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

Stephen Roddy & Brian Bridges

Music and Media Technologies,
Trinity College Dublin
roddyst@tcd.ie

School of Arts and Humanities,
Ulster University
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.

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 specialised 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]. The system described here is intended to address this issue.

Conceptual Blending

Conceptual Blending introduced by Fauconnier and Turner describes how new structures of meaning can be created in acts of creative and artistic thinking.

A conceptual blend involves the integration of two familiar concepts or input spaces. The resulting blended space contains properties that were not present in either of the original concepts in isolation.

The theory has been applied to describe and explain meaning making in a sonic context (Kendall)

Variational Autoencoders

VAEs are generative models that encode a probability distribution of the original data (mean and standard deviation) alongside a compressed representation of features from an original high dimensional data space into a latent space.

They learn regularised 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].

Here we use a latent space architecture, (Specifically Google Magenta's MusicVAE) to integrate materials together creating results that are similar to a conceptual blend.

System Overview

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.

Data Acquisition & Processing

The system is written in JavaScript ECMAScript 11 so that it might be easily integrated in web-based dashboard projects. Data can be acquired via the Fetch API. The implementation described here gathers data from APIs made available through the Dublinked 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 (TII), Dublin City Council (x2), Sonitus Systems, and OpenWeather Maps respectively.

The Generative Engine

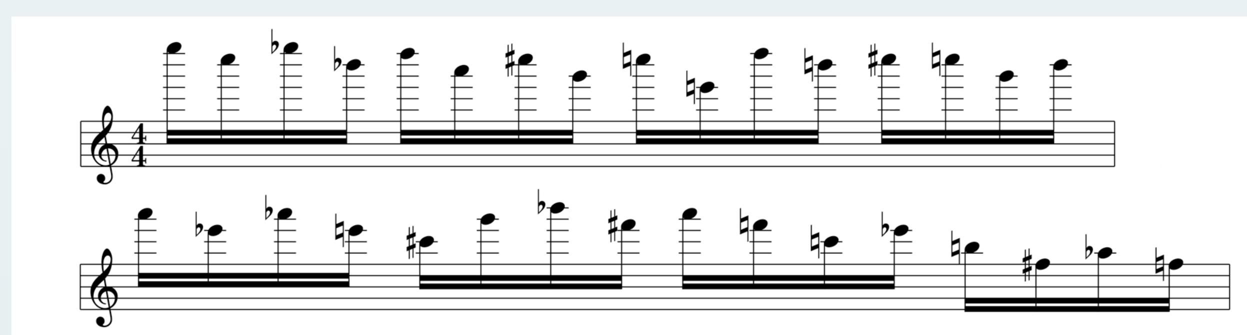

Figure 1. High Rainfall Motiff

The generative engine produces musical materials which are consistent with a harmonic/tonal syntax. We use MusicVAE to generate musical material. The model is pre-trained on the MAESTRO dataset.

The upper limit of a data set is represented by one musical concept or motif and the lower limit of the data is represented by another. Both musical concepts will share some similarities (e.g. similar timbres, both an arpeggio or both a motif, etc.) but their internal structure (e.g. pitches, number of notes, note lengths) will be different to reflect the two opposite limits of the dataset.

The generative engine produces musical materials which are consistent with a harmonic/tonal syntax. The upper limit of a data set is represented by one musical concept or motif and the lower limit of the data is represented by another. Both musical concepts will share some similarities (e.g. similar timbres, both an arpeggio or both a motif, etc.) but their internal structure (e.g. pitches, number of notes, note lengths) will be different to reflect the two opposite limits of the dataset.

Figure 2. Low Rainfall Motiff

Sound Synthesis Layer

The sound synthesis layer employs a parameter mapping sonification approach informed by Fitch and Kramer's" piggy-backing" approach to PMSon in which multiple streams of data are mapped to a variety of different parameters within a single sonification.

Data	Layer	Technique/Effect	Parameters & Range
Rain	Generative	Motif Blending	
Temp	Synthesis	2 Voice Oscil	Harmonicity: 1 - 4
	Post-proc	Chorusing	LFO Delay: 0-8hz
Wind	Post-proc	AutoPan	Freq: 0-10hz
	Post-proc	AutoFilter	0-5hz

Table 1: Mapping Strategy for Weather Data

Musical information is rendered into audio at the sound synthesis layer and further data is also mapped in, using parameter-mapping techniques. The current implementation of the system uses the tone.js library to handle audio in Javascript.

MusicVAE outputs notes values in a range from 0 to 100, a lower pitch resolution than the 128 available MIDI pitches. These outputs are mapped into a useful range allowing us to produce the motifs described in the previous section.

The weather motif is synthesized using two synthesis voices each consisting of an oscillator with an amplitude envelope connected to a filter with its own frequency envelope.

The traffic motif is synthesized using AM (amplitude modulation) synthesis methods where the amplitude of one oscillator is controlled by the output of a second oscillator.

The third component of the synthesis layer is a simple pink noise generator. This noise source will ultimately be used to represent the number of bicycles in use in the city when that data is mapped in at the post-processing layer.

Post-Processing Layer

Further data to sound mappings are carried out at the post-processing layer. For example , 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.

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.

Design Process

The system described here has emerged from a larger project, which is focused on the sonification of IoT data across a range of contexts.

This system described here is designed with integration into SC data dashboards in mind. The original project was concerned with IoT network data sonification.

This system is the result of a series of design iterations over original work that began with the design of sonification solutions for the Pervasive Nation, an Internet of Things test bed run from the Connect Centre at Trinity College Dublin.

The process began with stakeholder interviews to determine project requirements and develop design brief.

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.

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.

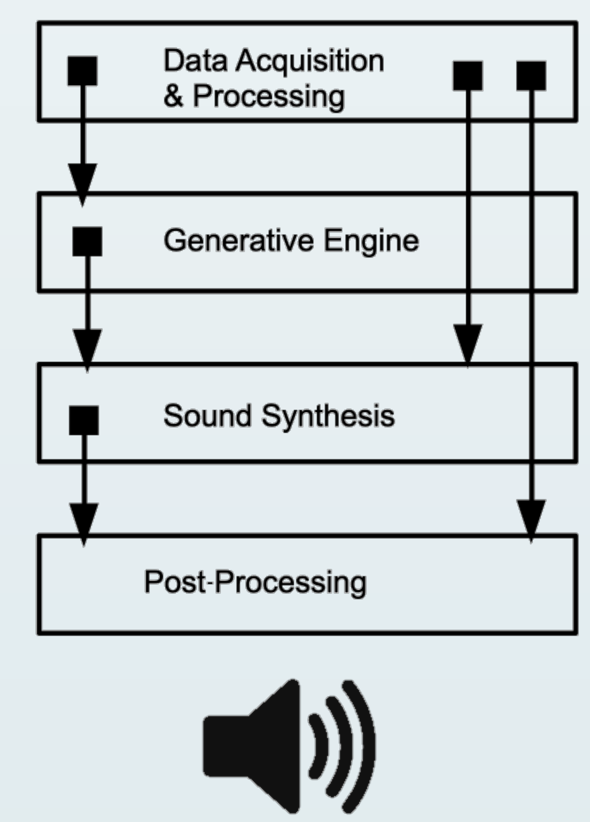

Figure 3. System Overview

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 [see paper].

The current system represents a further iteration of this ambient monitoring system.

Expert feedback experts suggested refocusing on SC data and creating a system that might augment or extend SC data dashboards that tend to present a purely visual experience.