Chapter 1

A SYSTEM FOR MANIPULATING AUDIO INTERFACES USING TIMBRE SPACES

Craig Nicol
Stephen Brewster and Philip Gray
Glasgow Interactive Systems Group
Department of Computing Science, University of Glasgow, Glasgow, UK
can@dcs.gla.ac.uk
http://www.dcs.gla.ac.uk/~can

Abstract The creation of audio interfaces is currently hampered by the difficulty of designing sounds for them. This paper presents a novel system for generating and manipulating non-speech sounds. The system is designed to generate Auditory Icons and Earcons through a common interface. It has been developed to make the design of audio interfaces easier. Using a Timbre Space representation of the sound, it generates output via an FM synthesiser. The Timbre Space has been compiled in both Fourier and Constant Q Transform versions using Principal Components Analysis (PCA). The design of the system and initial evaluations of these two versions are discussed, showing that the Fourier analysis appears to be better, contrary to initial expectations.

Keywords: user-interface design and specification methods and languages, multimedia interfaces, Auditory Icons, Earcons, Timbre Spaces

1. Introduction

Many authors, for example Gaver (1993) and Mynatt (1994), declare a lack of clear design tools for sounds or auditory interfaces. This paper presents ongoing work on a system to address this need.

The system we are developing will use a more natural interface than the current tools and allow sounds to be described not in terms of their wave properties, but in terms of the sources that produce those sounds, with an advanced level to edit sounds via auditory properties. It is hoped that this system will help sound

designers find useful sounds for their interfaces, and from this, a complete set of design guidelines for sonic interactions can be realized.

The interface will be designed around the timbre space concept typified by work by Hourdin *et al.* (1997a,1997b), who based their work on perceptual models developed in human experiments (Grey, 1977). Sounds will be analysed and loaded into this timbre space where they can be manipulated before being output via a suitable synthesiser.

After a short discussion on why this is an important topic, Section 3 discusses the current level of research in sound and in sonic interfaces, and contains brief discussions of human perception and other auditory interfaces. Section 4 overviews the technology and design behind the current implementation of the system that has been designed, in the context of the work that has been completed. Section 5 provides a quick summary of work to be completed on the project and possible future directions.

2. Overview

As computer displays get smaller on devices such as mobile phones and PDAs, audio interfaces will become even more important for providing information to users. Audio has also been used to enrich a user's information awareness by presenting information non-intrusively (Conversy, 1998).

With these new challenges, new methods of designing and prototyping audio interfaces need to be developed that can be understood not just by experts in music and acoustics, but also by designers with a background in HCI and psychology.

Sound has always been an important part of interacting with the physical world. Sound tells us when someone is coming up behind us, when we have drilled though the wall and when we need to change our spark plugs. Against our rich sonic environment, the computer interface is a poor cousin. Most sounds from our computers are static sounds that do not change to reflect changes in the environment and often have little to do with the events that trigger them.

Gaver noticed this (Gaver, 1993a) and developed a system of Auditory Icons whose sound was related to the action to which they were connected, and whose properties could be adjusted to reflect changes in the underlying environment.

Since then, many people have developed sonic interfaces to various virtual environments and data sets, but each one is distinct, and despite numerous guidelines defining how each type of interface should be designed, and how people will interpret these, there is no common tool for developing or evaluating these sonic interactions.

In the design of Audio Aura (Mynatt et al., 1998) for example, the following guidelines were followed to prevent an "alarm response" in the users:

> "[Background sounds are designed to avoid] sharp attacks, high volume levels, and substantial frequency content in the same range as the human voice (200 - 2,000 Hz)."

The quote is then followed by a note that current audio design often induces this alarm response, intentionally or otherwise.

The computer music community on the other hand has defined many methods for creating and manipulating sounds through a small number of common interfaces, each method having its own strengths and weaknesses.

The most basic musical interface is the MIDI standard, which defines a series of commands that specify musical notes and operations on these such as sustain, pan and instrument used. Short musical segments generated from these commands are known as Earcons (Blattner et al., 1989; Brewster et al., 1993).

The biggest problem with MIDI is that the output sound is not guaranteed. Although there has been some recent standardisation, the basic MIDI specification does not guarantee any particular sound is available on the playback device or that the device will reasonably interpret all the commands. In the most extreme cases, MIDI devices will ignore commands and only play one note at a time, ignoring any other notes requested at the same time.

Another popular field within computer music concerns the transformations made available by the various Analysis-Synthesis techniques, a review of which was done by Masri *et al.*(1997). Analysis-Synthesis allows composers to manipulate sounds directly rather than via the sources that produce them. It allows sounds to be sliced, stretched, reversed and slowly changed into other sounds. It achieves this by presenting the sound in a different format to that used for recording and storage.

Since the aims of the MIDI and Analysis-Synthesis techniques has been musical production, little thought has been given to their perceptual relevance or their use as a design tool for non-musical sounds.

The aim of this project is to go a long way to combining the work in these two areas, allowing interface designers access to complex acoustic and musical methods for developing sound through an interface defined in terms of human perception and design methodologies.

3. Hearing, Analysing and Producing Sound

This section discusses current research in the fields of audio perception and sound analysis. The focus here is on tools or the results of experiments that have been or could be applied to the production of a generic sonic design tool.

3.1 Perception of Sound

There are four components to the way we hear a single sound: the pitch, the loudness, the duration and the timbre. Where sounds are combined, the

relative values of these are important as is the temporal pattern of the sounds. This section concentrates on the issues of single sounds and leaves the sound combination components to the discussion of Earcons in Section 3.3.

The timbre of the sound is what differentiates two sounds whose pitch, loudness and duration are equal. Unlike these other measures, the timbre is a result of complex interactions of frequencies in the sound. The objective measurement of timbre has been a long-running problem in acoustics. Although Helmholtz did a lot of experiments in the 1870's (von Helmholtz, 1877), it has only been with recent advances in signal analysis techniques that timbre has been seriously investigated in papers such as (Grey, 1977) and (Hourdin et al., 1997a).

3.2 Playing Sounds

Once the sound has been designed, an appropriate and efficient output algorithm is needed to play it to the user. In this section, the most common synthesis techniques used on the desktop are discussed.

Sampled Sounds. A sound sample is simply a pre-recorded sound that can be played back at will. Most modern GUIs have support for sound samples that can be triggered in response to a user action such as closing an application. Sound samples are almost identical across a wide range of computers and generally have low performance requirements.

One of the major drawbacks of using sound samples in an audio interface is the size of the files. Even compressed samples are several orders of magnitude larger than the parameter definitions for synthesis algorithms such as FM synthesis or additive synthesis described below. Sampling also requires recording a different sample for every type of interaction we want to simulate. Not only does this require a large amount of storage space, but is also labour-intensive in the sound capture stage and may be impossible if you do not know all possible interactions beforehand.

Wave Synthesis. For sounds generated algorithmically by the machine, the simplest forms of synthesis are based on the manipulation of simple waves. The algorithm will start with one or more simple input waves and modifies them through various filters to create a complex output wave. The input wave is usually a simple sine wave, although others such as sawtooth and square waves are also common.

Waves can be shaped by an *envelope*, which defines the shape of one of the attributes of the wave such as the amplitute. At each point on the wave, the attribute is multiplied by the value of the envelope. We say that the wave attribute has been *convolved* with the envelope. In Figure 1.1 the amplitude of a sine wave has been convolved with the envelope shown.

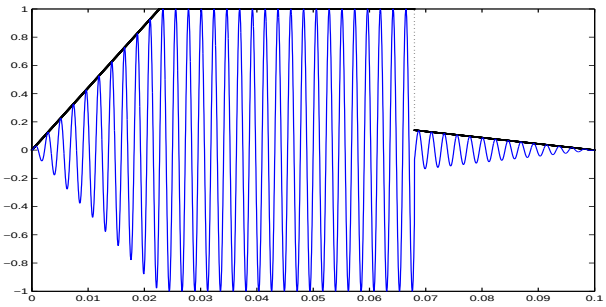

Figure 1.1. A sine wave shaped by an amplitude envelope. The thin line is the wave, the thick line is the envelope.

Additive Synthesis. Additive synthesis is the basis of Gaver's Auditory Icons (Gaver, 1993b). An additive synthesiser generates several waves, giving each wave has its own frequency, phase and amplitude envelopes. The output is the weighted sum of these input waves. Gaver used these to generate contact sounds. The phase and amplitude envelopes are parameterized in order to produce realistic timbre for a variety of real-world contacts.

Additive synthesis, though simple and powerful, is often a slow process as it can require many individual waves to reproduce a complex sound. Synthesis with 10 or more waves is not uncommon. This means that additive synthesis cannot be used in real-time without sacrificing sound quality.

FM synthesis. In the 1970's, Yamaha released a series of keyboards that used Frequency Modulation (FM) synthesis to generate their sounds, as developed by Chowning (1973). Since then, the technique has found its way into many soundcards available on modern machines, although in a more limited form.

For modern computers without a built-in FM synthesiser, the process is fast enough to be computed in real time on the main processor during an interactive session. A simple sine wave known as the carrier has its frequency perturbed by a modulating wave. The effect is to induce a range of tones around the original frequency, creating a complex sound from a small number of input waves. As a consequence, the output of an FM synthesiser is difficult to predict from its inputs, and it is non-trivial to achieve a desired output sound.

The characteristic FM sound is fairly metallic so the sound is then filtered in order to produce a more natural sound. The most useful filter for this purpose is the envelope, which is used to soften the onset and termination of the sound by modifying the amplitude.

3.3 Interfaces and Design

Our system is to be developed for a desktop environment, for situations where the expressiveness and flexibility of the sound development is far more important than accurately recreating a sound from the real world. An example given by Gaver is his Auditory Icon illustrating a file being copied (Gaver, 1993b). In this icon, the expressiveness of the pouring sound he uses is more important than using a less expressive realistic photocopying sound.

The idea behind an auditory interface is that the sounds produced will reflect the current state of the system. In some cases, as in Gaver, this is used to provide feedback and information on user actions. Others, such as Conversy (1998), use the sounds as a non-intrusive way of providing status information on background tasks.

A short review of current audio interface concepts on the desktop follows.

Auditory Icons. Auditory Icons were devised by Gaver (1993b). They are auditory representations motivated by real-world sounds. In his paper, Gaver discusses a variety of sounds designed to resemble tapping, scraping, pouring and other real-world actions. The tapping sounds are used to represent the act of clicking on an icon, the scraping to represent dragging an icon over the desktop and the pouring is used as a progress indicator.

The sounds are parameterized such that different file types are associated with sounds of taps on different materials and the size of the material being tapped represents the file size.

The major problem with Auditory Icons is that the parameterization is a hard problem. Even where the mapping between a perceptual description of a sound and the system state it represents is obvious, it is rarely easy to modify the sound signal in the correct way as standard sound editors operate at the signal rather than the perceptual level. All the icons Gaver presents have been developed after studying the processes that create the sound, which is a slow process.

Earcons. In contrast to Auditory Icons, Earcons (Blattner et al., 1989) do not attempt to describe an event with a real-world sound. Earcons are short musical segments that are abstract representations of a computer process.

Earcons are constructed as patterns of musical notes, where the instrument, duration, pitch, volume and other attributes are modified according to the state of the process. As Earcons are made of many notes, the rhythm and tempo of those notes and their relative volume and pitch are also important attributes that are used in Earcon design.

Unlike Auditory Icons, the connection between state and the sound is arbitrary. Hierarchical Earcons, as used in the experiments by Brewster *et al.* on the effectiveness of Earcons (Brewster et al., 1992), attempt to assign some

structure to Earcons by mapping different attributes to different levels of description of the interface. For example, menus are described by the timbre of the sound, and menu items are described by the rhythm of the sound, providing a consistency across the interface.

Earcons allow a much richer space of sound than Auditory Icons since Auditory Icons are independent of each other and can only be parameterized with respect to simple object interactions, such as a scrape. Earcons can be parameterized with a wide range of musical features as listed above, allowing a single Earcon to present much more information than an Auditory Icon.

Combined approach. A single Earcon is a complex unit formed of many notes. Earcons treat timbre as a single dimension, which is categorical rather than numerical.

Earcons use musical concepts such as pitch, rhythm, duration and tempo as further dimensions, which modify the notes within them and the relative positions of those to reflect changes in the underlying process.

Auditory Icons, however, treat timbre as the combination of many dimensions, and rarely use other dimensions, such as pitch and tempo. A rare example being Gaver's bouncing objects where the temporal proximity of events is an indication of the original height and the springiness of the dropped object.

By combining the complex timbral manipulation of Auditory Icons with the complex combination and musical manipulation of Earcons, we can see that a system that allows control over both timbre and musical dimensions will have a much richer design space than either idea on its own. Whether this richer space will provide a more flexible and useful design space is a matter for investigation.

3.4 Timbre Spaces

To effectively control sound, we need a representation that is flexible enough to allow designers a variety of ways to adapt it. A timbre space is one such representation, and has been chosen for our work because the studies detailed below suggest a link between human perception and a timbre space representation. This implies that designers should find it easier to create a desired sound with a timbre space representation than via traditional synthesis algorithms, where the parameters do not necessarily adapt the sound in the way the designer wishes.

Timbre Spaces. In 1977 Grey published a paper describing an experiment he had done on perception of timbre (Grey, 1977). In this, he played a series of sounds to volunteers and asked them to rate how similar the sounds were. Grey then constructed a 3-dimensional space to represent the sounds, such that each axis represented a property of the sound, for example, he related the first axis to the spectral distribution of energy.

In 1997, Hourdin *et al.* (1997a) first demonstrated an automated way to generate this space, although their space has ten dimensions to Grey's three. They compared their space with that of Grey, showing a correlation between the axes of the two. They then used this space to drive a synthesis system (1997b).

In their automated analysis, a set of input sounds is analysed to produce a sonogram representation where the sound is described by how its frequencies change over time. Taking each frequency as a separate dimension, and the sound as a path through this multi-dimensional space, the sound can be projected into a lower dimensional space known as the *timbre space*, where each dimension represents some higher-level description of the sound based on its frequency components.

There are many possible timbre spaces and each one has its own strengths and weaknesses for different tasks. Each combination of analysis method and dimensionality reduction produces a different space, and other spaces have been used for different tasks, such as for instrument recognition (Kaminskyj, 1999).

Signal Analysis. Signal analysis is the first stage in producing a timbre space. It is the process by which an input signal is broken into its constituent frequencies. There are many different methods for doing this, and good explanations for many of these techniques can be found in Roads (1996) and in Masri *et al.* (1997).

In the analysis, the frequency axis is separated into a number of *frequency bins*, each of which covers a range of frequencies. This range is known as the *bandwidth* of the bin. In general, a smaller bandwidth gives a more accurate frequency representation and a less accurate time representation. The time axis is also split into a number of shorter fragments which may have a window envelope applied, such as a Gaussian bell curve, to smooth the signal over the time segment. Unlike the frequency axis, the time segments can overlap, and this is used to retain any information lost by applying a window envelope.

The two main analysis methods used for this project are the Short-Time Fourier Transform (STFT) and the Constant Q Transform (CQT), both described in Roads (1996). In the STFT, the frequency scale is linear. In the CQT, the frequency scale is logarithmic. In general, the CQT will produce a smaller output but the STFT will compute faster and will have greater resolution across the frequency range.

Figure 1.2 shows how the Flute sound from our dataset is represented in the CQT and STFT forms. As you can see, due to the higher frequency resolution in the CQT presented here, the frequency data is spread across far more frequency bins than in the STFT.

Dimensionality Reduction. The dimensionality reduction techniques presented here take a set of data in n dimensions and create a mapping to convert

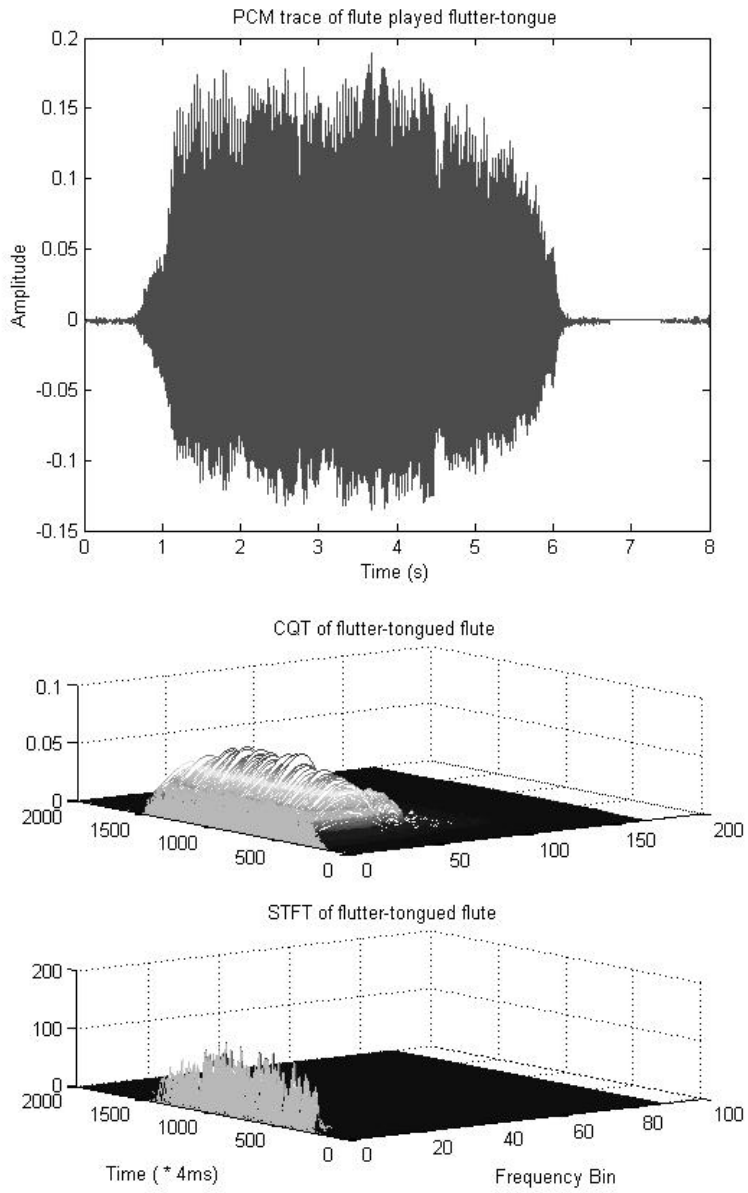

Figure 1.2. One of the 27 sounds used in our Timbre Space as PCM data and as analysed by two different techniques.

the data into m dimensions where m is less than n. The output m dimensions

are chosen to represent the directions of most importance in the data and are ordered such that the first dimension is the most important.

Principal Components Analysis (PCA) is the simplest of the dimensionality reduction techniques and the most susceptible to any noise in the input data. It is the technique used by Kaminskyj (1999) in his timbre space paper. In PCA, the data is rotated such that the direction where the data has the most variance is aligned with the primary axis and the other axes are aligned similarly with the remaining variance in the data.

4. System Details

The complete system we are developing comprises an analysis engine based on the Timbre Space work described above. New sounds presented to the system are converted into paths in this space. The system allows manipulation of these sounds via their path representation. When an output is required, the system maps the path from the Timbre Space onto an FM synthesiser which then outputs the sound to a specified device.

The sound manipulation component of the system allows morphing between sounds in the space. To generate completely new sounds, the paths can be warped into any shape in the space. Certain warps will describe easily heard transformations of the sound and these can easily be coded into the interface, added by the designer by hand or added by comparing two different sounds for the change required to convert one into the other.

This section will now discuss the work and evaluations completed so far on the system and will give preliminary analysis of the results given.

4.1 System Overview

Figure 1.3 shows how data is processed by the system that has been developed. There are 4 data stages and 3 translation stages that map one data type to another. The intermediate data is stored on disk and can be passed between separate devices if required.

The seven stages of the process are:

1 Get sounds (Wave Data)

2 Create timbre space representation of the sound

3 Manipulate path in timbre space (Point List)

4 Create control parameters for the output synthesiser

5 Send parameters to synthesiser (Event List)

6 Run synthesiser over parameter list

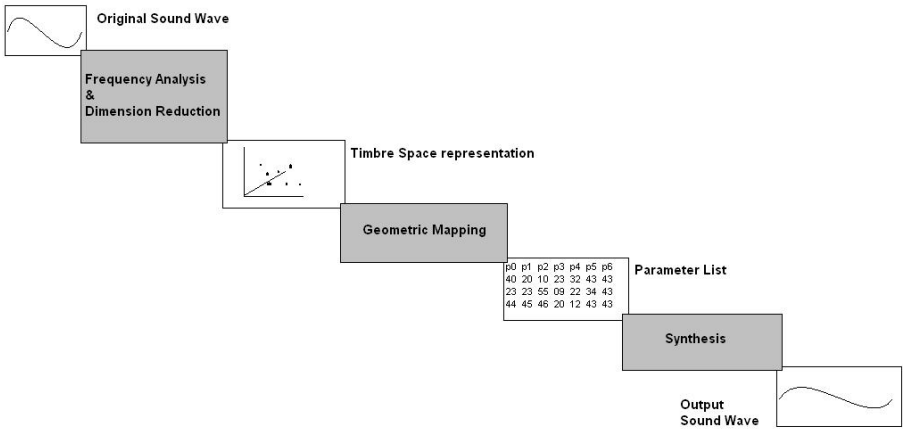

Figure 1.3. Data flow model of the system

7 Output sound to disk or speakers (Wave Data)

Of these, Step 1 and Steps 5-7 are complete. Current experiments are being run to determine the best timbre space to use for Step 2. Preliminary results for these are defined below. Since Step 4 depends on the type of timbre space created, this will be completed after the experiments.

4.2 Analysis of Timbre Spaces

Analysis has been performed on a range of musical and synthetic signals including output from an FM synthesiser and across a selection of sounds selected to match those in Hourdin *et al.*'s experiments. Out of their 40 sounds, 27 have been chosen based on those available to us.

Versions of the STFT and CQT algorithms have been tested with various configurations. The CQT has proved to be quicker than the STFT and produces less output data for the same input, suggesting a greater likelihood of a fixed number of dimensions being able to capture the input signal. Both methods take several minutes to complete the analysis once the input signal is much longer than 400,000 samples (or 9 seconds at a 44.1kHz sampling rate).

A set of functions has been developed to automatically generate a set of timbre spaces that will be compared on their compilation speed, timbre space size and accuracy of sound reproduction. These experiments are ongoing since each timbre space takes as long as 5 days to compile using all 27 input sounds. Once the space is compiled however, it takes under 5 minutes to process each new sound through to its path representation.

The test suite covers CQT and STFT based Timbre spaces with 8 different window conditions, including some where the input is not smoothed by a window before analysis. These are tested against 3 different time resolutions. In the case of the CQT, all these are tested against 3 separate frequency resolutions.

When passed through a Timbre Space, the CQT appears to lose a lot more information that the STFT, producing a lower quality sound. This is contrary to expectations as the CQT produces a smaller output and so has less information to lose. The STFT is much more memory intensive however and will not compile at its highest resolution setting as the resultant output is too large for the PCA algorithm under Matlab with 512Mb system memory and 2Gb virtual memory.

When the STFT is compiled without a window, the consequent PCA compilation takes exponentially longer with the time resoltion. Every other case appears to grow linearly against time resolution. This suggests that there is much less structure to the STFT output when no window is applied, which makes the PCA much less effective in this circumstance.

Once this testing is complete, the complete system will be tested for accuracy against previous Timbre Space work and FM synthesis analysis to ensure the results are perceptually sound.

4.3 Design of Interface

The completed tool will be implemented within a MIDI environment in order to take advantage of existing work on Earcon design. It will accept MIDI signals to control pitch and amplitude and will add an interface to allow real-time editing of timbre. This editing can be performed interactively by a human operator or by another machine process.

The interface will allow selection of any of the pre-selected 27 timbres included in the timbre space as well as any others the user has added to the system.

For each of these preset timbres, a selection of transforms will be made available. These transforms will include morphing between two or more presets, looping the sound within a preset, scaling the pitch of the preset or any other user-defined transformation within the timbre space.

The strength of each transform, or the relative strengths of the timbres affected by the transform, can be controlled by any MIDI controller. This allows the transform to be adjusted over time by any external input and allows the change in the timbre to be recorded in the same place as the change in the melody.

In addition to the preset timbres, the timbre space will also include a series of timbral effects. This will include, for example, ‘Alarm sounds’ as defined by Mynatt. These effects will be defined as a region of timbre space where the effect is greater the closer the path is to the region. When these effects are

used in a transformation, a timbre can be modified to be more like the effect or less like the effect as required. These effects can also be user defined and are expected to be based on psychoacoustic experiments performed by practitioners in that field.

5. Future Plans

With this system complete, a range of experiments will be possible. In particular, sounds can be developed according to both Auditory Icon and Earcon principles such that in any given interaction, the most important information will be mapped to parameters relating to the Auditory Icon portion of the sound (i.e. the note) and subsidiary information will be mapped to the Earcon properties of the sound (i.e. the pattern of notes).

We could perform experiments to see how much of our perceptual space each synthesiser covers in order to decide upon the best synthesiser or configuration to use for any particular sound. If any synthesiser is found to cover a particularly wide or narrow area of this space, this will be a major consideration in its ongoing usage.

6. Conclusions

We have set out to enable designers more flexibility when developing sounds. The project has so far completed the design stage and is currently evaluating the best way to represent the sounds to maximise this flexibility. The Timbre Space has been chosen for this representation due to the perceptual basis afforded to it by the work of Grey. Preliminary results show that the Timbre Space is heavily reliant on the quality of the audio analysis stage. STFT looks to produce the best quality output at this moment, but experiments on other techniques are ongoing.

References

Blattner, M. M., Sumikawa, D. A., and Greenberg, R. M. (1989). Earcons and icons: Their structure and common design principles. *Human Computer Interaction*, 4(1):11–44.

Brewster, S., Wright, P., and Edwards, A. (1992). A detailed investigation into the effectiveness of earcons. In Kramer, G., editor, *First International Conference on Auditory Display*, volume Auditory display, sonification, audification and auditory interfaces., pages 471–498, Santa Fe Institute, Santa Fe, NM. Addison-Wesley.

Brewster, S., Wright, P., and Edwards, A. (1993). An evaluation of earcons for use in auditory human-computer interfaces. In Ashlund, S., Mullet, K., Henderson, A., Hollnagel, E., and White, T., editors, *InterCHI 93*, pages 222–227, Amsterdam. ACM Press, Addison-Wesley.

Chowning, J. (1973). The synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society (JAES)*, 21(7):526–534.

Conversy, S. (1998). Ad-hoc synthesis of auditory icons. In Brewster, S. A. and Edwards, A. D. N., editors, *ICAD 98: Fifth International Conference on Auditory Display*, Glasgow, UK.

Gaver, W. W. (1993a). How do we hear in the world? Explorations in ecological acoustics. *Ecological Psychology*, 5(4):285–313.

Gaver, W. W. (1993b). Synthesizing auditory icons. In *ACM Interchi '93*. ACM.

Grey, J. (1977). Timbre discrimination in musical patterns. *Journal of the Acoustical Society of America*, 64:467–72.

Hourdin, C., Charbonneau, G., and Moussa, T. (1997a). A multidimensional scaling analysis of musical instruments' time-varying spectra. *Computer Music Journal*, 21(2):40–55.

Hourdin, C., Charbonneau, G., and Moussa, T. (1997b). A sound-synthesis technique based on multidimensional scaling of spectra. *Computer Music Journal*, 21(2):56–68.

Kaminskyj, I. (1999). Multidimensional scaling analysis of musical instrument sounds' spectra. In *Australasian Computer Music Conference (ACMC)*, pages 36–9, Wellington, NZ.

Masri, P., Bateman, A., and Canagarajah, C. N. (1997). A review of time-frequency representations, with application to sound/music analysis-resynthesis. *Organised Sound*, 2(3):193–205.

Mynatt, E. D. (1994). Designing with auditory icons. In *CHI '94*, Boston, Massachusetts. ACM.

Mynatt, E. D., Back, M., Want, R., Baer, M., and Ellis, Jason, B. (1998). Designing audio aura. In *CHI '98*. ACM.

Roads, C. (1996). *The Computer Music Tutorial*. Massachusetts Institute of Technology.

von Helmholtz, H. L. F. (1877). *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Dover (1954), New York, 4th edition.