Real-time Timbre Remapping with Differentiable DSP

Jordie Shier
Centre for Digital Music
Queen Mary University of London, UK
j.m.shier@qmul.ac.uk

Charalampos Saitis
Centre for Digital Music
Queen Mary University of London, UK
c.saitis@qmul.ac.uk

Andrew Robertson
Ableton AG
Berlin, Germany
andrew.robertson@ableton.com

Andrew McPherson
Dyson School of Design Engineering
Imperial College London, UK
andrew.mcpherson@imperial.ac.uk

ABSTRACT

Timbre is a primary mode of expression in diverse musical contexts. However, prevalent audio-driven synthesis methods predominantly rely on pitch and loudness envelopes, effectively flattening timbral expression from the input. Our approach draws on the concept of timbre analogies and investigates how timbral expression from an input signal can be mapped onto controls for a synthesizer. Leveraging differentiable digital signal processing, our method facilitates direct optimization of synthesizer parameters through a novel feature difference loss. This loss function, designed to learn relative timbral differences between musical events, prioritizes the subtleties of graded timbre modulations within phrases, allowing for meaningful translations in a timbre space. Using snare drum performances as a case study, where timbral expression is central, we demonstrate real-time timbre remapping from acoustic snare drums to a differentiable synthesizer modeled after the Roland TR-808.

Author Keywords

Differentiable Digital Signal Processing, Timbre Remapping

CCS Concepts

•**Applied computing** → **Sound and music computing;** Performing arts;

1. INTRODUCTION

Timbre is a musical concept that has distinctly resisted precise definition in psycoacoustics and music psychology research [31]. It has been referred to as the "psychoacoustician's multidimensional waste-basket category for everything that cannot be labeled pitch or loudness" [37, p.34]. Yet, within that multidimensional waste-basket resides a rich landscape of musical expression. Timbre is central to many musical traditions including Western classical music [36], electronic music [46], and diverse percussion traditions including tabla [7] and drum kit performance [9]. The sound synthesizer has played a particularly important role in expanding the timbral palette of musicians and entirely new musical cultures have formed around their use [2]. Timbre perception and synthesizers share a rich history, evidenced by the significant body of literature following Wessel's introduction of the timbre space as a control interface in 1979 [62]. Furthermore, deep learning has enabled novel timbre-focused synthesis tasks including timbral control [47] and musical timbre transfer [25, 13].

Timbre transfer has received considerable attention in recent years and refers to the task of altering "timbral information in a meaningful way while preserving the hidden content of performance control" [8, p.4]. One popular formulation of this task was introduced by Engel et al. [13] and involves learning a mapping from pitch and loudness control signals to the timbre of a target instrument, expressed as time-varying harmonic amplitudes. In other words, pitch and loudness are explicit control signals and timbre is implicitly learned, conditional on time-varying pitch and loudness. The side-effect of this technique is that timbral expression is effectively ignored in the input signal. Whilst this may suit musical contexts where pitch and loudness are the primary parameters of expression, what about the myriad contexts where this is not the case? In this paper we propose a novel, data-driven formulation for timbral control of synthesizers using differentiable digital signal processing (DDSP) [13], aiming to address musical contexts where timbre is a primary vehicle for musical expression.

We draw on the idea of timbre remapping, introduced by Stowell and Plumbley [54], who defined the task as mapping trajectories between two distinct timbre spaces. This concept bears similarity to the idea of timbre analogies [62, 38], that is, transpositions of sequences within a timbre space, and provides a conceptual starting point for our design.

Instead of learning to match absolute audio feature values, as is typical in AI-based audio synthesis, we propose to learn to match relative differences in audio features. This design decision is motivated by the role of timbre in the structuring of musical phrases [36] and the importance of subtle, graded timbral differences [61]. It is the relationship between the timbres of neighboring musical events that is important in the creation of a musical phrase, not the absolute values of each individual event taken in isolation. This is exemplified by drummers intentionally varying the intensity and timbre of certain hits *within* a groove to provide juxtaposition and indicate rhythmic intention [9].

NIME'24, 4–6 September, Utrecht, The Netherlands.

To this end, we present a *feature difference loss* function that considers pairs of sounds and differences between their features as opposed to absolute values in isolated notes. Paired with a differentiable synthesizer and a gradient-based optimizer, this enables us to learn to adjust synthesizer parameters to create timbral differences analogous to those observed between successive events in a musical passage – remapping the timbre from a musical passage onto a synthesizer.

As a case study, we consider the musical context of a snare drum performance [9] and demonstrate how this method can be used for timbre remapping from acoustic snare drums to a differentiable synthesizer modeled after the Roland TR-808 snare drum. An open source audio plug-in implementing the real-time system and training scripts are presented alongside this paper to allow musicians to experiment in their own musical contexts. Reflections on a session with a professional drummer are provided and point to both the effectiveness of this approach as well as areas for future improvement. Recordings from this session and software are available on a supplementary website[1].

2. BACKGROUND

The work presented in this paper fits into the broader landscape of work focused on the development of novel controllers and mappings for synthesizers [41]. It also contributes to the growing body of literature in NIME on machine learning for musical expression [29].

2.1 Timbre Spaces and Timbre Analogies

The perceptual foundation of this work is the timbre space, a multi-dimensional representation of sounds, derived from listening studies on perceived similarity [18] and the lineage of work that followed, seeking acoustic correlates with timbre perception [19] (see [35] for an overview). The MPEG-7 standard [26] defined a set of audio features to quantify timbre, although a multitude of other features have been proposed [44] and are commonly used. The concept of timbre analogies was first explored by Ehresman and Wessel in 1978 [12] and described transpositions of sequences within a perceptual timbre space. Wessel subsequently discussed timbre analogies in the context of musical control with timbre spaces [62] and McAdams and Cunible verified the perceptual viability of timbre analogies [38]. Perceptual studies performed by these researchers utilized pairs of stimuli and asked whether participants were sensitive to relative differences in timbre. More concretely, given a pair of sounds $\mathbf{x}_a$ and $\mathbf{x}_b$, participants were asked to select a sound $\mathbf{x}_d$ (from a set of choices) that differed from $\mathbf{x}_c$ by the same amount as $\mathbf{x}_b$ differed from $\mathbf{x}_a$. Results showed that $\mathbf{x}_d$ could be predicted by a parallelogram model of similarity within a perceptual timbre space. From a geometric perspective, this indicates that timbral sequences could be represented as vectors in a multidimensional timbre space and translated within that space while preserving the perception of relative difference. While the perception of these relative differences was not as strong as pitch and depended on the nature of the relationship [38], it suggests that with the correct perceptual scalings, translations of timbre are viable musical operations – an idea we build upon in this work.

2.2 Perceptual Control of Synthesizers

Fasciani defines a synthesis method as being perceptually related when it explicitly manipulates timbral attributes of the generated sound [16]. Timbre remapping can be considered an example of a perceptual control method as it seeks to explicitly manipulate timbre by mapping attributes from an input source to the generated sound. Stowell and Plumbley [54] introduced this concept and identified the difficultly of the task when the distribution of timbres differ between the control and target contexts. As a practical example, they presented the task of controlling a concatenative synthesizer using an audio signal, building on the work of Schwarz [50]. A key contribution by Stowell and Plumbley [54, 55], [53, Chapter 5] is the recognition of the context-dependent nature of timbre and multidimensional interactions between various features. They proposed an unsupervised regression tree method to learn associations between the distinct timbre spaces of the input control and synthesizer.

Another line of work considers the timbre space directly as musical control structure. Building on Wessel's 1979 paper [62], several researchers have explored computational methods for navigating timbre spaces with respect to synthesis parameters [24, 17, 16, 52, 59]. Timbral exploration methods are often motivated to cover the space of all possible sounds of a synthesizer to support searching; however, this can make subtle timbral variations more challenging to achieve [17]. It is these subtle timbral variations that we turn our attention to in this work – "graded timbral differences" that contribute to the perception of a continuous musical phrase [61].

2.3 Differentiable Digital Signal Processing

Differentiable digital signal processing (DDSP) was introduced by Engel et al. alongside an implementation of a harmonic plus noise synthesizer for modeling monophonic and harmonic instruments [13]. DDSP enables the integration of DSP algorithms directly into neural network training regimes, allowing for loss functions to be computed directly on generated audio as opposed to parameter values, better representing the complex relationship between the auditory and parameter space [14]. Following the initial DDSP paper, a large body of work on audio synthesis has followed exploring numerous synthesis methods including waveshaping [21], FM [5, 63], subtractive [34], and filtered noise synthesis [1]. Of particular relevance is the DDSP timbre transfer task [13, 4], which follows naturally from the choice of pitch and loudness as control signals. The timbre of an instrument learned during training, represented by time-varying harmonic amplitudes, can be mapped onto pitch and loudness contours of a different instrument during inference. In contrast to the DDSP timbre transfer formulation, we propose a method that considers how timbre can be explicitly used as a control signal. For a full review of DDSP for audio synthesis see Hayes et al. [23].

3. TIMBRE REMAPPING APPROACH

Here we outline the design of a timbre remapping approach, building on the concept of timbre analogies within the framework of DDSP. The goal is to transform the timbre of a target synthesized sounds by mapping specific dimensions of a performance from an input control signal, which we assume includes variations in timbre that can be measured using acoustic features. We also assume that the input and synthesized sounds occupy unique regions within a multidimensional timbre space. Based on research on timbre analogies introduced in section 2.1, we propose timbre remapping via translation within a computational timbre space. The basic idea is to measure how timbre changes

[1] `https://jordieshier.com/projects/nime2024/`

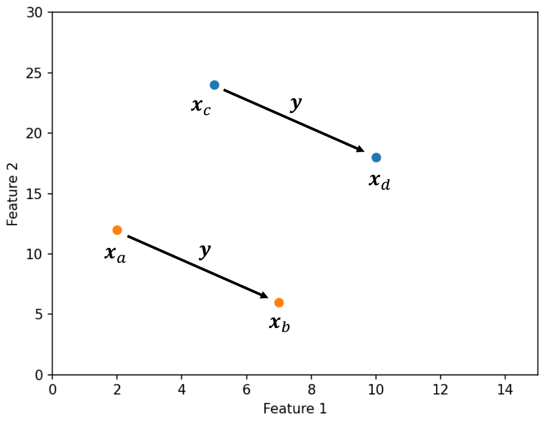

Figure 1: Two-dimensional representation of a timbre analogy. The sound pair $(\mathbf{x}_a, \mathbf{x}_b)$ form a timbre sequence and differ by $\mathbf{y}$. Given a new sound $\mathbf{x}_c$ (e.g., a synthesizer sound), an analogous timbre sequence can be created by applying the difference described by $\mathbf{y}$ to form the new pair $(\mathbf{x}_c, \mathbf{x}_d)$.

across successive musical events in an input audio control signal and translate those changes into synthesizer parameter modulations. In the next section we outline a method for performing this using DDSP.

3.1 Learning Feature Differences

We start with a timbral sequence comprising two sounds $\mathbf{x}_a$ and $\mathbf{x}_b$ from an input control source. A multidimensional timbre space is defined by an arbitrary audio feature extraction algorithm $f(\cdot)$ that returns a multidimensional vector of features. The timbral sequence can be described by the vector resulting from taking the difference between audio feature vector of $\mathbf{x}_a$ and $\mathbf{x}_b$:

$$\mathbf{y} = f(\mathbf{x}_b) - f(\mathbf{x}_a) \tag{1}$$

Now, given a synthesizer $g(\cdot)$ and a preset $\theta_{pre} \in \mathbb{R}^P$, where P is the number of synthesizer parameters, we can synthesize an audio signal $\mathbf{x}_c = g(\theta_{pre})$. We can also generate a modulated version of that preset by applying a parameter modulation $\theta_{mod} \in \mathbb{R}^P$ which results in a new synthesized signal $\mathbf{x}_d = g(\theta_{pre} + \theta_{mod})$. This forms a new timbre sequence:

$$\begin{aligned} \hat{\mathbf{y}} &= f(\mathbf{x}_d) - f(\mathbf{x}_c) && (2) \\ &= f\left(g(\theta_{pre} + \theta_{mod})\right) - f\left(g(\theta_{pre})\right) && (3) \end{aligned}$$

Our goal is to learn θ_{mod} such that $\hat{y} = y$. Figure 1 shows a visual overview of this process. In this formulation, θ_{pre} and $\mathbf{x}_a$ are fixed and can be selected based on the musical application. To situate this formulation within a gradient descent-based machine learning paradigm, we introduce a loss function to optimize θ_{mod} to match feature differences.

3.1.1 Feature Difference Loss

DDSP synthesizer training involves optimizing the parameters of synthesizer (and optionally a neural network that predicts parameters) to minimize a loss function. Typically, loss is computed between predicted audio and ground truth audio using an auditory loss function such as the multi-scale spectral loss [60]. This formulation minimizes the absolute error between spectrograms with the objective of replicating ground truth audio. However, we are interested in optimizing synthesizer parameters to match a difference in audio features instead of matching the absolute values of features. To this end, we define a *feature difference loss*:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\|_1 \tag{4}$$

where $\hat{\mathbf{y}}$ is the feature difference vector from equation 2, $\mathbf{y}$ is the reference difference vector from equation 1, and $\|\cdot\|_1$ is the L_1 norm.

3.1.2 Optimization Target

If both $g(\cdot)$ and $f(\cdot)$ are differentiable functions then θ_{mod} can be directly optimized using gradient descent. Putting this all together, we arrive at a final optimization target:

$$\hat{\theta}_{mod} = \text{argmin}_{\theta_{mod} \in \mathbb{R}^P} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) \tag{5}$$

This formulation makes no assumptions regarding the nature of $g(\cdot)$ and $f(\cdot)$ and only requires differentiability, which is relatively straightforward to achieve given the maturity of modern auto-differentiation software[2]. In the next section we provide a concrete example using this formulation.

4. CASE STUDY: SNARE DRUMS

As a case study we investigate the application of timbre analogies and DDSP for timbre remapping within the musical context of snare drum performances. The design of this study is motivated by prior work by Danielsen et al. [9], which investigated the role of dynamic and timbral variation on snare drums within drum kit performances. They found that drummers systematically varied intensity and timbre of snare drum hits within grooves to signal rhythmic and timing intentions. In this case study, we explore how variations in a snare drum performance can be mapped onto parameters of a snare drum synthesizer. Our goal is to perform an initial evaluation of the proposed approach for modeling timbre variation and to demonstrate a practical example supporting real-time music interaction.

4.1 Differentiable Drum Synthesizer

The design of our differentiable drum synthesizer $g(\cdot)$ is inspired by the popular Roland TR-808 snare drum. Although it is relatively simple in design, the Roland TR-808 has found widespread use within popular music [20]. We implemented a modified version of the TR-808 snare model based on schematics provided by Gordon Reid [45]. A block diagram of the synthesis model is shown in figure 2.

This synthesizer consists of two parallel paths, the first comprises a pair of sinusoidal oscillators to generate the main resonant frequencies and the second is a noise generator responsible for the sound of the "snares". Each sinusoidal oscillator has a frequency parameter and a parameter to control amount of frequency modulation applied by a control envelope. All sound sources have an independent amplitude control with gain and an envelope. Frequency and amplitude envelopes are exponentially decaying envelopes with control over the decay time. The noise source is filtered with a high-pass biquad filter and uses the differentiable implementation introduced by Yu and Fazekas [64]. We added a hyperbolic tangent waveshaper to the output as we found the ability to add harmonics beneficial for shaping

[2] For example, PyTorch `https://pytorch.org/` and TensorFlow `https://www.tensorflow.org/`

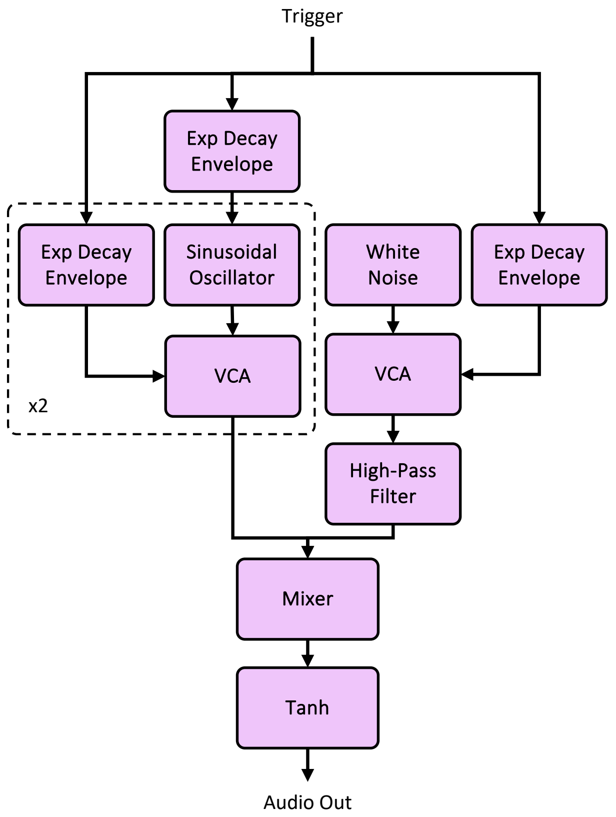

Figure 2: Drum synthesizer block diagram, modeled after the Roland TR-808 snare drum.

transients in addition to being aesthetically pleasing. There are fourteen synthesis parameters in total.

4.2 Audio Features

Next, we define a feature extraction algorithm $f(\cdot)$ for measuring dynamic and timbre variations. The exact definition of $f(\cdot)$ is not fixed and can be designed based on musical context. Here, we select features based on Danielsen et al. [9], who used sound pressure level, temporal centroid, and spectral centroid. Their selection was motivated by the MPEG-7 standard [26] and previous research on percussive timbre analysis [32, 43]. We use this set of features and also include spectral flatness [11] based on it's inclusion in SP-Tools[3]. Generally speaking, these features provide insight into amplitude (sound pressure level), envelope shape (temporal centroid), "brightness" (spectral centroid), and "noisiness" (spectral flatness). See Caetano et al. [3] for more in-depth information on timbre-related audio descriptors.

All audio features, except for temporal centroid which uses a 125ms window size, are computed using frame-based processing with a window size of 46.4ms with 75% overlap. Following recent suggestions [49], spectral features are windowed using a flat-top window prior to the FFT and the resulting magnitude spectrum is compressed using the following function: $p(X) = \log(1{+}X)$, where X is a magnitude spectrum. These modifications were shown to produce a smoother gradient for sinusoidal frequency estimation. Audio feature time-series are summarized using the mean.

Building on Danielsen et al. [9], all features except for temporal centroid are extracted from two temporal segments within the same audio sample. A short segment containing N_t windows are selected from the onset to capture transient phase information and a longer segment containing N_s windows are selected after the N_t windows to capture the sustain/decay phase information. The result is a seven dimensional audio feature space consisting of both timbral and dynamic features.

4.2.1 *Psychophysical Scaling*

Just as the equal-tempered scale enables transpositions of melodies between different keys, we seek a scaling that allows us to transpose sequences within the timbre/dynamic space defined by our audio feature extraction algorithm. Dynamic features, computed as the root mean square (RMS), are converted to loudness, k-weighted, relative to full-scale (LKFS) [58] by applying the following scaling function:

$$s_{\mathrm{LKFS}}(x_{\mathrm{RMS}}) = -0.691 + 10\log_{10}(h(x_{RMS})) \qquad (6)$$

where $h(\cdot)$ is a K-weighting pre-emphasis filter.

Kazazis et al. [30] derived the following scaling function for spectral centroid:

$$s_{\mathrm{SC}}(x_{\mathrm{SC}}) = -34.61x_{\mathrm{SC}}^{-0.1621} + 21.2985 \qquad (7)$$

where x_{SC} is spectral centroid measured in hertz.

Temporal centroid is related to duration within our synthesis framework. Schlauch et al. [48] found that perception of duration in damped sounds is dependent on frequency and timbre; however, derived power functions were all close to $d^{0.5}$ where d is duration. Accounting for the non-linear relationship between temporal centroid and duration of the exponential decay envelopes in our synthesizer, which was empirically determined by sampling envelopes, we derived the following psychophysical scaling for temporal centroid:

$$s_{\mathrm{TC}}(x) = 0.03x_{\mathrm{TC}}^{1.864} \qquad (8)$$

To our knowledge, there is no literature investigating the perceptual scaling of spectral flatness; however, taking guidance from the Librosa documentation [39], spectral flatness is converted to a decibel scale: $s_{\mathrm{SF}}(x) = 20\log_{10}(x_{\mathrm{SF}})$.

Now equipped with a differentiable synthesizer $g(\cdot)$ and an audio feature extractor $f(\cdot)$ with perceptually informed scalings, we are ready to define a machine learning task for snare drum timbre remapping.

4.3 Real-Time Timbre Remapping

We now consider how these concepts can be applied to the musical task of real-time control of a synthesizer. The proposed system is inspired by the SP-Tools library, developed by percussionist and researcher Rodrigo Constanzo using FluComa [56], and recent work on percussive DMI control [33]. These works support real-time machine learning tasks using audio features extracted at detected onsets. We explore here learning mappings between onset features and synthesizer parameter modulations for real-time timbral control. A diagram overviewing this approach is provided in Figure 3.

We frame timbre remapping as a regression problem and use a data-driven approach to model the relationship between onset features and parameter modulations. The goal is to estimate synthesizer parameter modulations $\hat{\theta}_{mod}$ from short-term audio features $f_o(\mathbf{x})$, where f_o is an onset feature extractor and $\mathbf{x}$ audio from a single acoustic snare drum

[3] https://github.com/rconstanzo/SP-tools

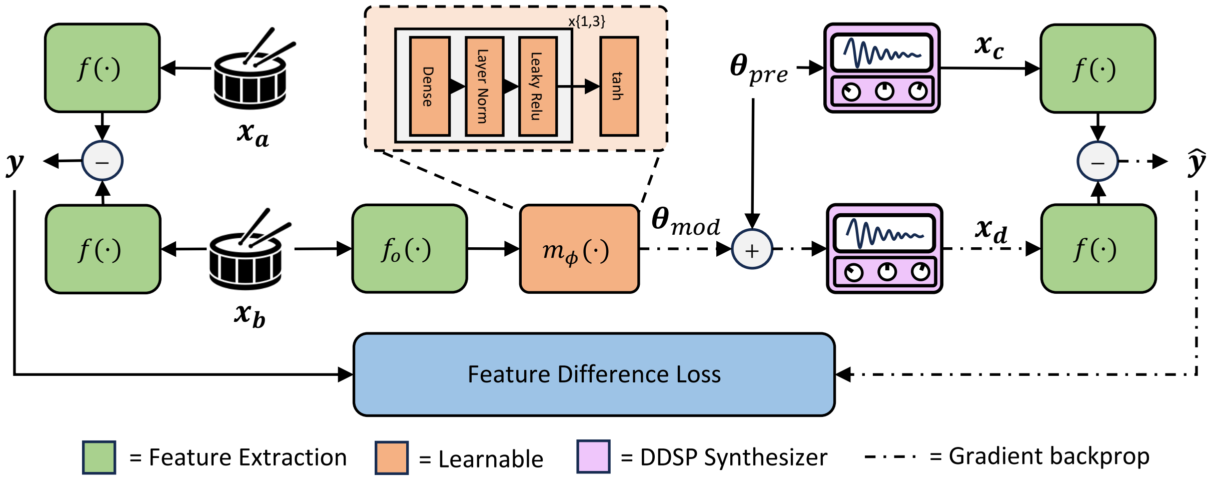

Figure 3: Real-time timbre remapping experiment overview. We learn a mapping network $m_\phi(\cdot)$ to predict synthesizer parameter modulations θ_{mod} to create timbre analogies. Feature differences $\mathbf{y}$ are measured between two input sounds $(\mathbf{x}_a, \mathbf{x}_b)$ and $m_\phi(\cdot)$ learns to modulate a synthesizer preset θ_{pre} to create a synthesized sound pair $(\mathbf{x}_c, \mathbf{x}_d)$ with a feature difference $\hat{\mathbf{y}}$. The feature difference loss measures the error between $\mathbf{y}$ and $\hat{\mathbf{y}}$. Real-time remapping is enabled by onset features $f_0(\cdot)$, which are measured on a short window of audio at a detected onset and are used as input to the mapping network.

strike. To do so we introduce a mapping function $m_\phi(\cdot)$ that outputs parameter modulations given onset features:

$$\hat{\theta}_{mod} = m_\phi(f_o(\mathbf{x})) \tag{9}$$

where ϕ are the learnable model parameters.

Now our objective is to learn ϕ to estimate $\hat{\theta}_{mod}$ to minimize the feature difference loss instead of directly optimizing $\hat{\theta}_{mod}$. To return to the concept of timbre analogies as a method for mapping, we construct a dataset of timbral sequences from an audio dataset of snare drum hits (e.g., snare drum hits extracted from a performance). This is done by selecting a reference sample $\mathbf{x}_a$ from the dataset and then measuring the difference between that and every other sample in the dataset. The reference sample acts as an anchor point in the dataset from which timbral/dynamic variations extend from and defines the sound for which the synthesizer preset is unmodulated (i.e., $\theta_{mod} = 0$).

Onset features are derived from a subset of features used in SP-Tools and are RMS, spectral centroid, and spectral flatness computed on a buffer of 256 samples after a detected onset. Spectral features are computed on a magnitude spectrum and time domain samples are windowed with a Hann window prior to an FFT. Onset detection is computed using an amplitude-based method derived on an implementation in the FluComa library [56] (AmpFeature[4]) and used in SP-Tools. All onset features are normalized to a $[0, 1]$ range.

5. EXPERIMENTS

Initial experimentation preceded the development of the concepts outlined in this paper and consisted of a manually-tuned mapping strategy where linear relationships between onset features and synthesis parameters were specified by a user on a user interface – no machine learning involved. This straightforward approach enabled everyday objects to be struck, transforming them into different elements of a synthetic drum kit. Furthermore, we presented this version of the project at the Agential Insrtuments Workshop at the 2023 AI and Music Creativity conference, where it was integrated into a project completed by two workshop participants.

This early success encouraged us to further develop this idea and address some of the main limitations: 1) manual-mapping is effective when there are relatively few features and synthesis parameters, but becomes unwieldy as complexity increases; 2) developing non-linear relationships with interdependencies between features is infeasible with manual-mapping; 3) creating mappings that lead to realistic graded timbral changes can be challenging. The ideas presented in this paper represent our attempt to address some of the challenges. Whilst we don't directly compare the manual mapping approach with the data-driven mappings in this paper, in the following subsections we present a series of numerical and musical experiments with the goal of providing insight into the efficacy and musical affordances of our timbre remapping design.

5.1 Numerical Experiments

Numerical experiments included training a set of different models to conduct real-time timbre remapping using an dataset of snare drum performances. We use subsets of the Snare Drum Dataset (SDSS) [6] to train and evaluate the real-time mapping model. In total there are 48 unique performances and each recording contains between 50 to 120 (median 85) different hits. We select audio from single microphone (an AKG-414 positioned on the top of the drum). A full dataset for training a single model is one performance.

To generate timbral/dynamic analogies for each dataset, onset features and full audio features were computed for each individual drum and a reference drum sound $\mathbf{x}_a$ was selected using the median value of transient LKFS. We found this feature correlated with strike velocity in a preliminary study and provides a reasonable centre point in the audio feature space to generate timbral/dynamic analogies from. Testing and validation splits were created from each dataset using approximately ten percent of the samples, selected to roughly cover the dynamic range of the dataset. Five different synthesizer presets were manually programmed to serve

[4] https://learn.flucoma.org/reference/ampfeature/

as the starting point for timbre analogies. Combining the 48 performances with five different presets meant that 240 individual models were trained for each variation during experimentation.

5.1.1 Models and Training

Three different model variations were included for experimentation. Two models based on the multi-layer perceptron (MLP) used by Engel et al. [13] were included. One with a single layer containing 32 hidden units (590 parameters) and a larger model with three layers of 64 hidden units (9.5k parameters). A linear model was also included for experimentation, which reflects the mapping capability of the aforementioned manual-mapping method.

Two window sizes for onset features were also included, one with short-term features of 256 samples, and one with a larger window of 2048 samples. The larger window ($\approx$ 43ms at 48kHz) would have too much latency for real-time percussion performance, which has an upper perceptual threshold of about 10ms [27]. We include this larger window to investigate the benefit of providing more temporal context during training. Parameters were also directly optimized to match differences (i.e., no modeling) to evaluate the effectiveness of the feature difference loss and provide an upper bound on performance.

Each model was trained for 250 epochs using an Adam optimizer and the learning rate was halved if validation loss did not improve for 20 epochs. To prevent over utilization of oscillator frequency parameters, modulations for those two controls were damped by a factor of 1e-3. Training a model takes under 2 minutes on a NVIDIA GeForce RTX 2080 Ti GPU and about 13 minutes on the CPU of a MacBook Pro M1. A full listing of hyperparmaters and model details are listed on the supplementary website.

5.1.2 Results

Metrics reporting how accurately each model variant was able to match audio feature differences are shown in table 1. All results were computed using samples from testing datasets and results are summarized with mean and standard deviation across all SDSS performance and preset pairs. The preset column shows the feature difference error computed against the preset before any optimization to provide a performance baseline – this would represent a one-shot triggering scenario. The direct optimization results performed the best across all features, which is as expected since no model is being trained to estimated parameters. While the non-linear MLP models tend to provide better feature matching capabilities across most features, they are not significantly better than the best linear model. These results show that we were able to relatively accurately learn to map synthesizer parameter changes using this approach compared to the baseline direct optimization and that these relationships can be modeled within a single snare drum performance with relatively simple models.

5.2 Musical Experiments

Deruty et al. [10] emphasize working alongside musicians in the development AI-tools and highlight the importance of creating usable prototypes that function within a musicians typical workflow. To facilitate experimentation within the intended musical context, an audio plug-in was developed to perform real-time timbre remapping. The only difference between the plug-in and training is that the audio feature extraction algorithms and synthesizer was re-written in C++ (as opposed to Python) and a rolling feature normalizer was added to ensure that input features were in the correct $[0, 1]$ range. The plug-in, source code, and recordings from the musical experiment are available on the supplementary website.

We conducted an informal session with professional drummer Carson Gant to record musical examples to accompany this paper and help situate this work within the practice of a groove-based drummer. The goal of this session was provide initial feedback of our approach within a musical context and is not intended to replace a formal user study, which the authors plan to conduct at a later date. Carson provided short recordings of performances on two different snare drums with and without dampening, which were used to train models ahead of our session. An important distinction between the recordings received from Carson and the SDSS dataset is that Carson played a much wider range of gestures including buzz roles and rim clicks.

After playing for a period of time, Carson remarked "there is some subtleness to it where you're not getting one-shotted[5], there are subtle changes to it ... it's nice to hear, it's reacting ... it's just figuring out how to play it and what causes it to trigger [or not]". This statement points to both a success of the timbre remapping in creating subtle variations and a limitation of relying on onset detection. Carson's statement reflects findings by Jack et al. [28], who observed percussionists reducing their gestural language when confronted with the bottleneck of discrete onset detection in a percussive DMI. It is worth noting that the setup in our session, which used a dynamic microphone as input, represents a challenging scenario for onset detection and could likely be significantly improved with the introduction of a drum trigger (p.c., Rodrigo Constanzo).

Carson played several different models, remarking that some "felt more reactive" or "were triggering better." This was interesting as the onset detection and triggering is separate from the mapping model. This points to a perceptual connection between the variation in sound produced and a sensation of reactiveness. A feeling of less reactivity was particularly salient in presets that contained higher levels of filtered noise. Small adjustments in high-pass filter parameters (cut-off and q) created perceptually significant changes and minor variations in the input features tended to feel over-emphasized. Carson mentioned that this resulted in a sensation of randomness, although also commented that playing on the edge of the drum resulted in one sound and in the middle another, suggesting that macro control worked well, but granular control over noise was marginal. This variability could also be attributed in part to the feature normalization in the audio plug-in, which would update over time and could cause outputs to change over time.

6. DISCUSSION

6.1 Differentiable Timbre Space

The timbre space has proved an enduring concept for control of sound synthesizers. The marriage of DDSP with timbre space in this work offers a novel perspective, enabling the direct learning of synthesis parameters from audio examples, circumventing the need for supervision on parameters [59] or generative training on large datasets [42]. Expressing perceptual knowledge directly within our DDSP training algorithm allowed us to explicitly specify the musical concepts that we deemed important for the task at hand. In this

[5] Referring to the effect of re-triggering a recorded sample repeatedly, sometimes called a "machine-gun effect" [15]

Table 1: Feature Difference Errors

Feature	Preset	Direct	Linear		MLP		MLP LRG	
			256	2048	256	2048	256	2048
$LKFS_T$	19.6 ± 4.7	0.473 ± 1.1	1.196 ± 0.8	$\mathbf{0.479 \pm 0.6}$	0.962 ± 1.0	0.743 ± 1.0	1.016 ± 1.0	0.808 ± 1.1
$LKFS_S$	60.8 ± 21	1.244 ± 1.3	2.662 ± 1.7	2.751 ± 1.8	2.131 ± 1.6	2.104 ± 1.6	2.083 ± 1.6	$\mathbf{2.081 \pm 1.5}$
SC_T	12.7 ± 1.4	0.120 ± 0.1	0.166 ± 0.1	0.163 ± 0.1	0.129 ± 0.1	$\mathbf{0.125 \pm 0.1}$	0.130 ± 0.1	0.133 ± 0.1
SC_S	12.8 ± 1.4	0.221 ± 0.1	0.228 ± 0.1	0.231 ± 0.1	$\mathbf{0.223 \pm 0.1}$	0.225 ± 0.1	0.230 ± 0.1	0.233 ± 0.1
SF_T	48.4 ± 45	1.392 ± 2.0	3.211 ± 2.1	2.091 ± 2.1	2.307 ± 1.8	$\mathbf{1.747 \pm 1.8}$	2.354 ± 1.9	1.953 ± 2.0
SF_S	34.2 ± 46	2.075 ± 2.7	4.284 ± 2.8	4.124 ± 2.5	$\mathbf{3.645 \pm 2.6}$	3.662 ± 2.6	3.799 ± 2.5	3.881 ± 2.6
TC	22.7 ± 22	1.907 ± 5.0	2.771 ± 5.0	2.593 ± 4.9	2.264 ± 4.9	2.222 ± 4.9	2.215 ± 4.9	$\mathbf{2.164 \pm 4.9}$

LKFS: Loudness, K-Weighted, relative to full-scale; SC: Spectral Centroid; SF: Spectral Flatness; TC: Temporal Centroid
T: Transient phase; S: Sustain phase.
Lower values are better for all values and results in bold highlight the best modeling approach for each feature.

case, we highlighted the importance of timbral differences between events in a musical phrase by using a feature difference loss. This enabled the efficient training of lightweight models capable of performing real-time timbre remapping. However, representing a sound as a point in a multidimensional and numerical timbre space also involves a reduction – a timbral bottleneck – similar to the gestural bottleneck introduced through onset detection [28]. Bottlenecks in our design had implications that were reflected in our musical experiment and reveal avenues for future investigation.

6.2 Limitations

The proposed feature difference loss was designed under the assumption that timbre variations can be represented as vectors in a computational timbre space and that vectors with the same direction (but different origins) will be perceived similarly. Previous research on timbre analogies and scaling of timbre-related audio descriptors supports this assumption, and our musical experiment offers an initial practice-based evaluation of its perceptual relevance. Further investigation into the perceptual relevance of the proposed method in psychoacoustical and musical contexts will be both enlightening and important for future development in this direction. Additionally, the choice of reference in the feature difference loss has implications on the end result and future work can explore different formulations such as dynamic references that are updated based on shorter musical phrases.

While DDSP offers numerous benefits, it also introduces some unique challenges. The difficulty of optimizing frequency with respect to audio loss functions is well-known [57, 34]. Our training scenario avoided the need to directly learn frequency; however, we observed uninformative gradients with respect to frequency parameters. The impact of uninformative gradients meant model weight initialization had a large impact on training and necessitated the use of frequency parameter dampening to mitigate bad solutions. Despite these challenges, and in light of the clear aforementioned benefits and recent insights in DDSP optimization [22, 49], we feel that continued research in this direction is merited. Future work comparing non-differentiable approaches [59] would also be worthwhile.

6.3 Opportunities

Beyond the real-time percussive timbre remapping application presented in this work, there are numerous applications of timbre remapping with DDSP. We highlight a few here. A simple extension of our current work is explore the benefits of exposing direct parametric control over timbral features. In our case study, onset features are used as input to the parameter mapping neural network. However, there is no reason why values from an external controller couldn't be mapped to these. One application that we have already started to explore is mapping MIDI and MPE values from a controller like the Ableton Push[6]. This could enable more nuanced timbral control over synthesis parameters in finger-drumming and other controller-based performance contexts. Furthmore, parametric timbre control could be used in sound design applications or to create stimuli for perceptual studies where independent control over individual features is beneficial and typically relies on additive synthesis [30]. Creation of meaningful variations in sounds is an another active area of research for drum one-shots [15] and sound effects for video games [51]. Timbre variations could be learned in a data-driven manner from sample libraries, for instance, by creating differentiable implementations of procedural synthesis methods [40].

7. CONCLUSION

In this paper we have explored how timbre analogies and differentiable audio synthesis can be leveraged together for the task of timbre remapping. Specifically, we sought to map subtle timbral changes from acoustic instruments onto controls for a synthesizer, motivated by musical contexts where timbre is a primary vehicle for expression. By expressing synthesis and feature extraction algorithms differentiably, and through the use of our proposed feature difference loss function, we showed how we could learn to adjust synthesis parameters to match timbral and loudness feature sequences. Importantly, we matched differences, as opposed to absolute values audio features, which emphasized the importance of trajectories in timbre space and enabled remapping. This was shown in a concrete example that explored real-time remapping from acoustic snare drum performances to a differentiable drum synthesizer inspired by the Roland TR-808.

8. ACKNOWLEDGMENTS

This work is supported by the UKRI through the Centre for Doctoral Training in Artificial Intelligence and Music (EP/S022694/1) and a UKRI Frontier Research grant (EP/X023478/1). We would like to thank Carson Gant for his help and feedback during the performance session. We would also like to thank Rodrigo Constanzo for his inspiring work on the SP-Tools library and for the insightful conversations. Thank you to all the NIME reviewers for their valuable feedback.

[6] https://www.ableton.com/en/push/

9. ETHICAL STANDARDS

All research carried out as a part of this work was conducted solely by the authors. Material related to the musical session with Carson Gant, including quotes and video recordings on the supplemental website, are included with his permission. We acknowledge the potential for machine learning technology such as this to misused, for instance in the creation of fake or misleading content. We've endeavored to use small data and models to mitigate this risk and to enable musicians to utilize this technology themselves.

10. REFERENCES

[1] A. Barahona-Ríos and T. Collins. NoiseBandNet: Controllable time-varying neural synthesis of sound effects using filterbanks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1573–1585, Feb. 2024.

[2] E. Bates. The interface and instrumentality of Eurorack modular synthesis. In *Rethinking Music through Science and Technology Studies*, pages 170–188. Routledge, May 2021.

[3] M. Caetano, C. Saitis, and K. Siedenburg. Audio Content Descriptors of Timbre. In *Timbre: Acoustics, Perception, and Cognition*, pages 297–333. Springer International Publishing, May 2019.

[4] M. Carney, C. Li, E. Toh, P. Yu, and J. Engel. Tone Transfer: In-Browser Interactive Neural Audio Synthesis. In *Joint Proceedings of the ACM IUI 2021 Workshops*, Apr. 2021.

[5] F. Caspe, A. McPherson, and M. Sandler. DDX7: Differentiable FM Synthesis of Musical Instrument Sounds. In *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, Dec. 2022.

[6] M. Cheshire. Snare Drum Data Set (SDDS): More snare drums than you can shake a stick at. In *Audio Engineering Society Convention 149*, Oct. 2020.

[7] P. Chordia. Segmentation and Recognition of Tabla Strokes. In *Proc. of the 6th Int. Society for Music Information Retrieval Conf.*, Sept. 2005.

[8] S. Dai, Z. Zhang, and G. G. Xia. Music Style Transfer: A Position Paper. In *Proceeding of International Workshop on Musical Metacreation (MUME)*, June 2018.

[9] A. Danielsen, C. H. Waadeland, H. G. Sundt, and M. A. G. Witek. Effects of instructed timing and tempo on snare drum sound in drum kit performance. *The Journal of the Acoustical Society of America*, 138(4):2301–2316, Oct. 2015.

[10] E. Deruty, M. Grachten, S. Lattner, J. Nistal, and C. Aouameur. On the Development and Practice of AI Technology for Contemporary Popular Music Production. *Transactions of the International Society for Music Information Retrieval*, 5(1):35–49, Feb. 2022.

[11] S. Dubnov. Generalization of Spectral Flatness Measure for Non-Gaussian Linear Processes. *IEEE Signal Processing Letters*, 11(8):698–701, Aug. 2004.

[12] D. Ehresman and D. L. Wessel. Perception of Timbral Analogies. Technical Report 13, IRCAM, 1978.

[13] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts. DDSP: Differentiable digital signal processing. In *International Conference on Learning Representations*, Apr. 2020.

[14] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos. Flow Synthesizer: Universal Audio Synthesizer Control with Normalizing Flows. *Applied Sciences*, 10(1):302, Dec. 2020.

[15] J. Fagerstrom, S. J. Schlecht, and V. Valimaki. One-to-Many Conversion for Percussive Samples. In *24th International Conference on Digital Audio Effects (DAFx)*, pages 129–135, Sept. 2021.

[16] S. Fasciani. Interactive Computation of Timbre Spaces for Sound Synthesis Control. In *Proceedings of Si15, the 2nd International Symposium on Sound and Interactivity*, pages 69–78, Oct. 2020.

[17] J. Gregorio and Y. E. Kim. Augmenting Parametric Synthesis with Learned Timbral Controllers. In *New Interfaces for Musical Expression*, June 2019.

[18] J. M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, May 1977.

[19] J. M. Grey and J. W. Gordon. Perceptual effects of spectral modifications on musical timbres. *The Journal of the Acoustical Society of America*, 63(5):1493–1500, May 1978.

[20] Z. Hasnain. How the Roland TR-808 revolutionized music. *The Verge*, Apr. 2017.

[21] B. Hayes, C. Saitis, and G. Fazekas. Neural Waveshaping Synthesis. In *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Nov. 2021.

[22] B. Hayes, C. Saitis, and G. Fazekas. Sinusoidal Frequency Estimation by Gradient Descent. In *ICASSP 2023 - 2023 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, June 2023.

[23] B. Hayes, J. Shier, G. Fazekas, A. McPherson, and C. Saitis. A review of differentiable digital signal processing for music and speech synthesis. *Frontiers in Signal Processing*, 3, Jan. 2024.

[24] M. D. Hoffman and P. R. Cook. Feature-Based Synthesis: Mapping Acoustic and Perceptual Features onto Synthesis Parameters. In *Proceedings of the 2006 International Computer Music Conference*, Nov. 2006.

[25] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse. TimbreTron: A WaveNet(CycleGAN(CQT(audio))) pipeline for musical timbre transfer. In *International Conference on Learning Representations*, May 2019.

[26] ISO/IEC. ISO/IEC FDIS 15938-4:2002, MPEG-7: Information Technology – Multimedia Content Description Interface - Part 4: Audio, 2002.

[27] R. H. Jack, A. Mehrabi, T. Stockman, and A. McPherson. Action-sound Latency and the Perceived Quality of Digital Musical Instruments. *Music Perception*, 36(1):109–128, Sept. 2018.

[28] R. H. Jack, T. Stockman, and A. McPherson. Rich gesture, reduced control: The influence of constrained mappings on performance technique. In *Proceedings of the 4th International Conference on Movement Computing*, pages 1–8, June 2017.

[29] T. Jourdan and B. Caramiaux. Machine Learning for Musical Expression: A Systematic Literature Review. In *New Interfaces for Musical Expression*, May 2023.

[30] S. Kazazis, P. Depalle, and S. McAdams. Interval and Ratio Scaling of Spectral Audio Descriptors. *Frontiers in Psychology*, 13, Mar. 2022.

[31] C. L. Krumhansl. Why is musical timbre so hard to understand? *Structure and Perception of Electroacoustic Sound and Music*, 9:43–55, 1989.

[32] S. Lakatos. A common perceptual space for harmonic and percussive timbres. *Perception & Psychophysics*,

62(7):1426–1439, Oct. 2000.

[33] A. Martelloni, A. P. McPherson, and M. Barthet. Real-time Percussive Technique Recognition and Embedding Learning for the Acoustic Guitar. In *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Nov. 2023.

[34] N. Masuda and D. Saito. Improving Semi-Supervised Differentiable Synthesizer Sound Matching for Practical Applications. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:863–875, Jan. 2023.

[35] S. McAdams. The Perceptual Representation of Timbre. In *Timbre: Acoustics, Perception, and Cognition*, pages 23–57. Springer International Publishing, May 2019.

[36] S. McAdams. Timbre as a Structuring Force in Music. In *Timbre: Acoustics, Perception, and Cognition*, pages 211–243. Springer International Publishing, May 2019.

[37] S. McAdams and A. Bregman. Hearing Musical Streams. *Computer Music Journal*, 3(4):26–60, Dec. 1979.

[38] S. McAdams and J.-C. Cunible. Perception of Timbral Analogies. *Philosophical Transactions: Biological Sciences*, 336(1278):383–389, June 1992.

[39] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto. Librosa: Audio and Music Signal Analysis in Python. In *SciPy*, pages 18–24, July 2015.

[40] D. Menexopoulos, P. Pestana, and J. Reiss. The State of the Art in Procedural Audio. *Journal of the Audio Engineering Society*, 71(12):825–847, Dec. 2023.

[41] E. R. Miranda and M. M. Wanderley. *New Digital Musical Instruments: Control and Interactions Beyond the Keyboard*. A-R Editions, Inc., July 2006.

[42] J. Nistal, S. Lattner, and G. Richard. DrumGAN: Synthesis of Drum Sounds With Timbral Feature Conditioning Using Generative Adversarial Networks. In *Proc. of the 21st Int. Society for Music Information Retrieval Conf.*, July 2020.

[43] E. Pampalk, P. Herrera, and M. Goto. Computational models of similarity for drum samples. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):408–423, Feb. 2008.

[44] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, Apr. 2004.

[45] G. Reid. Practical Snare Drum Synthesis. *Sound on Sound*, Apr. 2002.

[46] S. Reynolds. *Energy Flash: A Journey through Rave Music and Dance Culture*. Faber & Faber, June 2013.

[47] F. Roche, T. Hueber, M. Garnier, S. Limier, and L. Girin. Make That Sound More Metallic: Towards a Perceptually Relevant Control of the Timbre of Synthesizer Sounds Using a Variational Autoencoder. *Transactions of the International Society for Music Information Retrieval*, 4(1):52–66, May 2021.

[48] R. S. Schlauch, D. T. Ries, and J. J. DiGiovanni. Duration discrimination and subjective duration for ramped and damped sounds. *The Journal of the Acoustical Society of America*, 109(6):2880–2887, June 2001.

[49] S. Schwär and M. Müller. Multi-Scale Spectral Loss Revisited. *IEEE Signal Processing Letters*, 30:1712–1716, Nov. 2023.

[50] D. Schwarz. *Data-Driven Concatenative Sound Synthesis*. PhD thesis, IRCAM, Jan. 2004.

[51] S. Siddiq. Real-time morphing of impact sounds. In *Audio Engineering Society Convention 139*, Oct. 2015.

[52] Z. Sramek, A. J. Sato, Z. Zhou, S. Hosio, and K. Yatani. SoundTraveller: Exploring abstraction and entanglement in timbre creation interfaces for synthesizers. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, Dis '23, pages 95–114, New York, NY, USA, July 2023.

[53] D. Stowell. *Making Music through Real-Time Voice Timbre Analysis: Machine Learning and Timbral Control*. PhD thesis, Queen Mary University of London, 2010.

[54] D. Stowell and M. D. Plumbley. Timbre remapping through a regression-tree technique. In *Sound and Music Computing (SMC)*, July 2010.

[55] D. Stowell and M. D. Plumbley. Learning Timbre Analogies from Unlabelled Data by Multivariate Tree Regression. *Journal of New Music Research*, 40(4):325–336, Nov. 2011.

[56] P. A. Tremblay, O. Green, G. Roma, J. Bradbury, T. Moore, J. Hart, and A. Harker. Fluid Corpus Manipulation Toolbox. *Zenodo*, July 2022.

[57] J. Turian and M. Henry. I'm Sorry for Your Loss: Spectrally-Based Audio Distances Are Bad at Pitch. In *"I Can't Believe It's Not Better!" NeurIPS Workshop*, Dec. 2020.

[58] I. T. Union. Algorithms to measure audio programme loudness and true-peak audio level. *ITU-R BS.1770*, 2006.

[59] G. L. Vaillant and T. Dutoit. Interpolation of Synthesizer Presets using Timbre-Regularized Auto-Encoders, Dec. 2023.

[60] X. Wang, S. Takaki, and J. Yamagishi. Neural Source-filter-based Waveform Model for Statistical Parametric Speech Synthesis. In *ICASSP 2019 - 2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, May 2019.

[61] D. Wessel, D. Bristow, and Z. Settel. Control of phrasing and articulation in synthesis. In *Proceedings of the 1987 International Computer Music Conference, ICMC*. Michigan Publishing, Aug. 1987.

[62] D. L. Wessel. Timbre Space as a Musical Control Structure. *Computer Music Journal*, 3(2):45, June 1979.

[63] Y. Yang, Z. Jin, C. Barnes, and A. Finkelstein. White Box Search Over Audio Synthesizer Parameters. In *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Nov. 2023.

[64] C.-Y. Yu and G. Fazekas. Singing Voice Synthesis Using Differentiable LPC and Glottal-Flow-Inspired Wavetables. In *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Nov. 2023.