishing aural characteristics. The Planet GravityObject has a string made up of 128 cells. In addition, the Planet GravityObject utilises four separate exciters to excite its string, mapped in the same way as the four groups of Quads to the interface's four force sensors. The amplitude of the exciters' outputs is determined by the amount of force applied to the cor-

responding force sensor. The audio generated by the exciters is essentially a (relatively rapidly) granulated sine wave, oscillating at a specified pitch. Each exciter's position on the string is determined by the rotation around the sphere of the first Quad in the group[4].

Of the other sensors on the interface, the lengthwise ('stretch') potentiometer controls the degree to which the string decays after it has been excited, with a squeezing action serving to heavily damp the string, and a stretching action serving to reduce the amount of damping to the point where the string can essentially vibrate continuously. This potentiometer also controls the speed at which the Quads move, with the heavily damped values corresponding to faster motion than the relatively undamped values. The reasoning behind this mapping is that when the string is heavily damped, the string pluck from collisions is very prominent, and gives a kind of jittery sense of motion to the sound. When the string is relatively undamped, however, the plucks are far less noticable, and the sound is far more static. The motion of the Quads therefore reflects the perceived motion in the sound, as well as influencing it to a degree (slow moving Quads means fewer collisions).

The rotational ('twist') potentiometer controls the length of the string, and thus its pitch. In an attempt to allow for the instrument to be performed with alongside more traditional instruments,

[4]Originally, each Quad was to have its own exciter, but that proved too computationally expensive, hence the current approach.

this pitch follows the notes of a 12-note equal tempered scale (A4-G#5). The potentiometer is essentially used to navigate through the scale. By twisting it to one or other of its extremes, the performer can change the pitch to the next note up the scale, or the next one down. When the potentiometer is centred, the pitch is that of the current note. When it is between 10-40% of the potentiometer's range the pitch is the interpolated value of the current pitch and the previous one (and vice versa for 60-90% of the pot's range), allowing the performer to work with glissandi or vibrato. The order of the notes in the scale can be customised according to the performer's wishes (so A# does not have to follow A, for example).

Finally, the interface's accelerometer is used to determine the Planet's position within the 3D space of the Ashitaka ecosystem. Obtaining accurate positional data from the accelerometer proved to be quite hard (it may indeed be impossible) however, due to the inherent noise of the device. As such, simply differentiating the signals from it was insufficient, as the positional data calculated would drift off to infinity. The following is pseudo code representing the actual steps taken to obtain stable positional data:

```
//Threshold acceleration data.
//(to make it easier to hold it at (0,0,0) )
if(abs(acceleration) < calibration value)
    acceleration = 0;

//Filter acceleration data to minimise discontinuities.
acceleration = lowPassFilter(acceleration);

//Update velocity.
```

```
velocity = acceleration + previous velocity;
velocity = dcFilter(velocity);

//Update position.
position = velocity + previous position;
position = dcFilter(position);
```

As can be seen, both the velocity and positional data obtained from the initial acceleration data are passed through dc-blocking filters (a high-pass filter set to pass any frequencies above a certain value - the actual values have been hand-tuned for each filter). This is to counteract the aforementioned tendency of the positional data to drift off to infinity if left unchecked. The result of this filtering is that the resultant positional data will always return to zero, but this approach is not without problems. Because of the filtering, there is a noticable degree of lag and overshoot to the positional data when compared with the actual motion of the accelerometer. The effective range of the positional data is also limited to a rough U-shape (this is also partly due to our calibration of the accelerometer to take gravity into account), with the lower-left and lower-right sections of the screen essentially off limits to the performer. Despite these issues, the approach taken represents a reasonable compromise, as there remains a clear link between the performer's gestures and the results on screen.

The colour of the Planet GravityObject's Quads is set to black, as can be seen in Figure 8.6, unless the amplitude output of its string rises above a certain threshold, at which point they become yellow. The reason for this is that, when the 'twist' potentiometer

is at a particular position, the pitch of the exciters corresponds to the resonant frequency of the string. As a result, its amplitude rises significantly (with the soft clipping becoming audible) and the sound perceptibly changes to more of a constant tone. The change in colour is therefore an attempt to link two perceptual audio and visual parameters in a way that is easily recognisable.

Pulsating

Figure 8.7: Pulsating GravityObject type

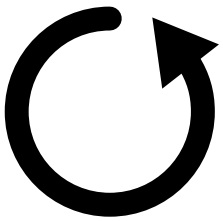


Figure 8.8: Pulsating GravityObject Triggering Gesture

The Pulsating GravityObject type is a simple GravityObject that is intended to possess a pulsating kind of motion. Figure 8.9 is an approximation of the 'pulse' shape it follows. At the start of the pulse cycle, its Quads are arranged close to the GravityObject's position within the 3D ecosystem. As time passes, the GravityObject iterates along the pulse envelope, using the current value on

the envelope to determine how far its Quads should radiate from its own position. At the end of each pulse cycle, the GravityObject jumps to a new random position within the 3D ecosystem, and repeats the pulse cycle.

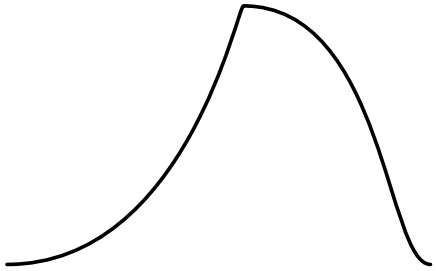


Figure 8.9: Pulsating GravityObject pulse shape

The Pulsating GravityObject's audio is a simple sawtooth wave, the amplitude of which is determined by the GravityObject's current position on the pulse envelope. A sawtooth was chosen as the high partials should help to set up a more complex resonance in the string than something like a simple sine tone. The Pulsating GravityObject's string is 96 cells long.

The GravityObject also sets the colour of its Quads based on the pulse envelope. At the beginning and end of the cycle (the lower section of the envelope), the Quads tend towards red, while at the top of the cycle they tend to pink.

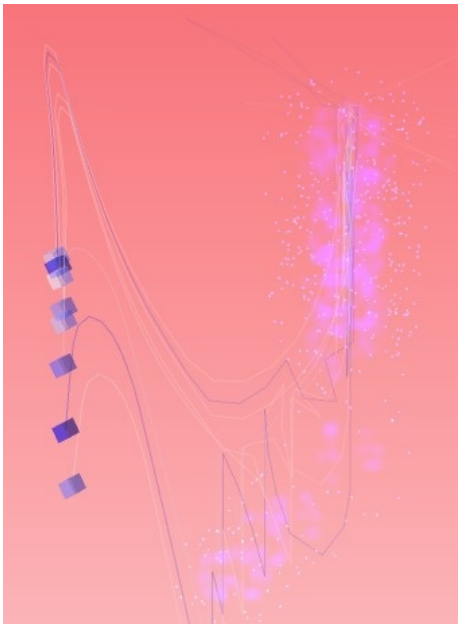


Figure 8.10: Flocking GravityObject type

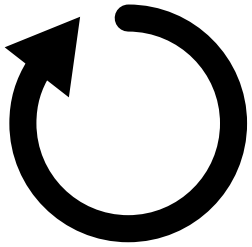


Figure 8.11: Flocking GravityObject Triggering Gesture

Flocking

The Flocking GravityObject is based on Craig Reynolds' original Boids algorithm[96] for modelling the motion of flocks of birds, schools of fish, etc. The GravityObject follows a preset, looped path, with the Quads following it according to Reynolds' flocking algorithm. The algorithm used for the Flocking GravityObject can be seen in Figure 8.10.

This algorithm has been modified from Reynolds' original algorithm to include some rudimentary path-following in place of the original 'steer towards centre of flock' step, though the end result is somewhat similar to the motion of the original algorithm. The only difference is that the global collision detection applied to all

Ashitaka Quads sometimes results in a motion that is often more jittery than the familiar smooth motion common to flocking algorithms.

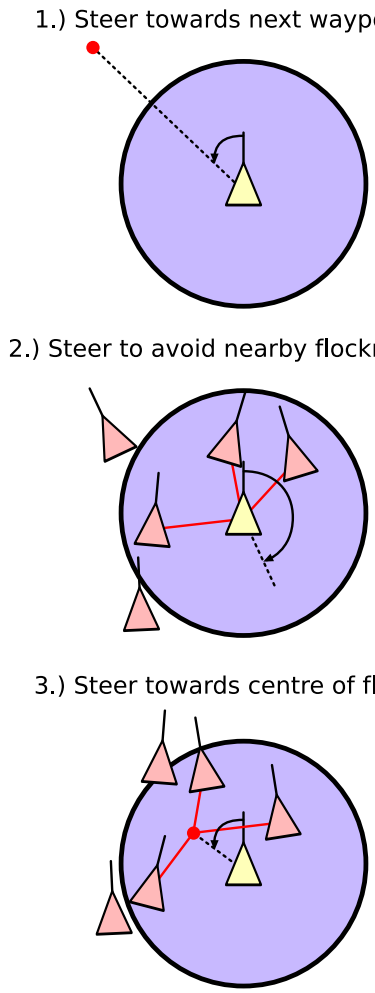


Figure 8.12: Flocking Algorithm

The path that the GravityObject follows is determined by the performer's most recent manipulation of the main (Planet) GravityObject in the scene; specifically, Flocking GravityObjects follow that GravityObject's recent path through the space. If the main GravityObject has been static for a length of time however, a random path is generated for the Flocking GravityObject to follow. In this way, the performer is given control of the Flocking GravityObject's path through the space, and can set up a series of specific trajectories, hopefully making good use of Heilan's audio spatialisation capabilities.

The exciter for the Flocking GravityObject is made up of multiple pulse trains, one for each of the GravityObject's Quads. The amplitude of each pulse train is determined by the distance from the associated Quad to the centre of the flock. The pitch for each pulse train derives from the same frequency, with an offset added according to the angle between the

Quad's vector (with respect to the flock's centre) and an arbitrary 'up' vector[5]. These pulse trains are then summed together and subjected to an amplitude envelope determined by the flock's distance to the next waypoint, such that they are loudest at the position between two waypoints. The summed, enveloped pulse trains are then applied to the GravityObject's string, at a position also determined by the flock's distance to the next waypoint. Again, the use of pulse trains was decided upon due to their posession of significant high frequency partials. The Flocking GravityObject's string is 80 cells long.

Circle

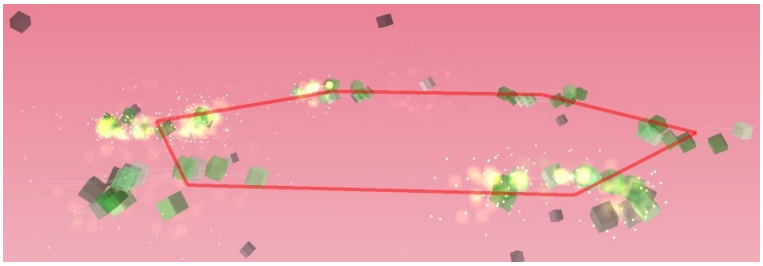


Figure 8.13: Circle GravityObject

The Circle GravityObject is somewhat different to Ashitaka's other GravityObjects, in that it is not a single GravityObject, but a group of six GravityObjects. When this type of GravityObject is triggered by the performer, these six GravityObjects are constructed, and arranged in a rough circle (strictly speaking, a hexagon), spaced

[5]Left of centre equates to a negative offset, right of centre a positive one.

so that their strings are connected as can be seen in Figure 8.13. As such, all the strings will tend to resonate in sympathy when one is excited. Each string is damped to a different degree, and the length of every second string is half that of the others[6] (so its pitch is an octave higher). This is to make the sympathetic resonances somewhat more complex and unpredictable.

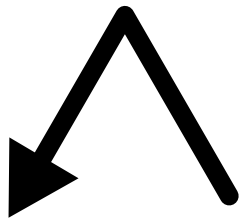


Figure 8.14: Circle GravityObject Triggering Gesture

Visually, each individual GravityObject is responsible for moving its Quads in a smaller circle around itself. This smaller motion is replicated by the GravityObjects themselves slowly rotating around the centre of main circle.

The excitation of the GravityObjects' strings takes the form of filtered noise. Each GravityObject, in turn, modulates the passband frequency of a band pass filter (applied to white noise), following the pattern in Figure 8.15. When this happens, the GravityObject also (following the same modulation pattern) moves outward from the centre of the main circle, breaking the connection between its string and those of its two neighbours.

[6]The lengths are: 128-64-128-64-128-64.

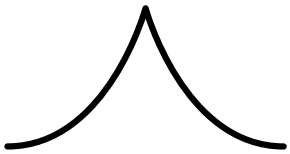


Figure 8.15: CircleGO Excitation Shape

Chapter 9

Evaluation

This chapter presents an evaluation of the Ashitaka instrument. To begin with, its scores on the previously-introduced marking scheme:

- **Visual Expressiveness:** 8
- **Aural Expressiveness:** 7
- **Apparent Strength of Audiovisual Connection (audience):** 5
- **Apparent Strength of Audiovisual Connection (performer):** 6
- **Amount of Effort Required:** 4
- **Fun to Play:** 8
- **Reproducability:** 7

As can be seen, it scores relatively highly. Its performance in these categories will be examined in detail throughout this chapter. It should be noted that this evaluation is based largely on my own

experience with the instrument, as time constraints meant I have had little chance to evaluate other peoples' experience with it.

Before moving on to more subjective evaluation criteria, a brief note on the technical aspects of the project. The whole system (software and hardware) works as intended, with no known performance-hindering issues. The Heilan software could perhaps be better optimised, but does not handicap the performance of the Ashitaka instrument. The software has been tested on three operating systems (Windows, Linux and OSX), and has exhibited no significant problems. As discussed previously, the physical interface does not suffer from noticable latency, though it does use up batteries rather faster than I would wish (generally a single 9V battery will be used up within 1-2 weeks of typical usage).

Given the focus of this thesis, the first question to be answered is how strong the audiovisual connection is. While I initially judged Ashitaka's audiovisual connections to be relatively strong, in retrospect I believe I was rather too lenient in this area during the development of the instrument. Having spent some time away from the instrument and since come back to it in order to perform with it in front of an audience (and gauge their reactions), Ashitaka seems distinctly opaque in terms of the connection between sound and visuals.

Essentially, Ashitaka offers an audience no audiovisual events that are anywhere near as strong as that of a cinematic punch.

Individually there are certain links which work relatively well[1], but when viewed/auditioned together, these links are somewhat drowned out by the 'noise' of so many events occurring simultaneously. A clear example of this is the way a visual particle explosion is generated whenever Quads collide and a string is plucked. On its own this is a very effective audiovisual connection. There is a distinct problem, however, when there are a number of these collisions happening at once. When this happens, the collisions themselves are neither visible nor exactly audible, because the particles generated obscure the visual collision between Quads, and because the string is being plucked so rapidly that the sound takes on a granular nature and the individual plucks get lost within the sound mass. This information overload is perhaps characteristic of Ashitaka's output, and is something I did not perceive during the development of the instrument simply because I was so close to it. Having designed and built the instrument myself, I am intimately aware of what it is doing, and how the various elements relate to each other. As such I did not notice just how opaque the instrument's audiovisual connection can be.

This does demonstrate, however, a tension that ran throughout Ashitaka's development - that of creating a sufficiently rich and interesting instrument while at the same time developing a strong audiovisual connection. These two factors seem to be somewhat

[1]The way The Planet GravityObject will turn its Quads yellow when the sound audibly changes character, the way the visual/spatial motion of the Pulsating and Flocking GravityObjects is linked to their audio...

at odds with each other. Creating a strong audiovisual connection similar to a cinematic footstep or punch seems to require a very simple and easily-graspable set of interactions. As we have seen, synchresis remains strong even when abstracted away from representational, real world sources. The main difference between the short example videos discussed in the Synchresis chapter and Ashitaka is the complexity and multiplicity of the audiovisual connections. The example videos seem to offer a far clearer connection, which is far easier to perceive. On the other hand, it seems that for an instrument to be particulary rich and inviting to potential performers, there must be a significant degree of complexity on offer. As discussed previously, the aim is to entice the performer to continue playing the instrument, to make them want to explore it and develop some kind of mastery with it. My experience with Ashitaka certainly leads me to believe that this desire for complexity works against the aim of creating a strong audiovisual connection.

A possible solution could be to create an environment made up of objects with very limited and simple behaviours (very clear audiovisual connections), and letting complexity develop out of their interactions with each other. Certainly Ashitaka could be seen as the beginnings of such an approach, even it is not entirely successful as an instance of synchresis itself.

In terms of expressive range, the instrument is quite successful. Visually, the inclusion of the accelerometer in the interface

makes it possible for the performer to essentially 'draw' with the instrument in three dimensions. While admittedly more limited than a pen and paper in terms of creating complex, permanent visual forms, this 'drawing' ability still affords the performer a substantial range of potential visual forms or gestures. In addition, the particle system of the Quads and GravityObjects is capable of a range of different forms, and affords the performer significant control over them. Aurally, the instrument presents the performer with what is more or less the same range of expression they would have with a physical string instrument, with some additions in terms of excitation methods.

Regarding the aim of creating an instrument which encourages exploration, I have not had a chance to fully evaluate other people's responses to the instrument. Certainly there are aspects that are intended to encourage exploration, such as the triggering of new GravityObjects from particular gestures (something I would expect novice performers to happen upon accidentally), and the instrument's inherent unpredictability (requiring accurate gestures to get it fully under control). It should also be possible to pick the instrument up without any prior training and create *something* with it, so that performers are not put off by encountering an extreme learning curve. Time constraints have, however, meant that I have not been able to evaluate this aspect of the instrument, and my own experience with it is insufficient for these purposes given

my part in designing and developing it.

The instrument's unpredictability also goes some way to defining how the performer should interact with it, and perhaps makes the instrument more suitable for improvisation, rather than score-led performance. As well as being somewhat unpredictable, the system of GravityObjects and Quads possesses a degree of independence from the performer, in that it will - if left to its own devices - still follow its defined set of actions. This independence is limited in that the system's motion will die away after a period of time, but it is clear to the performer that they are never going to be in complete control of the ecosystem.

A question raised by this part of the discussion is to what degree the performer *can* gain control over the ecosystem. From my own experience with the instrument it seems likely that, given enough time and practice, it should be possible to develop mastery over it such that it is possible to follow a score. While there are aspects of the ecosystem that the performer will never have control over (for example the starting positions of the Quads, which are randomised), there is enough control available that, given suitably accurate gestures, it should be possible for a skilled performer to consistently reproduce actions or phrases. From my own practice I have found that it is the triggering of new GravityObjects which is the hardest action to reproduce consistently. This is most likely due to the fact that it relies on the performer to 'draw' a gesture in

the air. With no guides or aids to help them do so, doing this accurately is not an easy task. I see no reason to believe, however, that a performer could not learn to produce the gestures consistently, given enough time.

One aspect of the instrument which is not particularly satisfying is its use of the 3D soundfield afforded it by the Heilan software. For the most part, sound is concentrated to the centre front. While the various sound sources will move away from this position to a degree, they do not move far, and they will never move behind the camera/virtual ambisonic microphone. There are a number of reasons for this, all related to the main Planet GravityObject. First, the Heilan software is currently only capable of driving a single display output. As such, it cannot present the performer with the visual equivalent of a 3D soundfield. This presents a problem when considering the aim of this PhD to try and connect sound and visuals, in that moving a sound source behind the camera will mean the audience no longer sees the associated visual stimulus, and the connection is broken. To avoid this problem, I tried to make sure the Planet GravityObject is always visible, in front of the camera. A second, related issue is that the performer cannot actually move the Planet GravityObject far beyond the bounds of the camera, due to the need to filter the data from the accelerometer to avoid instability.

The other types of GravityObject perhaps make better use of the soundfield, but still remain - for the most part - in front of the camera. The reason for this is, firstly, that they tend to follow the performer's previous path with the Planet GravityObject, and secondly, that their motion is limited to an 8x8x8 box, centred on the Planet GravityObject's point of origin. Given that the camera is 8 OpenGL units away from that point of origin, it is relatively rare for these GravityObjects to move offscreen, and impossible for them to move behind the camera. This 8x8x8 box is, however, entirely arbitrary, and I intend to enlarge it in the future. It should also be noted that the ability of the performer to rotate the camera represents an attempt to make better use of the soundfield. Considering it always rotates around the Planet GravityObject's point of origin however, it does not have a huge impact, and is instead better at highlighting the 3D nature of the visuals.

Another less than satisfying aspect of the instrument is the way the performer controls its pitch. I believe this is primarily due to the design of the physical interface, which does not offer a particularly sophisticated method of selecting particular pitches. Looking at traditional, physical instruments, almost every instrument in existence seems to present the performer with a range of possible pitches which may be selected and articulated at any time, discretely. Ashitaka, however, requires the performer to 'scroll' through multiple pitches to get to the one they want, an action

which feels (and possibly sounds) somewhat clumsy. The instrument that perhaps comes closest to Ashitaka's method of pitch control is the trombone, but in that case the slide is just one part of an integrated physical system and not necessarily comparable to a simple linear potentiometer. The physical design of the trombone is also perhaps better suited to its purpose than Ashitaka's rotational method of articulation. The trombone's slide has both a longer range than Ashitaka's rotary potentiometer, and it represents an easier articulation on the part of the performer (particularly since the rest of Ashitaka's interface must be held static if the accelerometer is not to be disturbed).

In a similar vein, Ashitaka's use of colour is extremely simplistic, and the performer is offered very little control over it. The use of colour is an issue I have struggled with throughout the development of the instrument, eventually settling on a more or less arbitrary colour scheme for the various visual elements. It may be that all that is required to surmount this problem is a detailed investigation of colour theory in visual art, perhaps drawing on Lawrence Marks' findings on colour perception[82] (i.e. red is warm, blue is cold, etc.).

The trouble I have had with colour may be a symptom of a deeper problem, however. My conception of synchresis as being based on related motion relies on this motion being easily perceived by an audience. It is primarily based on aural and visual attributes

which can be easily quantified in relation to themselves. By this I mean attributes like amplitude (it is easy to say whether one sound is (perceptually) louder than another) and position (ditto whether one object is higher, further away etc. than another). It seems to me though that colour does not quite fit this scheme. In perceptual terms red is not quantatively 'more' than blue - there is no fixed or universally agreed-upon relationship between two colours. Given that the relationship is so uncertain, mapping colour 'motion' (that is, the changing of colour over time) to an aural parameter will always involve a subjective judgement of the relationship between particular colours. This subjectivity, however, is contrary to my aim of constructing an audiovisual connection which is perceived in the same way by anyone, regardless of their cultural background. As such it seems that colour may be incompatible with synchresis used in this fashion. The simplest solution may be to simply regard colour as being separate from the audiovisual connection, and subject to different rules.

In terms of my experience of playing the instrument, I've found that it almost feels more like playing a computer game than playing a traditional instrument. This is, I believe, due to the instrument's visual output, which draws the eye's attention. When playing a traditional acoustic instrument (or even a DMI, with no visual output), I find that, though my focus on visual detail tends to recede behind my focus on sound and haptic sensations, I am always aware of

the space in which I am performing. With Ashitaka, however, I find that I become immersed in the virtual space of the instrument, and far less aware of the physical space in which I am situated. This, to me, is a very similar sensation to that of playing a computer game, where the outside world falls away in favour of the virtual one. This sensation is perhaps heightened by the design of the instrument - in game terminology, the Planet GravityObject would be seen as the player's avatar, charged with embodying them in the virtual space.

Perhaps related to this is the fact that I have found my gestures with the instrument to be far smaller, and less dynamic, than I had expected. The inclusion of an accelerometer in the interface had led me to believe that my own gestures would be affected by it, and that performing with the instrument would involve a lot of large-scale movement and action. Watching footage of myself playing the instrument though, I've found I tend to stand almost stock still, with the only large(-ish) gestures being those intended to trigger new GravityObjects. I think this could be partly due to my immersion in the instrument's virtual space - my awareness of my own body is very diminished compared to that of the Planet GravityObject. As a result, my focus is solely on the effect my gestures may have on the instrument, leading to quite small, focused gestures. This could be seen as a design flaw in the instrument, as the accelerometer was intended to encourage the performer to use large, dynamic gestures. The instrument should react clearly if the performer is particularly active, but this is not the case. Such

large gestures have no more effect on the instrument than small ones do, and if a performer wants to incorporate a lot of physical movement into a performance, it will surely require a divided focus between the instrument's virtual space, and the physical space of the performance. Ideally though, this divided focus would not be necessary, and the performer could concentrate solely on the instrument itself.

Chapter 10

Future Work

10.1 Artistic

The most significant area of this project which begs further work is that of the audio-visual(-performer) connection, and how such an audiovisual medium can be performed with. While this appears to be an area of considerable interest to artists[1], there is very little in the way of theory or guidance available to aid an artist/instrument designer/performer in developing a particular vocabulary or working practice. This effectively forces any such practitioners to start from scratch when attempting to work with such a medium. One of the aims of this thesis was to go some way to correcting this deficiency and provide (and *document*) a potential starting point for artists who want to perform with a music-derived audiovisual

[1] Evidenced in the *'Audiovisual Art'* section in the *Background* chapter, and particularly the numerous audiovisual instruments performed with at conferences such as NIME and the ICMC.

artform. As such, it deliberately focusses on performance with a very specific set of audiovisual materials, only considering abstract visuals and synthesized sounds. A possible next step would be to widen the scope of investigation to incorporate more representational forms (such as photographs, field recordings etc.), and look at the issues which arise from performance with such materials. Such an approach would require a deeper understanding of existing artforms such as cinema and theatre, and borrow from musical practices like musique concrete and electroacoustic composition. This approach would also come closer to Nick Cook's conception of a multimedia artwork, as it would of necessity involve the forces of conformance, complementation and contest outlined in *'Analysing Musical Multimedia'*. The manipulation of such forces in a performance could prove to be a very interesting area for exploration.

It follows that a related goal would be to investigate the design of instruments which allow the performer to manipulate the relationship between audio and visuals. This would turn the audiovisual connection into a performable attribute as opposed to the fixed relationship offered by Ashitaka. It may effectively increase the amount of work required of the performer, however, as it raises the possibility of audio and visual materials which (at times) exist entirely independent of one another. Creating an instrument which allows for such a degree of control without becoming too unwieldy or complex would prove a significant challenge, though it should also provide some significant benefits if successful.

One of the most significant features of the Heilan software environment is its use of Ambisonics to provide a fully 3-dimensional soundfield. Ashitaka does not, however, make good use of this capability, and the instrument's use of the soundfield is extremely simplistic and lacking any real nuance or sophistication. Given Heilan's potential, this is unfortunate, and an in-depth study of this area could yield significant benefits. Such a medium would perhaps be more closely related to architecture than such audiovisual artforms as music video or visual music. The focus on 3-dimensional space certainly sets it apart from most of the artworks discussed in this thesis. In addition, Heilan's ability to create dynamic 3D shapes, and set in motion multiple point sound sources, lends it to the creation of a kind of active audiovisual architecture impossible to realise in the physical world.

Though it is an area which is not explored in Heilan or Ashitaka, one of the issues which caught my attention while researching for this thesis is that of the burgeoning movement to treat computer games as an artform. Particularly while reading reviews of Space Giraffe[2], it became clear to me that certain forms of gameplay make use of a mode of concentration or consciousness that is extremely similar to that practised by performing musicians. Specifically this is to do with the constant repetition of particular physical movements, and the muscle memory that develops as a result. In music

[2]See Background chapter, 2.1.6.

the aim is to train the body to essentially perform the low level physical actions autonomously so that the performer can focus their thoughts on higher level issues such as how to articulate a particular passage in order to impart a particular 'feel'. In games - though it is sometimes debatable to what degree the use of muscle memory is intended - the same processes frequently come into play, with players often talking about being 'in the zone'. This can be explained as the point where their consciousness is altered in such a way that they no longer consciously focus on what their hands (fingers, thumbs...) are doing, instead reacting to the game in an immediate, almost subconscious fashion. This parallel between musical performance and gameplay suggests to me that there is substantial scope for an exploration of how computer games may be leveraged in the design of new audiovisual instruments. I would even suggest that games with this kind of (rapid reaction, 'twitch'-based) gameplay constitute a form of audiovisual instrument, albeit one with a typically very limited range of expression. From my own - very limited - review of the literature in the field of video games research, it seems that Bayliss' paper *'Notes Toward a Sense of Embodied Gameplay'*[47] (itself based on Paul Dourish's ideas of embodied interaction) may form a possible starting point for a more detailed investigation of the similarities between games and musical instruments.

A perhaps obvious omission from this thesis is the lack of an

evaluation - by an audience - of the audiovisual connection constructed for Ashitaka. That connection was based on Chion's notion of synchresis (and my own experience of synchresis), but I have no real experimental data to back up my use of this approach. This omission was simply due to lack of time, but detailed experiments along the lines of those outlined in [103] could yield important insights into the phenomenon of synchresis. A more thorough review of psychological research in this area should also prove advantageous.

While it has - to date - only been used for solo performances and rendered (non-realtime) audiovisual works, Heilan's extensive Open Sound Control support makes it ideal for collaborative performances. This is an area I would like to explore in the future, though more development of the software may be required beforehand. Firstly, at the time of writing Heilan only acts as an OSC server, meaning it can receive OSC messages but not transmit them. The result is that any collaborative performances would have to be run on a single instance of Heilan, with a single video output. The performers could interact with this instance from other computers (running software to output OSC messages), but they could not view the performance from those computers. Depending on the setup, this may hinder the performance. The solution to this problem is simply to make Heilan also act as an OSC client, able to transmit as well as receive OSC messages. This would mean

that the primary instance of Heilan (the one presumably connected to a projector in this collaborative scenario) - as well as receiving the performers' inputs via OSC messages - could update any number of secondary instances to display the current X3D scene as it unfolds.

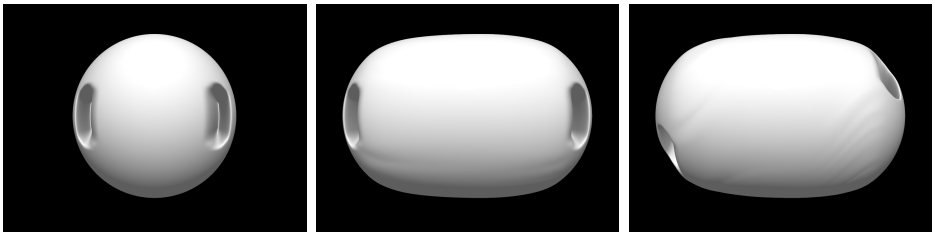
The second issue is that, although it was developed to aid audiovisual performance, Heilan does not offer much in the way of (fused) audiovisual materials for performers to work with (barring Ashitaka and the experimental instruments). Instead, due to its X3D origins, the malleability of the visuals is far greater than that of the audio, which consists solely of very simple sound file manipulations. As such, the development of a more sophisticated set of audio node types should be a high priority, as well as an infrastructure which may be used to link the two domains. By doing this, it should be far easier for performers to develop their own audiovisual instruments within the software, and the audiovisual relationships will be far more flexible than the relatively static connections present in Ashitaka. It also raises the possibility of the software being treated as one big, collaborative instrument, as opposed to a virtual environment which houses a number of separate individual instruments.

10.2 Technical

The most immediate technical issue remaining to be solved for Ashitaka (and the wider Heilan environment) is Heilan's inability to correctly rotate the Ambisonic soundfield to match the rotation of the 3D graphics. As it stands, rotation around a single axis works correctly, but rotation around multiple axes results in the soundfield getting out of sync with the OpenGL visuals. Initially even single axis rotation was out of the question, as I naïvely assumed I could simply rotate the (by this point B-format encoded) soundfield by the same degree as the visual camera, and they would both line up. This was a false assumption due to the order of the translate and rotate operations in the two domains. In the OpenGL scene, the scene is first rotated according to the camera's rotation, then translated according to its position. In the soundfield, however, the two operations happen in the opposite order, resulting in the positions of sounds not matching up with the positions of related visual objects. To solve this problem, I rearranged the method of calculating sounds' positions, so that each sound is rotated and translated in the same way as any visual objects are, before it is Ambisonically encoded. I am unsure why multiple axes rotation is still broken, but I believe it is likely the result of another discrepancy between the way visual and audio objects are placed in the scene.

Ashitaka's physical interface also requires some work. The cur-

rent version is essentially a (fairly ugly) prototype of the intended design. My original intention was for the design to somewhat mimic the kind of rounded shapes which may be obtained from manipulating a blob of clay, and this remains my goal. Figure 10.1 demonstrates my vision of an ideal interface for Ashitaka. For this design the electronics and moving parts would all be covered by some kind of stretchable fabric. The handholds would allow the performer to stretch and twist the interface in the same way as is possible with the prototype, with the four force sensors being mounted inside, at the end of the holes visible towards the left and right of the interface.



(a) Centred (b) Stretched (c) Stretched and Twisted

Figure 10.1: New Interface Mockup

While the current approach of using a PIC microcontroller and separate bluetooth module seems to work well, future versions of the interface may be better served by making use of an Arduino[1] board with integrated bluetooth. The Arduino platform is programmed using a Java-esque language (based on the Wiring[42] language), and has been specifically designed for artistic projects like Ashitaka's interface. Most importantly, however, the integrated

bluetooth module would considerably simplify the design of the electronic hardware for the interface. The only additional electronics required for this approach would be a power supply and the various sensors. Had the bluetooth Arduino board been available when I started work on the prototype interface I would have used it in preference to the current approach.

While the previous two paragraphs are mainly concerned with, essentially, 'finishing' the interface, I would also like to extend it somewhat so that it is less passive with regard to the performer's actions. Firstly, I would like to provide the ability to generate light, in accordance with what happens on screen. The mockups in Figure 10.1 would lend themselves particularly well to the addition of LEDs below the stretchable fabric, transforming the white, spherical object into a potentially more interesting glowing, stretchable ball. Secondly, and perhaps more significantly, I would like to add some degree of haptic feedback to the interface. There are at least two potential uses for haptics with this interface; vibrational motors which activate when the performer moves the main GravityObject close to other GravityObjects, and forces which act against the performer when they try to stretch or twist the interface. The inclusion of these forces would potentially enhance the performer's connection to the instrument and possibly aid in the articulation of certain gestures.

At the time of writing, Ashitaka is essentially a self-contained

system. While it is possible to add other X3D elements to a scene containing Ashitaka, the instrument will not interact with these elements in any way. One way to add interaction between the various elements would be the addition of collision detection. At present, Ashitaka's collision detection is limited to collisions between its own Quads. The X3D specification does, however, specify collision behaviour which can be applied to all X3D (geometry) nodes. Implementing this would add a significant extra dimension to the instrument, as complex 3-dimensional scenes could be developed for Ashitaka to interact with. It also raises the possibility of a more collaborative performance environment. One option could see one performer manipulating Ashitaka while another manipulates the environment Ashitaka exists in, perhaps enclosing the instrument in a small space only to let it explode out at a later point. As mentioned in the Heilan chapter, the software's OSC support does lend itself to collaborative performances, and the addition of global collision detection should substantially increase the collaborative possibilities of the software.

One area where Heilan's design is not optimal is its scenegraph. The scenegraph is the structure which the software uses to describe and manipulate the X3D scene it is displaying. Heilan's scenegraph is somewhat naïvely derived from the structure of the X3D files it parses. While this has not proved to be a major issue to date, if Heilan is to be expanded the scenegraph will need to be-

come more sophisticated. For instance, the implementation of collision detection and X3D mouse interaction nodes such as TouchSensor will require a mechanism for querying geometry nodes' geometry. Such a mechanism does not currently exist in Heilan, as all geometry data is passed straight to the graphics card, and never exists in the kind of generalised form which would be required[3]. The other issue with the scenegraph is that Heilan treats a node's attributes (i.e. field types such as SFFloat, MFString etc.) differently to its child nodes (anything derived from X3DNode). While not immediately limiting, it would possibly be more flexible to treat both field types and node types as being fundamentally the same. One thing I would like to add to Heilan is the ability to manipulate a scene via OSC, and then save the result. At present this would require each node type to have something like a *'writeAttributes()'* method implemented for it, but if all field types possessed the ability to output their value as text, the whole process could be generalised considerably. The ability to treat child nodes as attributes (possibly via an overloaded *operator=()*) would also simplify the code necessary to generate the geometry for certain nodes such as IndexedFaceSet.

Barring some minor exceptions, Heilan currently supports the X3D Interchange Profile in full[4]. This profile, however, was only

[3]For example, an IndexedFaceSet node stores its geometry data in a different form to that of a TriangleStripSet node, which stores its geometry differently to a Cone node, etc.

[4]I should point out that it has not been subject to the standards body's con-

designed to provide the base level of support for exchanging and rendering 3D graphics. It does not provide for the level of interactivity and audio performance which Heilan (as an environment for audiovisual performance) demands. As such, I feel it is important for Heilan to support at least the Interactive profile, if not the Immersive one. Adding support for these profiles will substantially enhance Heilan's capabilities, and will potentially allow for a far greater range of audiovisual expression. Some node types which would be of substantial benefit to Heilan are as follows:

- **MovieTexture:** Allows the user to make use of videos to texture geometry. At the time of writing this node has been partially implemented, making use of the Xine[44] library.

- **Particle Systems Component:** A number of node types which can be used to generate particle-based visual effects. Particle effects are widely used in computer visuals and would surely get a lot of use if implemented in Heilan.

- **Programmable Shaders Component:** Shaders are used to reconfigure the graphics card's rendering pipeline, and are extremely useful in generating new and unique visual effects. This component is partly implemented at the time of writing.

- **Rigid Body Physics Component:** This component consists of a set of nodes which define physics-based interactions (so the

formance testing suite, as that requires membership of the Web3D organisation, which itself costs a reasonable amount of money. Enough, at least, to put it beyond my own means.

scene's geometry can be made to move in a realistic fashion). Its addition would prove extremely valuable to Heilan.

Somewhat related to the need for a wider range of supported node types is the need for Heilan to be able to delete nodes from a scene while it is running. This is needed by, for example, the Anchor node type, which will replace the current scene with a new one upon activation (similar to the HTML anchor tag). At the moment this is not possible in Heilan due to its multithreaded nature and the integrated form of the scene graph. Essentially, deleting a node from the scene in one thread could easily crash another thread which is not aware that the node it's operating on has been deleted. One solution to this would be to have separate scene graphs for each thread (graphics thread, audio thread...), which is how the X3DToolkit library approaches the problem. For Heilan though, this complicates the (conceptual) connection between sound and visuals, as any audiovisual objects would no longer be integrated, *audiovisual* objects, but a combination of separate *audio* and *visual* objects. A better solution would be to implement a form of rudimentary garbage collection, whereby a node which is to be deleted will first be removed from its parent node's list of children and stored in a separate list. After a 'safe' amount of time then, the nodes waiting on this list can be deleted, safe in the knowledge that any threads which were acting on them have finished what they were doing and are now aware that the scenegraph has

changed.

The last issue related to Heilan's implementation of the X3D specification is support for scripting. As it stands, Heilan's support for X3D's events and routing mechanisms make it possible for scene authors to set up a degree of interactivity and time-based triggers or sequences. This system, however, is not easily extended by scene authors, as (among other things) they have no way of defining and adding new node types. The X3D specification defines scripting bindings for the ECMAScript (née javascript) and Java languages, which essentially provide scene authors with the tools to overcome these limitations. As such, the addition of scripting capabilities would make Heilan a substantially more flexible tool for audiovisual artists.

Chapter 11

Conclusion

This thesis has presented an approach to constructing audiovisual instruments based on Michel Chion's notion of synchresis. Several experimental audiovisual instruments were developed to investigate how to work with synchresis in a performance situation, resulting in a mapping strategy which proposes the use of multiple 'Audiovisual Parameters'.

As a base for these experimental instruments, a software environment named Heilan was developed. Heilan is an X3D browser capable of encoding and decoding sound to Ambisonic B format to generate a full 3D soundfield given enough speakers.

The Ashitaka instrument was developed to represent the culmination of these efforts to link sound and visuals in a musical instrument. This instrument was designed as a kind of ecosystem, made up of various objects, or actors, which interact with each other. The performer is given direct control over some parts

of the ecosystem, but may only interact with other parts indirectly. The performer interacts with the ecosystem through the use of a custom physical interface. This interface uses bluetooth and Open Sound Control to communicate with Heilan, and presents the performer with various possible gestures similar to those possible with a block of clay, as well as incorporating a 3D accelerometer. While the instrument is not entirely successful in terms of creating an inseperable, fused audiovisual material, it nevertheless fulfils many of the criteria (outlined earlier in the thesis) required of a successful instrument. These criteria include; allowing a novice performer to pick it up and produce something satisfactory without prior training; possessing a degree of unpredictability, to make the performer work (and learn the instrument) in order to achieve the desired results; and encouraging exploration, to entice the performer to keep playing the instrument.

Creating such an instrument naturally gave birth to a specific audiovisual style, which is largely unique to Ashitaka. The development of the instrument was largely focused on creating a situation out of which complex interactions could arise, and which would encourage exploration on the part of the performer. As such, the specific implementation details of the instrument were often - initially - chosen more for functional than purely aesthetic reasons. The choice of physical modelling for the audio, for example, was strongly influenced by the inherent modularity and ease of connection offered by physical models.

Ashitaka's particular aesthetic results in part from the interaction between the sound and visual generation methods and the physical interface. Looking again at the audio, Ashitaka tends to produce droning and feedback (similar to electric guitar feedback) sounds. This is partly because the string models involved lend themselves to these kind of sounds, but it is also due to the method of excitation used, which is itself tied into the design of the physical interface and the ecosystem structure which guided the design of the instrument. The force sensors on the interface seem to encourage gentle, gradual squeezing motions if the performer is to be accurate with their gestures and exploit the instrument to its fullest. The design of the ecosystem also, however, means that Quads move with a degree of latency or momentum (in order to avoid sudden discontinuous jumps of position). The combination of these two factors essentially means that the performer is not in a position to articulate the kind of gestures that would usually be expected of string instruments (or physically-modelled strings). Their gestures with the force sensors tend towards long, flowing movements, which naturally translates into the kind of droning sounds Ashitaka seems most at home with. Obviously this is all related to the distinct difference between Ashitaka's physical interface and that of acoustic string instruments. While Ashitaka incorporates a number of playing techniques common to string instruments (plucking, damping, etc.) these are not under direct control of the performer. The performer is not given a string-type interface which

they can manipulate as they would, for example, a violin. Instead the design of Ashitaka's interface puts them at an extra remove from the sound generation mechanism (a feature common to all digital musical instruments, but perhaps pronounced in Ashitaka due to its particular interface).

Perhaps the most characteristic aspect of Ashitaka's audio is its loud, noisy nature, somewhat dissimilar to the sounds typically produced by physical modelling. There is a slightly chaotic and organic aspect to the sound which is perhaps more similar to circuit-bent electronics and more left-field electric guitar performance (belying my own interest in such sounds) than most computer-based synthesis algorithms. Such algorithms are often used in such a way as to emphasize clarity and the detail in the sound, whereas Ashitaka tends to present quite distorted sounds, often obscuring the details which would be present in a purer implementation. This noisy, organic character is possibly the result of two main factors; the soft clipping distortion applied to the string models, and the varied methods of excitation of the models. The soft clipping is an example of how Ashitaka's audiovisual style developed from both functional and aesthetic principles. The technical reason this distortion is applied to the string models is to ensure their output is restricted to a set range, as without this limiting, the models can become unstable under certain conditions and will resonate at exponentially-increasing amplitudes. A significant factor in choosing *this* particular soft clipping algorithm, however, was that it

produces a distortion not unlike that commonly applied to electric guitars. As such, it performs a significant role in defining the instrument's sound, and setting it apart from existing physically-modelled instruments. To the best of my knowledge the methods of excitation applied to the string models are also fairly unique to Ashitaka, and represent a combination of organic (the string models themselves, based on real world sound generation mechanisms) and distinctly synthetic sounds (the pulse train and sawtooth excitations). Again, these methods of excitation were partly chosen for functional reasons - the high partials involved excite the strings in such a way as to create a fairly rich resonance - but they also play a significant role in defining the instrument's noisy, chaotic aesthetic, adding a certain roughness and a jagged character to the sound.

A further point to note is that physical modelling is possibly the only synthesis method which can accommodate the disparate sound generation techniques used in Ashitaka in such a coherent manner. While it should be possible to create similar sounds with other methods, it would require far more detailed control over individual synthesis parameters. One of the primary benefits of physical modelling is the ease with which these kind of complex sounds can be created. The particular sonic aesthetic Ashitaka ended up with was very much an organic development, born from the possibilities the string models presented. To arrive at the same aesthetic via a different synthesis method would both require a longer devel-

opment period, and a far more detailed idea of the specific sound I was looking for from the start. It was the choice - from a functional perspective - to use a synthesis method which offers such possibilities that gave birth to a sonic aesthetic which I could then refine.

Visually, Ashitaka draws on the visual aspects of various previous projects of mine. The particles emphasizing collisions come from the original Particle Fountain VST plugin, the Quads' trails come from the Brush Strokes VST plugin, and the use of cubes as the core visual material comes from an earlier audiovisual piece I did called *origins*. In *origins* I treated the cubes like voxels (3d pixels), only to then let them break free of the grid, and Ashitaka can be seen to do a similar thing, albeit without an explicit recognition of the grid.

Ashitaka's audiovisual style is the result of the combination of a number of existing audio and visual algorithms or organisational approaches. Physically-modeled strings, particle generators, granular synthesis (in the shape of the string plucks), and geometric shapes are all involved in the instrument's output. This combination of otherwise disparate elements gives Ashitaka what is to the best of my knowledge a unique character. Golan Levin's AVES[80] is perhaps the closest example of an alternative approach in this area, and there are marked differences between the two. By focusing on the painterly interface metaphor, Levin's visuals tend to appear very organic, lacking straight edges and corners. Ashitaka,

however, mixes organic contours (the trails of the Quads) with strict geometric shapes (the cubes representing the Quads' position). Sonically, the organic/inorganic distinction is switched, with Ashitaka's string models deliberately making use of sonic characteristics common to acoustic instruments (albeit with certain synthetic additions), in contrast to the clearly computer-generated sounds of Levin's creations. Perhaps the biggest difference though is the interface. Ashitaka is very much a musical instrument in the tradition of existing acoustic instruments - it has a unique physical interface with multiple degrees of freedom, and the performer is not tied to the computer (in the sense that the bluetooth communications allow them to freely move about). By making use of existing interfaces such as the mouse and the stylus, however, Levin's instruments situate themselves largely outside this tradition. Ashitaka was designed to be performed in a similar way to a violin or a guitar - it was designed as a musical instrument - whereas Levin's instruments seem to have been designed more for visual artists who want to work with sonic as well as visual materials.

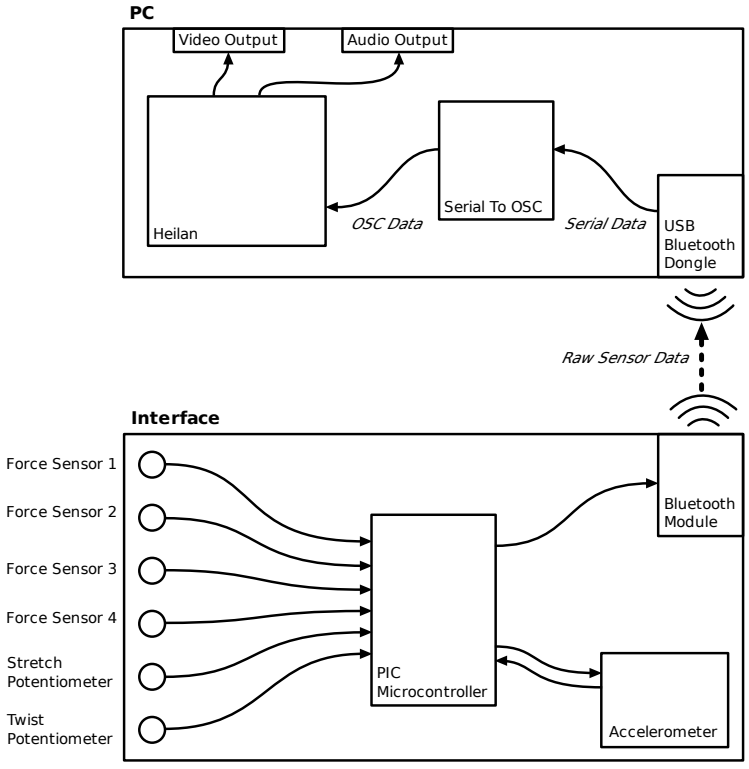
Perhaps the most salient point to arise from the development of Ashitaka is the tension between the desire to create a rich and involving instrument, and that of creating a strong audiovisual connection. My experience with Ashitaka leads me to conclude that this is an opposition of complexity versus simplicity. It appears that for an instrument to entice a performer to explore, and de-

velop some skill with it, the instrument must possess a degree of complexity. This encourages the performer to spend time with the instrument and try to get better at playing it, and is a feature common to all widely-used musical instruments. On the other hand, it seems that strong audiovisual connections desire simplicity. As we are dealing with the perception of a connection between two otherwise separate domains (sound and vision), it is important that an audience can clearly perceive similar motion in both domains, and simplicity and clarity is vital. Ashitaka itself falls more on the side of complexity, of creating an instrument that is involving and fun to play. In creating the ecosystem which largely fulfils this aim, the audiovisual connection has been sacrificed to some degree, in the sense that it is often obscured by the mass of information competing for the performer's (and the audience') attention. Nevertheless, Ashitaka is quite successful as an instrument in its own right, and it points towards some interesting directions for the development of further audiovisual instruments.

Appendices

Appendix A

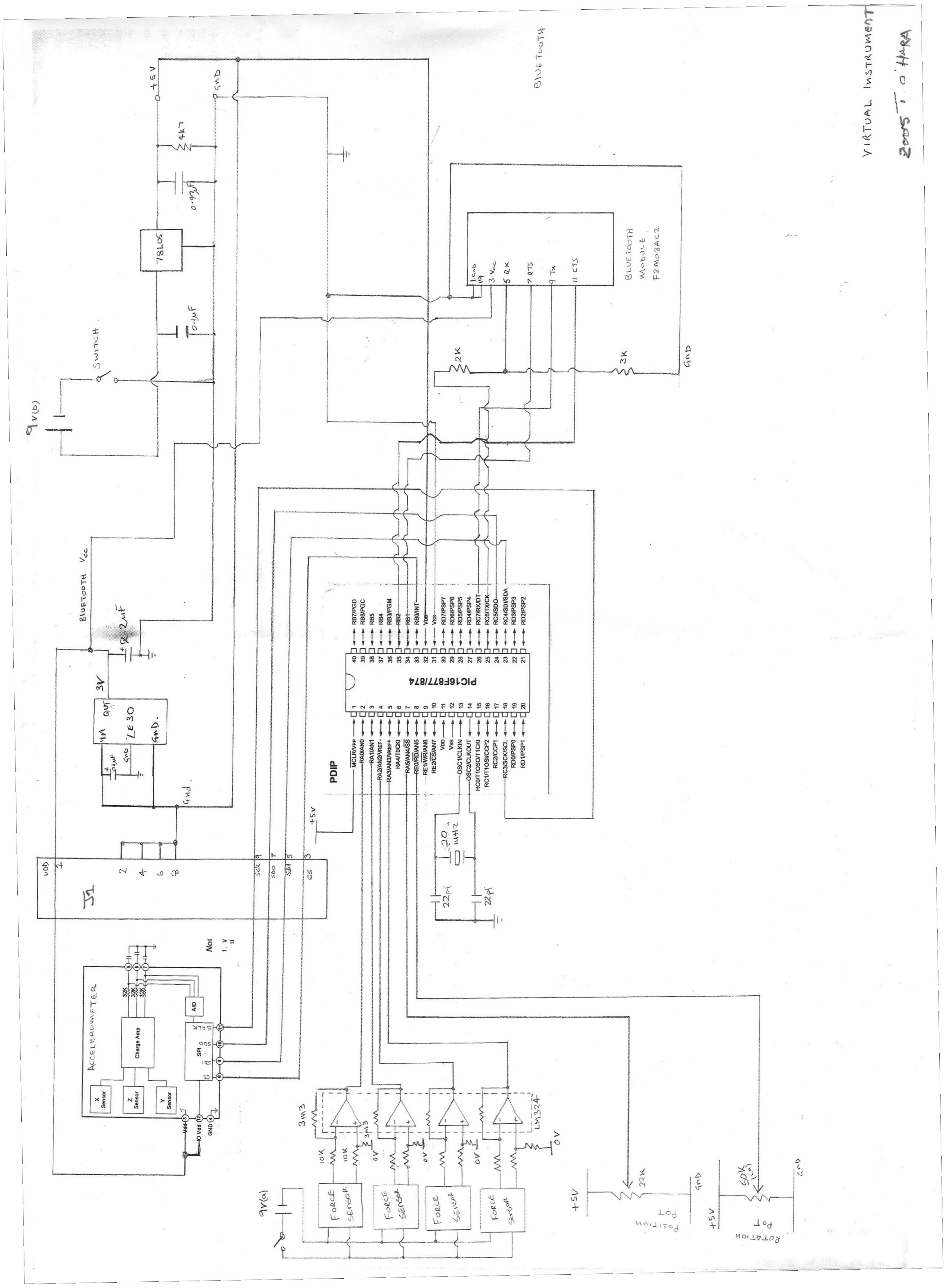
Ashitaka System Diagram



Appendix B

Ashitaka Interface Schematic

Courtesy Tom O'Hara.


VIRTUAL INSTRUMENT
2005 T. O'HARA
BLUETOOTH
BLUETOOTH MODULE F2M03AC2
PIC16F877/874
PDIP
ACCELEROMETER
78L05
LE30
LM324
SWITCH
FORCE SENSOR
POSITION POT
ROTATION POT
BLUETOOTH Vcc

Bibliography

[1] Arduino. Website: http://www.arduino.cc/. Accessed: 04/02/08.

[2] Bluetooth Profiles Overview. Website: http://bluetooth.com/Bluetooth/Technology/Works/Profiles_Overview.htm. Accessed: 16/04/08.

[3] Cthugha. Website: http://www.afn.org/~cthugha/. Accessed: 28/09/07.

[4] Different ways of implementing factories. Website: http://www.codeproject.com/cpp/all_kinds_of_factories.asp. Accessed: 23/11/07.

[5] FFTW. Website: http://www.fftw.org/. Accessed: 19/11/07.

[6] Free2move F2M03AC2 Data sheets. Website: http://www.free2move.se/index.php?type=view&pk=27&pk_language=1. Accessed: 16/04/08.

[7] FreeWRL. Website: http://freewrl.sourceforge.net/. Accessed: 15/11/07.

[8] Honeywell FSG15N1A Datasheet. Website: http://sccatalog.honeywell.com/imc/printfriendly.asp?FAM=force&PN=FSG15N1A. Accessed: 16/04/08.

[9] Independent JPEG Group. Website: http://www.ijg.org/. Accessed: 19/11/07.

[10] iTunes. Website: http://www.apple.com/itunes/download/. Accessed: 28/09/07.

[11] Kionix KXP74-1050 Data sheets. Website: http://www.kionix.com/accelerometers/accelerometer-KXP74.html. Accessed: 16/04/08.

[12] libsndfile. Website: http://www.mega-nerd.com/libsndfile/. Accessed: 25/4/06.

[13] libvorbisfile. Website: http://www.xiph.org/downloads/. Accessed: 25/4/06.

[14] Magnetosphere visualizer. Website: http://software.barbariangroup.com/magnetosphere/. Accessed: 28/09/07.

[15] Microchip PIC16F874A Data sheets. Website: http://www.microchip.com/stellent/idcplg?IdcService=

SS_GET_PAGE&nodeId=1335&dDocName=en010238. Accessed: 16/04/08.

[16] Neon lightsynth. Website: http://www.llamasoft.co.uk/neon.php. Accessed: 28/09/07.

[17] New Interfaces for Musical Expression. Website: http://www.nime.org/. Accessed: 19/10/07.

[18] OpenGL Home Page. Website: http://www.opengl.org/. Accessed: 16/11/07.

[19] PortAudio. Website: http://www.portaudio.com/. Accessed: 25/4/06.

[20] Processing programming language. Website: http://www.processing.org/. Accessed: 28/09/07.

[21] Rewire protocol. Website: http://www.propellerheads.se/technologies/rewire/index.cfm?fuseaction=mainframe. Accessed: 15/11/07.

[22] Second life. Website: http://secondlife.com/. Accessed: 28/02/08.

[23] Simple Direct Media Layer Home Page. Website: http://www.libsdl.org/. Accessed: 19/11/07.

[24] TinyXML. Website: http://www.grinninglizard.com/tinyxml/. Accessed: 25/4/06.

[25] Vimeo. Website: http://www.vimeo.com. Accessed: 11/03/08.

[26] VJamm: VJ Software. Website: http://www.vjamm.com/. Accessed: 22/04/08.

[27] *VRML97 Functional specification ISO/IEC 14772-1:1997*. Website: url-http://www.web3d.org/x3d/specifications/vrml/. Accessed: 05/08/08.

[28] Web3D Consortium - The standards body behind X3D. Website: http://www.web3d.org/. Accessed: 15/11/07.

[29] Wikipedia Black & White Article. Website: http://en.wikipedia.org/w/index.php?title=Black_%26_White_%28video_game%29&oldid=208405950. Accessed: 29/04/08.

[30] Wikipedia Colour Red Article. Website: http://en.wikipedia.org/w/index.php?title=Red&oldid=204919917. Accessed: 29/04/08.

[31] Wikipedia Dance Dance Revolution Article. Website: http://en.wikipedia.org/w/index.php?title=Dance_Dance_Revolution&oldid=208830934. Accessed: 29/04/08.

[32] Wikipedia Demoscene Article. Website: http://en.wikipedia.org/w/index.php?title=Demoscene&oldid=153918468. Accessed: 25/09/07.

[33] Wikipedia Guitar Hero Article. Website: http://en.wikipedia.org/w/index.php?title=Guitar_Hero_%28series%29&oldid=208616161. Accessed: 29/04/08.

[34] Wikipedia Music Visualization Article. Website: http://en.wikipedia.org/w/index.php?title=Music_visualization&oldid=160263952. Accessed: 27/09/07.

[35] Wikipedia Rez Article. Website: http://en.wikipedia.org/w/index.php?title=Rez&oldid=208994969. Accessed: 29/04/08.

[36] Wikipedia Serial Peripheral Interface Article. Website: http://en.wikipedia.org/w/index.php?title=Serial_Peripheral_Interface_Bus&oldid=208939147. Accessed: 29/04/08.

[37] Wikipedia Space Giraffe Article. Website: http://en.wikipedia.org/w/index.php?title=Space_Giraffe&oldid=207825564. Accessed: 29/04/08.

[38] Wikipedia Tempest Article. Website: http://en.wikipedia.org/w/index.php?title=Tempest_%28arcade_game%29&oldid=210970463. Accessed: 13/05/08.

[39] Wikipedia Virtual Studio Technology Article. Website: http://en.wikipedia.org/w/index.php?title=Virtual_Studio_Technology&oldid=205828677. Accessed: 29/04/08.

[40] Wikipedia VRML Article. Website: http://en.wikipedia.org/w/index.php?title=VRML&oldid=171292686. Accessed: 15/11/07.

[41] Winamp. Website: http://www.winamp.com/. Accessed: 28/09/07.

[42] Wiring. Website: http://wiring.org.co/. Accessed: 04/02/08.

[43] World of warcraft. Website: http://www.worldofwarcraft.com/index.xml. Accessed: 28/02/08.

[44] Xine multimedia player. Website: http://www.xinehq.de. Accessed: 05/02/08.

[45] YouTube. Website: http://www.youtube.com. Accessed: 11/03/08.

[46] D. Arfib, J.M. Couturier, L. Kessous, and V. Verfaille. Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces. *Organised Sound*, 7(2):127–144, 2002.

[47] Peter Bayliss. Notes Toward a Sense of Embodied Gamplay. In *Digital Games Research Association Conference*, 2007.

[48] David Bernard. Visual wallpaper. Website: `http://www.vjtheory.net/web_texts/text_bernard.htm`. Accessed 09/11/07.

[49] Jonathan Blow. In which I compare Space Giraffe to Ulysses. Website: `http://braid-game.com/news/?p=100`. Accessed: 02/10/07.

[50] Bert Bongers and Yolande Harris. A Structured Instrument Design Approach: The Video-Organ. In *New Interfaces for Musical Expression*, 2002.

[51] Kerry Brougher, Jeremy Strick, Ari Wiseman, and Judith Zilczer. *Visual Music: Synaesthesia in Art and Music Since 1900*. Thames and Hudson, April 2005.

[52] Gemma Calvert, Charles Spence, and Barry E. Stein, editors. *The Handbook of Multisensory Processes*. MIT Press, 2004.

[53] Stuart Campbell. Why the Long Face? The rights and wrongs of Space Giraffe. Website: `http://worldofstuart.excellentcontent.com/giraffe/giraffe3.htm`. Accessed: 02/10/07.

[54] Michel Chion. *Audio-Vision: Sound on Screen*. Columbia University Press, 1994.

[55] Fred Collopy. RhythmicLight.com. Website: http://rhythmiclight.com/, 7 August 2005. Accessed 17/01/2006.

[56] Nicholas Cook. *Analysing Musical Multimedia*. Oxford University Press, 1998.

[57] Paul Davis. JACK Home Page. Website: http://jackit.sourceforge.net/.

[58] D.G. Malham. Spatial Hearing Mechanisms and Sound Reproduction. Website: http://www.york.ac.uk/inst/mustech/3d_audio/ambis2.htm, 15 February 2002.

[59] Walt Disney. Fantasia. Film, 1941.

[60] Oleg Dopertchouk. Recognition of Handwritten Gestures. Website: http://www.gamedev.net/reference/articles/article2039.asp. Accessed: 21/04/08.

[61] Roger Ebert. Games vs. Art: Ebert vs. Barker. Website: http://rogerebert.suntimes.com/apps/pbcs.dll/article?AID=/20070721/COMMENTARY/70721001. Accessed: 15/04/08.

[62] Michael Faulkner, editor. *VJ: audio-visual art and vj culture*. Laurence King Publishing Ltd., 2006.

[63] Sidney Fels, Ashley Gadd, and Axel Mulder. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, 2002.

[64] Elfriede Fischinger. Writing light. Website: `http://www.centerforvisualmusic.org/WritingLight.htm`. Accessed: 18/10/07.

[65] Oskar Fischinger. Sounding Ornaments. *Deutsche Allgemeine Zeitung*, 8 July 1932. Website: `http://www.oskarfischinger.org/Sounding.htm` Accessed 25/01/06.

[66] Richard Furse. First and Second Order Ambisonic Decoding Equations. Website: `http://www.muse.demon.co.uk/ref/speakers.html`. Accessed: 23/06/08.

[67] Michael Gerzon. Surround-sound psychoacoustics. *Wireless World*, 1974. PDF available: `http://www.audiosignal.co.uk/Gerzon%20archive.html`.

[68] Dave Green. Demo or Die! *Wired*, 3.07, 1995.

[69] Gareth Grimshaw, Mark Schott. Situating Gaming as a Sonic Experience: The acoustic ecology of First-Person Shooters. In *Digital Games Research Association Conference*, 2007.

[70] Thomas Gruetzmacher. PC Demoscene FAQ. Website: `http://tomaes.32x.de/text/faq.php`. Accessed: 13/05/08.

[71] I.P. Howard and W.B. Templeton. *Human Spatial Orientation.* Wiley, 1966.

[72] Andy Hunt and Marcelo M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(2):97–108, 2002.

[73] Cockos Incorporated. Ninjam: Realtime music collaboration software. Website: `http://www.ninjam.com/`. Accessed: 25/02/08.

[74] Sergi Jorda. Digital instruments and players: Part 1 - efficiency and apprenticeship. In *New Interfaces for Musical Expression*, 2004.

[75] Wassily Kandinsky. *Concerning the Spiritual in Art.* Dover Publications, Inc. New York, 1977 (originally published 1914).

[76] Wolfgang Köhler. *Gestalt Psychology.* G. Bell and Sons Ltd., 1930.

[77] Marilyn S. Kuchner. *Morgan Russell.* Hudson Hills Press, 1990.

[78] Mattias Lantinga, Sam Engdegård. SDL image. Website: `http://www.libsdl.org/projects/SDL_image/`. Accessed: 19/11/07.

[79] Jan Le Goc. X3D ToolKit. Website: http://artis.imag.fr/Software/X3D/. Accessed: 15/11/07.

[80] Golan Levin. *Painterly Interfaces for AudioVisual Performance*. PhD thesis, Massachusetts Institute of Technology, 9 2000. Website: http://acg.media.mit.edu/people/golan/thesis/index.html. Accessed 18/01/2006.

[81] Perttu Mäki-Patola, Teemu Hämäläinen. Latency Tolerance for Gesture Controlled Continuous Sound Instrument without Tactile Feedback. *International Computer Music Conference*, 2004.

[82] Lawrence E. Marks. *The Unity of the Senses: Interrelations among the Modalities*. Academic Press, 1978.

[83] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[84] Scott Meyers. *Effective C++*. Addison-Wesley Professional, 2005.

[85] Jeff Minter. Llamasoft virtual light machine history. Website: http://www.llamasoft.co.uk/vlm.php. Accessed: 27/09/07.

[86] Ali Momeni and Cyrille Henry. Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis. *Computer Music Journal*, 2006.

[87] Niall Moody. Brush Strokes VST Plugin. Website: http://www.niallmoody.com/ndcplugs/brushstrokes.htm. Accessed: 6/12/07.

[88] Niall Moody. Particle Fountain VST Plugin. Website: http://www.niallmoody.com/ndcplugs/particlefountain.htm. Accessed: 6/12/07.

[89] Niall Moody. Stimulating Creativity in Music Software. Master's thesis, University of Glasgow, 2004.

[90] Dr. William Moritz. "The Importance of Being Fischinger". In *Ottawa International Animated Film Festival*, 1976. Retrieved from http://www.centerforvisualmusic.org/library/ImportBF.htm, Accessed 17/05/07.

[91] William Moritz. The Dream of Color Music, And Machines That Made it Possible. *Animation World Magazine*, 1 April 1997. Website: http://mag.awn.com/index.php?ltype=search&sval=Moritz&article_no=793 Accessed 17/01/2006.

[92] John Mundy. *Popular Music on Screen: From Hollywood musical to music video*. Manchester University Press, 1999.

[93] Bryan Ochalla. Are Games Art? (Here We Go Again...). Website (Gamasutra Article): http://www.gamasutra.com/features/20070316/ochalla_01.shtml. Accessed: 15/04/08.

[94] Dr. Garth Paine. Gesture and Musical Interaction: Interactive Engagement Through Dynamic Morphology. In *New Interface for Musical Expression*, 2004.

[95] Mark Pearson. Tao. Website: `http://taopm.sourceforge.net`, 31 August 2005. Accessed 18/01/2006.

[96] Craig Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. In *SIGGRAPH*, 1987.

[97] Simon Reynolds. Seeing the Beat: Retinal Intensities in Techno and Electronic Dance Music Videos. Website: `http://stylusmagazine.com/articles/weekly_article/seeing-the-beat-retinal-intensities-in-techno-and-electronic-dance-videos.htm`. Accessed 09/11/07.

[98] A. Wallace Rimington. *Colour-music : the art of mobile colour*. Hutchinson, 1912.

[99] Arnold Schoenberg, Wassily Kandinsky, Jelena Hahl-Koch, and John C. Crawford. *Arnold Schoenberg Wassily Kandinsky: Letters, Pictures and Documents*. Faber and Faber, 1984. trans. John C. Crawford.

[100] Ladan Shams, Yukiyasu Kamitani, and Shinsuke Shimojo. What you see is what you hear. *Nature*, 408:788, 2000.

[101] R.W. Sheppard. Kandinsky's Abstract Drama Der Gelbe Klang: An Interpretation. *Forum for Modern Language Studies*, XI:165–176, 1975.

[102] Julius O. Smith III. Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects. Website: `http://www-ccrma.stanford.edu/~jos/pasp/pasp.html`. Accessed: 17/01/08.

[103] Salvador Soto-Faraco and Alan Kingstone. *The Handbook of Multisensory Processes*, chapter Multisensory Integration of Dynamic Information, pages 49–68. MIT Press, 2004.

[104] Paul Spinrad. *The VJ Book: inspirations and practical advice for live visuals performance*. Feral House, 2005.

[105] .theprodukkt. .kkreiger. Website: http://www.theprodukkt.com/kkrieger. Accessed: 26/09/07.

[106] Richard Thomson. What is The Demo Scene? Website: urlhttp://pilgrimage.scene.org/demoscene.html. Accessed: 05/08/08.

[107] Carol Vernallis. *Experiencing Music Video: Aesthetics and Cultural Context*. Columbia University Press, 2004.

[108] Max Shaw Robert Verplank, Bill Mathews. Scanned Synthesis. In *International Computer Music Conference*, 2000.

[109] Marcelo M. Wanderley. Gestural Control of Music. Website: http://www.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/kassel.pdf. Accessed: 22/06/06.

[110] David Wessel and Matthew Wright. Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*, 2002.

[111] John Whitney. *Digital Harmony: On the Complementarity of Music and Visual Art*. Byte Books, 1980.

[112] Matt Wright. Open Sound Control Home Page. website: http://www.cnmat.berkeley.edu/OpenSoundControl/, 23 June 2004.

[113] Gene Youngblood. *Expanded Cinema*. Studio Vista, 1970.