THE DESIGN OF A SMART CITY SONIFICATION SYSTEM USING A CONCEPTUAL BLENDING AND MUSICAL FRAMEWORK, WEB AUDIO AND DEEP LEARNING TECHNIQUES

Stephen Roddy
Department of Electronic & Electrical Engineering, Trinity College Dublin, Dublin, Ireland
roddyst@tcd.ie

Brian Bridges
School of Arts and Humanities, Ulster University, Magee Campus, Derry/Londonderry, United Kingdom
bd.bridges@ulster.ac.uk

ABSTRACT
This paper describes an auditory display system for smart city data for Dublin City, Ireland. It introduces and describes the different layers of the system and outlines how they operate individually and interact with one another. The system uses a deep learning model called a variational autoencoder to generate musical content to represent data points. Further data-to-sound mappings are introduced via parameter mapping sonification techniques during sound synthesis and post-processing. Conceptual blending and music theory provide frameworks, which govern the design of the system. The paper ends with a discussion of the design process that contextualizes the contribution, highlighting the interdisciplinary nature of the project, which spans data analytics, music composition and human-computer interaction.

1. INTRODUCTION
A Smart City is any contemporary urban space that uses Internet of Things (IoT) technologies to collect data that can then be used to manage, govern and define civil resources and policies within that area [1]. In essence, it is a city that is run with the aid of IoT technologies which allow everyday objects, and some more specialized objects, to connect to the internet in ways that make them readable, recognizable, locatable, addressable, and controllable. This allows a city to be more effectively and democratically managed and should in turn have a positive impact on the quality of life for urban citizens [2].

As a Smart City’s infrastructure becomes increasingly complex, the data generated becomes more difficult to present in a meaningful manner and progressively more difficult for citizens to understand and extract meaning from [3]. Sonification, the mapping of data to sonic parameters to communicate information about the original data source, is emerging as a critical tool for understanding and communicating large complex data flows in the context of increasingly networked and data-driven societies [4,5] and researchers have recently begun to uncover strategies for representing complex Smart City [6] and IoT data [7] with sound. This has resulted in increased interest in sonification as a tool for observing network activity patterns, extending Smart City dashboards, and monitoring network security, as evidenced by a number of notable contemporary projects (see [8-10]. The project described in this paper provides a framework and system for sonifying IoT data related to Smart Cities (SC) that can be integrated into broader Smart City Dashboard projects.
blends between the concepts of a bird and a horse and the concepts of a man and a horse respectively [14]. Likewise, the conceptual blend ‘Cyborg’ is represented in figure 1 above.

Kendall [15] relates conceptual blending theory to sound and more specifically electroacoustic music. He argues that the novel emotional and phenomenal qualities found in music are blends of everyday emotional and phenomenal qualities familiar to listeners and music-makers. Goguen and Harrell [16] have argued for the generation of multimedia content, and analysis of style, on the basis of conceptual blending principles. Eppe et al. [17] put forward a novel computational framework for conceptual blending demonstrating how the system can be used to invent novel musical and mathematical concepts. Conceptual integration is employed here as a guiding principle in the design of the generative engine described shortly.

2.1. Variational Autoencoders

![Variational Autoencoder Diagram](image)

Figure 2: Variational Autoencoder

The system described here uses a machine learning model called a variational autoencoder (VAE) to mimic or simulate the process of conceptual blending. An autoencoder is a machine learning model that uses a neural network to encode a compressed representation of features from an original high dimensional data space in a lower-dimensional ‘latent space’. A decoder network can then be used to approximate the original input data from the compression in the latent space [18]. VAEs are generative models that also encode a probability distribution of the original data (mean and standard deviation). They learn regularized latent spaces of continuous feature vectors, where nearby points have similar yet slightly differing properties, resulting in smooth incremental changes across points in a latent space [18]. A VAE is a generative model, meaning that it can generate new content that was not present in the original data on which it was trained. Decoding a sample from a given point in the latent space essentially produces a blend of the features adjacent to that point. This essentially allows us to generate blends of features in the original dataset. The system described in this paper uses Google Magenta’s MusicVAE [18] to generate ‘blends’ of musical materials.

3. SYSTEM SPECIFICATION

The system is comprised of four component layers: Data Acquisition & Processing, the Generative Engine, Sound Synthesis Engine, and Post-processing. The output consists of three distinct sonifications presented in sequence. The first sonification represents weather data the second represents traffic data and the third represents the number of available bikes at city bike stands. When attended to in sequence, they are intended to give an overall sense of the state of the city in terms of these three categories.

3.1. Data Acquisition & Processing

The system is written in JavaScript ECMAScript 11 so that it might be easily integrated into web-based SC data dashboard projects. Data can be acquired via the Fetch API. The implementation described here gathers data from APIs made available through the Dublinlinked Open Data Store (https://data.smartdublin.ie/), managed by Smart Dublin, and the OpenWeather Maps API (https://openweathermap.org/api). The system can use live metrics for estimated travel time on key routes around the city, multi-story car parking space availability, the availability of bikes in the city’s bike-sharing scheme, noise monitoring data at 14 locations across the city, and weather data. This data is provided by Transport Infrastructure Ireland (TI), Dublin City Council (x2), Sonitus Systems, and OpenWeather Maps respectively.

3.2. The Generative Engine

![High Rainfall Motif](image)

Figure 4: High Rainfall Motif.

As noted previously, the generative engine uses a MusicVAE to generate musical material. The model is pre-trained on the
MAESTRO dataset. This was sufficient for generating the melodic content required. It was trained with an Adam optimizer with a learning rate annealed from 10–3 to 10–5 with an exponential decay rate of 0.9999 and a batch size of 512. It was run for 50k gradient updates respectively with a cross-entropy loss function.

Data can then be mapped to make selections from the latent space. When the data reaches a certain level, the corresponding musical concept from the latent space is played. This musical concept will represent a blend of the two input concepts as determined by the data. The degree to which the blended musical concept sounds like either the upper or the lower limit concept is indicative of the value of the current data point. The system produces 20 iterations of blended motifs by default. Therefore, changes in the data which represent 5% of the overall data result in the assignment of a new blended motif to the sound object. The motifs used to represent rainfall data are represented in figures 4 and 5 while the motifs used to represent travel time data on the key motorways that feed Dublin are represented in figures 6 and 7. The musical motifs in the project were written to sound analogous to the phenomena each one represented, using approaches described in detail elsewhere [7].

### 3.3. Sound Synthesis Layer

Parameter mapping sonification (PMSon) involves the mapping of data to auditory parameters such as pitch, amplitude, duration, or timbre to communicate information about the original data source to a listener [20]. Fitch and Kramer [21] developed a “piggy-backing” approach to PMSon in which multiple streams of data are mapped to a variety of different parameters within a single sonification. They found that this approach to mapping resulted in a more effective auditory display. PMSon techniques, and the aforementioned “piggy backing” approach, in particular, are employed in the sound synthesis layer of the system described here.

### Table 1: Mapping Strategy for Weather Data

<table>
<thead>
<tr>
<th>Data</th>
<th>Layer</th>
<th>Technique/Effect</th>
<th>Parameters &amp; Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Generative</td>
<td>Motif Blending</td>
<td></td>
</tr>
<tr>
<td>Rain</td>
<td>Temp</td>
<td>Synthesis</td>
<td>Harmonicity: 1 - 4</td>
</tr>
<tr>
<td></td>
<td>Post-proc</td>
<td>Chorusing</td>
<td>LFO Delay: 0-8hz</td>
</tr>
<tr>
<td>Wind</td>
<td>Post-proc</td>
<td>AutoFilter</td>
<td>Freq: 0-10hz</td>
</tr>
<tr>
<td></td>
<td>Post-proc</td>
<td>AutoPan</td>
<td>0-5hz</td>
</tr>
</tbody>
</table>

Table 2: Mapping Strategy for Traffic Data

### 3.4. Post-processing Layer

<table>
<thead>
<tr>
<th>Data</th>
<th>Layer</th>
<th>Technique/Effect</th>
<th>Parameters &amp; Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic</td>
<td>Generative</td>
<td>Motif Blending</td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>Post-proc</td>
<td>Distortion</td>
<td>Distortion: 0-75%</td>
</tr>
<tr>
<td>Parking</td>
<td>Post-proc</td>
<td>Reverb</td>
<td>Wet/Dry: 0 – 75%</td>
</tr>
<tr>
<td></td>
<td>Post-proc</td>
<td>Low Pass</td>
<td>Cutoff: 220hz – 22khz</td>
</tr>
</tbody>
</table>

This work is licensed under Creative Commons Attribution – Non-Commercial 4.0 International License. The full terms of the License are available at http://creativecommons.org/licenses/by-nc/4.0/
3.4.1. Traffic Data

At the post-processing level, a distortion unit and a reverb unit are added to the signal chain for the traffic object. Noise data is mapped to both distortion and reverb W/D amount. The reverb in question is an implementation of the popular Freeverb algorithm and the tone.js distortion implementation employs a waveshaping technique to generate rich spectra from more simple harmonic components. The reason both distortion and reverb are used is because the distortion alone isn't noisy enough and reverb helps to increase the sense of noisiness.

Noise Level maps to Distortion & Reverb scale of 0 to 75% for both and Freeverb retains a room size of .7 and a dampening factor of 3000. The reason this scale was chosen was so that the combination of reverb and distortion does not overpower and obscure the underlying motif.

The number of free car park spaces in the city is mapped to control the cutoff frequency of a lowpass filter. The filter runs from 220hz to 22000hz. The polarity of the mapping is reversed so that higher numbers of free parking spaces result in lower cutoff frequencies thus creating more empty space in the frequency spectrum of the sound.

![Figure 3: Mapping Strategy for Bike Data.](image)

3.4.2. Weather Data

The temperature data is mapped to modulate the frequency of the LFO controlling the delay time between a range of 0 to 8hz. Depth is set to 1 or 100% and delay time is 2.5 ms. The polarity of the mapping is reversed so that the lower the temperature, the higher the chorusing.

On the FX Level wind speed is mapped to control the frequency of an auto panning algorithm in the range of .0 to 2, or more specifically hard left to hard right. Depth is set to 1 (100%) and the shape of the LFO is a sine wave.

Wind speed is also mapped to the depth and frequency of an auto-filter. The frequency of the LFO driving the is mapped to the data in the range of 0hz to 4hz and depth of the LFO is mapped in the range of 0 to 100%. The LFO uses a sine wave shape. The base frequency for the filter is 440. This is a low pass filter so everything below 440 is passed through. The filter is applied to everything in a range of 4 octaves above 440 and the filter has a roll-off of -24db and a Q factor of 1.

3.4.3. Bike Availability Data

A filter, amplitude envelope, and two delay units are added to the signal chain for the noise generator. The delay units are configured to create a classic ping-pong delay effect. This is a feedback delay effect that can be applied across two or more channels where the first echo is heard in channel A, the second is heard in channel B, the third in channel A again, and so on until the echoes have faded out. This is implemented in our system with 2 feedback delays routed to the left and right stereo channels using the tone.js PingPongDelay object. Bicycle availability data is mapped to control the ping-pong delay and the amplitude envelope for the noise generator is controlled to simulate the sound generated when cycling a bike. The amplitude envelope has even attack, decay, and sustain periods, of 100ms each followed by a sharp 1ms release. The ping pong delay introduces repeats of the original pattern alternating between hard left and hard right presentations of each successive repeat to create a sound similar to a cycling motion. The data is mapped to control the delay in the range of .15 to 1.5.

3.5. Discussion of Design Process and its Implications

The system described here has emerged from a larger project, which is focused on the sonification of IoT data across a range of contexts. One such context was the Pervasive Nation, an Internet of Things testbed run from the Connect Centre at Trinity College Dublin. The design process involved an initial gathering of requirements through discussions with stakeholders who had originally designed the system, those responsible for running and maintaining the system, and additional stakeholders who were making use of the system. These meetings focused on the kinds of tasks for which they would require auditory displays and the kinds of information such displays should provide for stakeholders with different roles. Initial prototypes, written with Python and Csound, were developed and presented to the stakeholders who provided further feedback to guide the design of data to sound mapping strategies.

Iterative design cycles and rapid prototyping techniques were employed to drive the development of the system. An early result of this process was a design framework for the sonification of IoT network traffic data [7]. Feedback on later iterations of that system suggested that the techniques developed in the context of network traffic data sonification might be further extended through generative music, AI, and machine learning techniques.

At this point, two new systems were devised. The first system was designed for live electronic music improvisation and incorporates IoT data from the Smart Dublin Project with an AI-driven generative music system to augment and develop musical ideas presented to the system by a performer. An early iteration of this system is discussed elsewhere [7]. The second, the ambient monitoring system, was an iteration upon and extension of concepts integrated into the design of the original Pervasive Nation sonifications. This system made use of evolutionary computing and deep learning approaches to generative music composition in the context of an ambient auditory display of cryptocurrency data. It is discussed in depth elsewhere [6].

The current system represents a further iteration of this ambient monitoring system. Feedback on the system after its initial presentation [6] at the Third Conference on the Computer Simulation of Musical Creativity suggested that Smart City data might be of more interest as a source domain than cryptocurrency data and that multiple streams of data should be integrated into the display. The idea of creating a tool that might augment or provide an auditory alternative to purely visual SC dashboards was also introduced, and this became a core concept driving the design of the current implementation. Feedback also suggested the adoption of a standard auditory display approach, as an ambient
information systems [6] approach was not relevant to the data being represented and were not likely to be relevant in an SC context either. Another suggestion was that the system should sound less like a single cohesive piece of music and more like a presentation of information using a mixture of sound and music.

The system, in its current state, accounts for the feedback and input of stakeholders involved at each stage of the design process. A particular attempt has been made here to integrate the tonal language of the Western harmonic system with a broader conception of music akin to Varèse’s definition of music as “Organised Sound” [22]. We rely on MusicVAE to generate variations on harmonic materials originally composed by the author. These outputs have broadly tonal syntaxes. The timbres of these blended motifs however are then determined using Fitch and Kramer’s PMSon-based piggy-backing techniques which results in complex sound objects that, on the textural level, bear relation to Godoy’s [23] Gestural-Sonorous Objects. This is especially true of the bike data sonification, which contains no melodic content and instead acts as what Smalley [24] might describe as a surrogate for the physical, gesture (cycling), that a listener might associate with the generation of the sound. On a practical level, these techniques allow us to sonify an increased number of data streams at any given time, but more importantly, they allow us to better exploit the listeners capacity for sonic and musical meaning-making, or simply put, the listener’s ability to interpret meaning from, and /or assign meaning to, a given sound. Our general approach to designing sonifications that utilize a fuller range of listening skills and sonic meaning-making skills is informed by the embodied sonification listening model elucidated elsewhere [25].

Conceptual blending theory is becoming increasingly important in the design of effective HCI [26, 27] and sonification [11, 12] solutions. Here, we use a machine learning model (MusicVAE) to provide listeners with the kind of blended harmonic spaces that Zbikowski [19] and Brower [28] have shown to be compatible with our faculties for meaning-making in Western musical contexts. With the further mapping of data at the sound synthesis and post-processing levels, we achieve a sonic result more akin to Kendall’s [15] model of meaning-making via conceptual blending in electroacoustic music. The authors hope that the system and design process represented here, leveraging conceptual blending via MusicVAE, as well as consideration of conceptual associations of various auditory affordances in the sound design (e.g. the use of cyclical, modulated noise representing bike availability data, and the use of filtering to create space in the harmonic spectrum of a sound, with more such space representing more available parking spaces) can provide a guide for similar work carried out across the field in the future.

4. ACKNOWLEDGMENT

This publication has been partly supported by an Irish Research Council Government of Ireland Postdoctoral Fellowship Award. This publication has partly emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

5. REFERENCES


