Synthesising timbres and timbre-changes from adjectives/adverbs

Alex Gounaropoulos and Colin Johnson

Computing Laboratory
University of Kent
Canterbury, Kent, CT2 7NF
England
ag84@kent.ac.uk, cgj@kent.ac.uk

Abstract. Synthesising timbres and changes to timbres from natural language descriptions is an interesting challenge for computer music. This paper describes the current state of an ongoing project which takes a machine learning approach to this problem. We discuss the challenges that are presented by this, discuss various strategies for tackling this problem, and explain some experimental work. In particular our approach is focused on the creation of a system that uses an analysis-synthesis cycle to learn and then produce such timbre changes.

1 Introduction

The term *timbre* is used in various ways in music. One way is in describing gross categories of sounds: instrument types, the sound of certain combinations of instruments, different stops on a pipe organ, a discrete choice of sounds on a simple electronic keyboard, and so on.

A second aspect of timbre is the distinctive sound qualities and changes in those qualities that can be produced within one of those gross categories. To a skilled player of an acoustic instrument, such timbral adjustments are part of day-to-day skill. A notated piece of music might contain instructions concerning such timbres, either in absolute terms ('harshly', 'sweetly') or comparative terms ('becoming reedier'), and musicians use such terms between each other to communicate about sound ('Can you sound a little more upbeat/exciting/relaxed').

From here onwards we will denote these two concepts respectively by the terms *gross timbre* and *adjectival timbre*.

The player of a typical electronic (synthesis-based) instrument does not have access to many of these timbral subtleties. Occasionally this is because the synthesis algorithms are incapable of producing the kinds of changes required. However in many cases this lack of capability is not to do with the capacity of the synthesis algorithm—afterall, a typical synthesis algorithm is capable of producing a much larger range of sound changes than a physically-constrained acoustic instrument—but to do with the *interface* between the musician and the instrument/program [11, 15]. In current systems, the know-how required in order to

effect the timbral change suggested by an adjectival description of timbre-change is vast.

Providing tools for manipulating timbre is an underexplored problem in computer music. In this paper we will discuss ongoing work on a project that aims to combine machine learning methods for searching synthesis parameter space and classifying timbre, together with analysis methods such as spectral analysis and principal component analysis. The long-term aim of this project is to produce systems that:

- Allow the synthesis of timbral changes to a sound from natural language descriptions of the desired change.
- Facilitate the automated discovery of transformations in synthesis parameter space that have meaningful timbral effects in sound space.
- Providing a framework whereby advances in the computer-based analysis of timbre can be used automatically to synthesise timbre and timbral changes.

2 Approaches to Timbre

2.1 Theory and Notation of Timbre

Compared to other aspects of music such as pitch and rhythm, timbre is not well understood. This is evidenced in a number of ways. For characteristics such as pitch and rhythm, there exist theories of how they work and produce perceptual effects; there are well-understood notations for them; and we understand how to synthesize them from fundamental components to get a particular effect.

By contrast, timbre lacks this repertoire of theory and notational support (as explored by Wishart [19]). Nonetheless there is a large repertoire of language associated with timbre and timbral changes. These timbral adjectives and metaphors provide a powerful means for musicians to communicate between themselves about timbre; but by contrast to the more formal notations for, say, pitch or rhythm, they do not provide a usable structure for inputting desired timbres or timbral changes into computer music systems [2, 11, 18, 3].

One approach would be to come up with a new language for timbre, which is more closely aligned with the way in which timbre is generated in electronic instruments. However this has many problems. For example timbre words convey information that has musical meaning, and we would like to create systems so that electronic and acoustic instruments can be played side-by-side and the players able to communicate using a common vocabulary. For these reasons we focus on how we can use traditional timbre words in a computer music setting.

2.2 Timbre as Gross Categorisation

At the beginning of this paper we introduced two notions of timbre: timbre as a gross categorisation of sounds, and timbre as the differences in sound qualities within those gross categories.

These two aspects of timbre are very different; most of the literature on timbre in computer music has focused on the gross categorisation, beginning with the early of Wessel [17].

An example of studies of gross timbre is the work of McAdams *et al.* [9]. In this work three dimensional timbre space was defined, the dimensions being attack time (time taken for volume of a note to reach maximum), the spectral centroid (the relative presence of high frequency versus low-frequency energy in the frequency spectrum), and spectral flux (a measure of how much the spectral changes over the duration of a tone). A number of instruments where then analysed by these three techniques and a graph of the results showed how each occupied a different part of the timbre space.

Such representations are useful when the aim is purely *analytic*, i.e. we want to understand existing sounds. However the aim of our work is oriented towards *synthesis*, and so we need to consider what representation is appropriate for being used 'backwards' to go from analysis to synthesis. Whilst a representation such as the three-dimensional model in [9] might yield acceptable results for categorising sounds, this representation is not adequate for synthesis of sound. We certainly could not work backwards from a three dimensional timbre representation and hope to synthesise the starting sound, since the representation is oversimplified and too much information has been lost.

Much of the recent work in the area of gross timbre has focused on the developing MPEG-7 standard. This work defines a framework for describing sounds in terms of spectral and temporal measurements of the sound, extracted through some analysis method. This work is interesting in that it identifies a number of features that are proven to be important for recognition and perception of timbre based on past research.

A large proportion of other research into timbre in computing has focused on automated recognition or categorisation of instruments. For example Kostek [8] describes a system that uses machine learning methods to classify which instrument is playing.

This has possible applications in databases of music for automated searching of stored sounds. The automated approach eliminates the need for a human to enter metadata identifying each sound, thus greatly simplifying the process of creating large sound databases. The common approach is to use neural networks to learn the sound of various instruments after being presented with various recordings of each. The key in this sort of work is to find common features between different recordings of a certain type of instrument, where the recordings may have different pitches, loudness, or playing style. Such features may be specifically symbolically represented, e.g. if the classification is performed using a decision tree method; or, they may be subsymbolically represented e.g. if a neural network was used.

Analysis of real instruments reveals that the tone of a single instrument can vary greatly when pitch is changed, or with changes in the volume of the playing. Therefore, the challenge in gross timbre identification is to identify the

common features of the sound that identify a certain instrument, despite the large variations in tone that can be produced.

2.3 Timbre Analysis for Adjectival Timbre

A different body of work focuses on the concept of adjectival timbre. Here, the focus is not on studying the sound of an instrument as a whole, but on looking at individual generic characteristics of sounds such as brightness, harshness, or thickness. Early work on this was carried out by Grey [5], who identified some features of a synthesis algorithm which correlate strongly with timbral characteristics.

There are many studies in the field of psychoacoustics where experiments have been carried out to identify salient perceptual parameters of timbre. This experiments have usually taken the form of listening tests where volunteers have produced verbal descriptions of sounds, and the results are analysed to find correlations in the language used. This is useful as it identifies adjectives that are important for describing sounds, and this could form the basis for the types of perceptual features we might aim to control in the synthesiser program we are developing. However, these psychoacoustic experiments by themselves are not enough in order to synthesise the given perceptual features, since we also need to find a correlation of an adjective with certain spectral and temporal features of a sound; then more specifically with the parameters within a specific synthesis algorithm that give rise to those timbres or timbral changes.

The *SeaWave* project [2] made some progress in finding these correlations. Certain spectral and temporal features of starting sounds where modified, and the perceived changes in timbre where recorded. Some correlations where found and these were used to develop a synthesis system where certain timbral features such as resonance, clarity, or warmth could be controlled. The number of adjectives that where available to user to control the sound where limited, suggesting that more a much more extensive study of synthesis parameters and their perceptual correlates is needed.

It is interesting to note that while machine learning techniques have been used for automated classification of different instruments, it does not appear that a general system has been developed for automatically identifying adjectives that describe a certain sound. The small amount of work that has been carried out in this area has focused on specific domains. For example a recent paper by Disley and Howard [1] is concerned with the automated classification of timbral characteristics of pipe organ stops. It does not appear that any work has been carried out on automated classification of timbral *differences* between pairs of sounds.

2.4 Synthesis of Timbre

The most limited range of work to date has been on the automated *synthesis* of timbres or timbral changes.

Some work has been done on the automated synthesis of gross timbre. Clearly it is not possible to synthesise gross timbre from just words, but machine learning methods can be applied to learn synthesis parameters for a particular instrumental sound. In these cases the learning is guided either by interaction with a human [7, 10] or by the comparison of spectral features between the synthesized instrument-candidates and recordings of real instruments [20].

Of greater interest the this project is the automated synthesis of adjectival timbre. There are two basic concepts: associating adjectives/adverbs and classifications with timbres ('wooden', 'bright'), and words which are describe characteristics that sit on a timbral continuum ('can you play less reedily please?', 'let's have some more bounce'). A preliminary attempt to create a dictionary of such timbre-words, and to group them into classes, was attempted by Etherington and Punch [2].

A small amount of work has attempted to do automated synthesis of sounds from timbral descriptions. The *SeaWave* system [2] is based on a number of listening experiments which attempt to match specific sets of transformations of sound signals with words that describe those transformations. This works well up to a point; however the transformations required to achieve many kinds of transformations are likely to be complex, requiring more than simply the increase or decrease of a couple of synthesis parameters; and also they will typically be dependent on the starting sound.

Another attempt to generate quasi-timbral aspects of sound is given by Miranda [11]. He made use of machine learning methods to deduce correlations between parameters that could be used in synthesis and their perceived effects. This was then used to build up a database of matches between descriptive terms and characteristics which are used in synthesis; when the user requests a sound these characteristics are looked up in the database and a synthesis algorithm called with these characteristics. This provides a powerful methodology for generating sounds *ex nihilo*; however it was not applied to transforming existing sounds.

Since these two groundbreaking pieces of work, there appears to be no further work on linking linguistic descriptions of adjectival timbre to synthesis.

3 Complex Mappings: a Challenge for Timbre Exploration

One of the main difficulties with synthesis of timbres from descriptions is the complex nature of the mapping from the parameter space of a synthesis algorithm to the space of sounds, and then to the space of features that are described when we hear sounds (a more general exploration of such complexities in AI is given by Sloman [16]). Typically, many different closed subsets in parameter space will map onto the same timbre adjectives. Furthermore, features of timbre are influenced by previous exposure. For example, we are familiar with 'wooden' and 'metallic' sounds, and believe these to be contrasted; however in a synthesis algorithm it is possible to realise sounds that are physically unrealistic,

e.g. sounds which are 'between' wooden and metallic, or which have both such characteristics.

This complexity contrasts with, say, loudness, where the mapping from the parameter (amplitude) to the perceptual effect (loudness) is straightforward. This presents a challenging problem for interfaces for timbre [15]; the timbral equivalent of the volume knob or piano keyboard is not obvious, nor is it obvious that such an interface could exist.

4 Experiments Timbre Synthesis via Machine Learning

So far in this paper we have discussed work on the automated *analysis* of timbre, and on the *synthesis* of timbre. However there has been little progress in combining the results of these two approaches. in the remainder of this paper we present experimental work which attempts to combine these two ideas.

4.1 Approaches

There are basically two approaches to this problem. One is an *analytic* approach, where we work out directly how changes in the production of a sound lead to changes in the perceived timbral properties of the sound. The second, which we have used in our work, is a *machine learning* approach, where we take many examples of sounds, associate human-written metadata about timbre with them, and then apply machine learning methods [13] to create the relevant mappings.

An initial, somewhat naïve, perspective on this is to view it as being an *inverse problem*. That is, we take a set of sounds (recorded from acoustic instruments) that demonstrate the desired timbres (or timbral changes), and analyse what characteristics are common between the sounds that fit into an similar timbre-class. Then we apply analysis methods (e.g. spectral analysis) to these sounds to understand what characteristics of the sound 'cause' the different perceived timbral effects, and then apply these same characteristics to our desired sound to transform it in the same way.

However there are (at least!) two problems with this naïve model. Firstly, it is usually difficult to extract the characteristics that characterise a particular timbre or change of timbre. We can analyse sounds in many different ways, and not all of the characteristics that we can extract will be relevant to the production of a particular timbre. Even if we can eliminate some features by removing those characteristics that are common to sounds in various classes, there may be some arbitrary features that are irrelevant.

A second difficulty is found in the final stage of the process. Even if we can isolate such a characteristic, it is not easy to apply this to another sound: sometimes the characteristic can be difficult to express within a particular synthesis paradigm, and even if we can apply it, changing that characteristic in synthesis parameter space will not have the same effect in perceptual space. An additional problem of this kind is that, even when the changed sound is available within the synthesis algorithm being used, finding an appropriate change of parameters to effect the timbral change can be difficult.

4.2 System Overview

In our system we have tackled this problem in this indirect fashion, whilst avoiding the naïve approach of inverting the mapping. An overview of the program is given in figure 1.

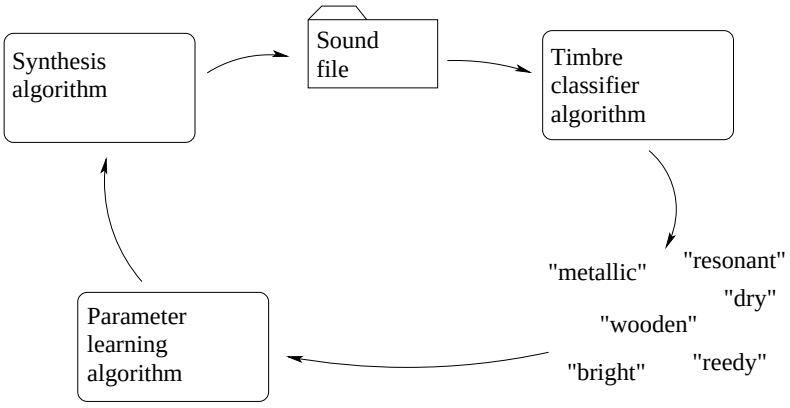

Fig. 1. Indirect learning of timbral change.

An initial stage of the process consists of training a timbre classification algorithm. This takes a training set of many acoustic sound samples that have been hand-annotated with timbral metadata, and uses those to train a classifier which will sort sounds into relevant classes based on the timbre-words of interest. Some initial experiments with a neural network based classifier have proven promising.

The other main stage of the process consists of learning the parameter choice to feed into the synthesis algorithm. Sounds are generated using a synthesis algorithm (the algorithm used remains fixed throughout the process). The resultant sound files are then fed into the trained timbre classification algorithm, which outputs a measure of how strongly the sound fits into each timbral class.

This measure is then used as a quality measure (e.g. a fitness measure in a genetic algorithm [12]) in a machine learning algorithm. This could be used in a number of ways. One way would be to learn the values of certain input values in the synthesis algorithm which then remain fixed (in a similar fashion to [20, 14]). Another would be to learn which parameter changes (or characteristics characterised by covarying parameter changes, as in attribute construction in data mining [4]) are important in making particular perceived timbral changes.

The remainder of this section consists of a description of these two core components of the system.

4.3 Timbre Classification

In order for our learning system to be able to create a sound with the desired timbre, we need to be able to test the fitness of each solution that the system proposes. Of course, this needs to be an automated process, so our solution is to use a neural network capable of recognising certain timbral features in a sound.

Firstly, some pre-processing is carried out on the input sound wave in order to greatly reduce the complexity of the data, and therefore make it suitable for use as input to a neural network. Spectral analysis is carried out on the audio using an FFT, and the partials making up the sound are extracted from this, including the amplitude envelope and tuning information for each partial. From this data, a set of 20 inputs for the neural network is generated. Inputs 1-15 are the peak amplitude of each of the first 15 partials of the sound, which should describe the general 'colour' of the timbre. The next input is the average detuning of the partials, which describes how much the tuning of the partials differs from a precise harmonic series. The remaining inputs describe the overall amplitude envelope of the sound, and are attack time (time taken for the amplitude to reach its peak), decay time (time from the peak amplitude to the end of the note), and finally attack and decay slopes (rate of change in amplitude) which describe the general shape of the amplitude envelope.

The aim of the neural network is to map a set of inputs onto a set of values describing the timbral features present in the sound. In order to define the expected output of the network in our prototype, samples of notes from 30 (synthesised) instruments were collected, and 9 adjectives where chosen to describe the timbre of each instrument (bright, warm, harsh, thick, metallic, woody, hit, plucked, constant amplitude). Listening tests where carried out on each instrument sample, and values ranging from 0 to 1 were assigned indicating how well each adjective described the instrument (a value of 1 meaning the particular feature was strongly present in the sound, while 0 indicating that the adjective did not apply to the sound at all). This work decided that our neural network would have 9 outputs onto which the inputs are mapped.

An application (figure 2) was developed that takes a list of sound files and their associated values for each adjective, carries out the necessary pre- processing to generate a set of inputs, and then trains a neural network to associate the correct adjectives with each sound's input data. A 3-layer back-propagation network was used, with 100 neurons per layer (this value was chosen empirically and gave reasonable training times as well as better generalisation than was achieved with smaller networks). Once the network is trained, the application allows the user to select an instrument sound that was not included in the training data, and run it through the system to classify its timbre.

4.4 Timbre Shaping

The second main part of the process is shaping the timbre, i.e. adjusting the parameters of the synthesis algorithm to produce the desired timbral characteristics. A screenshot of this program is shown in figure 3. The system uses

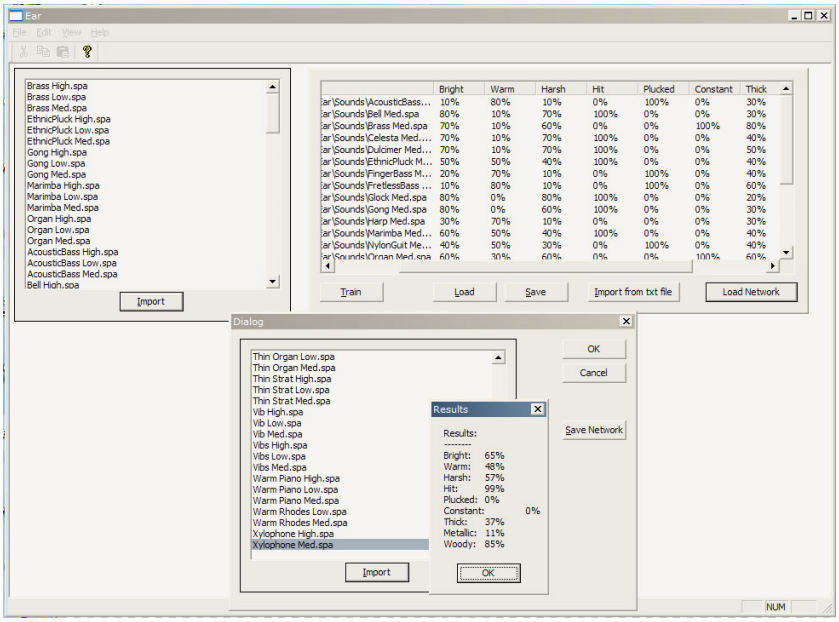

Fig. 2. Timbre recognition process

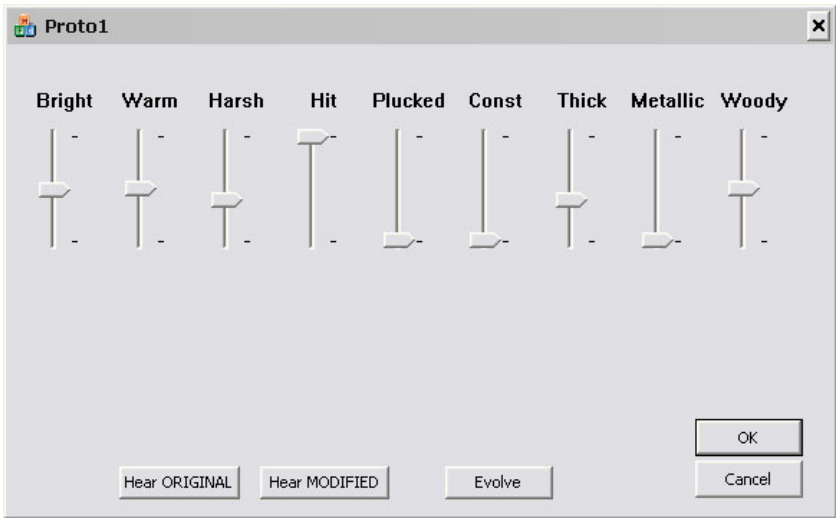

Fig. 3. The timbre shaping program

an additive synthesis algorithm, using the parameters described in the previous section for synthesis rather than analysis.

In initial experiments, the system used a genetic algorithm (similar to [7]) to explore the parameter space and find a sound with the desired characteristics. The neural network timbre classifier was used to calculate a fitness for each solution in the population. However, tests of the program showed that the genetic algorithm had difficulty in finding suitable solutions, and did not show signs of converging towards a good solution. This is probably due to the sheer complexity of the mapping between synthesis parameters and the description of the timbre that they produce.

We have developed an alternative algorithm that is much more successful at finding the synthesis parameters for the desired timbre. It is designed to make use of the information that stored in the timbre classification neural network in order to search through the synthesis parameter space. The algorithm is similar to the back-propagation method used in training neural networks. Firstly, we take an arbitrary set of input values as a starting point and feed them into the neural network. These input values represent synthesis parameters. We run the network in order to obtain a set of results which represent a description of the timbre that would be produced. An error value for each output is then calculated, based on a comparison between the desired output and the actual output of the network. This error is then passed back through the network just as it is in the back-propagation algorithm, the only difference being that we do not actually modify any of the weights in the neuron connections. The error eventually propagates down to the inputs, therefore telling us how we need to adjust each input in order to obtain a better solution. The process then repeats until the overall error rate drops to an acceptable amount

There are clear reasons why this algorithm is more successful at finding a solution than the genetic algorithm. When using a genetic algorithm, each proposed solution is given a single fitness value which reflects how well it solves the problem. This means that we have no way of knowing how good each parameter in the solution is, since we can only judge the solution as a whole. Our algorithm however, gives us a separate error value for each parameter that makes up the solution, allowing us to move towards a good solution more effectively. Our algorithm makes use of the knowledge about timbre that is contained in the neural network, whereas with the genetic algorithm the genetic operators did not have sufficient power in combining and making small changes to timbral characteristics.

5 Results

5.1 Results of the Timbre-Classification Algorithm

The results of the timbre recognition process are presented in table 1. This shows a comparison between the timbral characteristics of five sounds, classified by a list of adjectives and a value indicated how strongly each characteristic was detected in the sound, by both a human listener and the neural network.

Instrument	Bright	Warm	Harsh	Thick	Metallic	Woody	Hit	Plucked	Constant Amplitude
Vibraphone	0.6	0.5	0.4	0.4	0.5	0.3	1.0	0.0	0.0
	0.6	**0.8**	**0.4**	**0.3**	**0.1**	**0.2**	**1.0**	**0.0**	**0.0**
Elec. Guitar	0.7	0.2	0.7	0.2	0.4	0.1	0.0	1.0	0.0
	0.3	**0.6**	**0.7**	**0.4**	**0.2**	**0.3**	**0.6**	**0.4**	**0.0**
Piano	0.6	0.5	0.1	0.6	0.2	0.3	1.0	0.0	0.0
	0.7	**0.4**	**0.3**	**0.6**	**0.0**	**0.2**	**1.0**	**0.0**	**0.0**
Xylophone	0.8	0.3	0.7	0.1	0.0	0.8	1.0	0.0	0.0
	0.7	**0.5**	**0.6**	**0.4**	**0.0**	**0.9**	**1.0**	**0.0**	**0.0**
Elec. Piano	0.5	0.5	0.2	0.4	0.2	0.2	1.0	0.0	0.0
	0.2	**0.9**	**0.1**	**0.1**	**0.1**	**0.2**	**1.0**	**0.0**	**0.0**

Table 1. The table first shows the expected value from a user listening test, followed by the neural network's actual answer in bold. A value of 1.0 indicates that a feature is strongly present in the sound, whereas a value of 0.0 indicates that the feature is absent.

Early results from our timbre classification system are encouraging. In the experiment, a single human listener first assigned values to describe the timbre of the five test sounds, then the neural network was used to obtain a classification. The test sounds of course had not been used as part of the training set for the network. The results table shows that the prototype at this stage generally works well. There is evidence that the system is extracting common timbral features across different types of instrument sounds. It is particularly successful in detecting harshness, or sounds that are 'woody', or sounds that are hit. Unsurprisingly, it has trouble distinguishing between hit or plucked instruments, which is to be expected since the network's input data contains no information about the spectrum of the sound's attack portion, which is known to be significant in recognising sound sources.

5.2 Results of the timbre shaping algorithm

Some of the results from the timbre-shaping process can be heard at
`http://www.cs.kent.ac.uk/people/staff/cgj/research/evoMusArt2006/evoMusArt2006.html`

6 Future Directions

There are many future directions for this work. Some of these are concerned with timbre recognition, for example using ear-like preprocessing (as in [6]) of sound to generate the inputs to the neural network. We will also carry out more extensive human trials of the timbre-recognition experiments.

There are many future directions beyond this. A major limitation of many attempts at automated synthesis of timbre or timbre change is that the learning

has been applied to a single sound. Future work will focus on learning transformations of synthesizer parameter space with the aim of finding transformations that will apply to many different sounds.

References

1. A.C. Disley and D.M. Howard. Spectral correlates of timbral semantics relating to the pipe organ. *Speech, Music and Hearing*, 46, 2004.
2. Russ Etherington and Bill Punch. SeaWave: A system for musical timbre description. *Computer Music Journal*, 18(1):30–39, 1994.
3. Rosemary A. Fitzgerald and Adam T. Lindsay. Tying semantic labels to computational descriptors of similar timbres. In *Proceedings of Sound and Music Computing 2004*, 2004.
4. Alex A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer, 2002.
5. John M. Grey. *An Exploration of Musical Timbre*. PhD thesis, Stanford University, Department of Music, 1975.
6. David M. Howard and Andy M. Tyrrell. Psychoacoustically informed spectrography and timbre. *Organised Sound*, 2(2):65–76, 1997.
7. Colin G. Johnson. Exploring the sound-space of synthesis algorithms using interactive genetic algorithms. In Geraint A. Wiggins, editor, *Proceedings of the AISB Workshop on Artificial Intelligence and Musical Creativity, Edinburgh*, 1999.
8. Bożena Kostek. *Soft Computing in Acoustics*. Physica-Verlag, 1999.
9. S. McAdams, S. Winsberg, S. Donnadieu, G de Soete, and J. Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, 58:177–192, 1995.
10. James McDermott, Niall J.L. Griffith, and Michael O'Neill. Toward user-directed evolution of sound synthesis parameters. In Rothlauf et al., editor, *Applications of Evolutionary Computing*, pages 517–526. Springer, 2005.
11. Eduardo Reck Miranda. An artificial intelligence approach to sound design. *Computer Music Journal*, 19(2):59–75, 1995.
12. Melanie Mitchell. *An Introduction to Genetic Algorithms*. Series in Complex Adaptive Systems. Bradford Books/MIT Press, 1996.
13. Thomas Mitchell. *Machine Learning*. McGraw-Hill, 1997.
14. Janne Riionheimo and Vesa Välimäki. Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation. *EURASIP Journal on Applied Signal Processing*, 8:791–805, 2003.
15. Allan Seago, Simon Holland, and Paul Mulholland. A critical analysis of synthesizer user interfaces for timbre. In *Proceedings of the XVIIIrd British HCI Group Annual Conference*. Springer, 2004.
16. Aaron Sloman. Exploring design space and niche space. In *5th Scandinavian Conference on AI*. IOS Press, 1995.
17. David M. Wessel. Timbre space as a musical control structure. *Computer Music Journal*, 3(2), 1979.
18. Trevor Wishart. *Audible Design*. Orpheus the Pantomime, 1994.
19. Trevor Wishart. *On Sonic Art*. Harwood Academic Publishers, 1996. second edition, revised by Simon Emmerson; first edition 1985.
20. Jennifer Yuen and Andrew Horner. Hybrid sampling-wavetable synthesis with genetic algorithms. *Journal of the Audio Engineering Society*, 45(5):316–330, 1997.