“Genre Bender” - Project Proposal

Arran Lyon and Valentin Vogelmann

Goal:

- Raw audio input
- Selection of “genre”/style transformation
- Raw audio output in selected genre
- Prototype/proof of concept
- Basic user interface for demonstration that allows exploration of genre space.

Prerequisites

- A *spectrogram* is a representation of audio data in time and frequency, essentially converting the 1D sequence of amplitudes of an audio waveform into a 2D “image”, with time on the x-axis and frequency components on the y-axis. This representation (made using the Short-Term Fourier Transform, *STFT*) contains all the information, and so the reverse operation is possible to convert from a spectrogram back to an audio waveform.

Approach

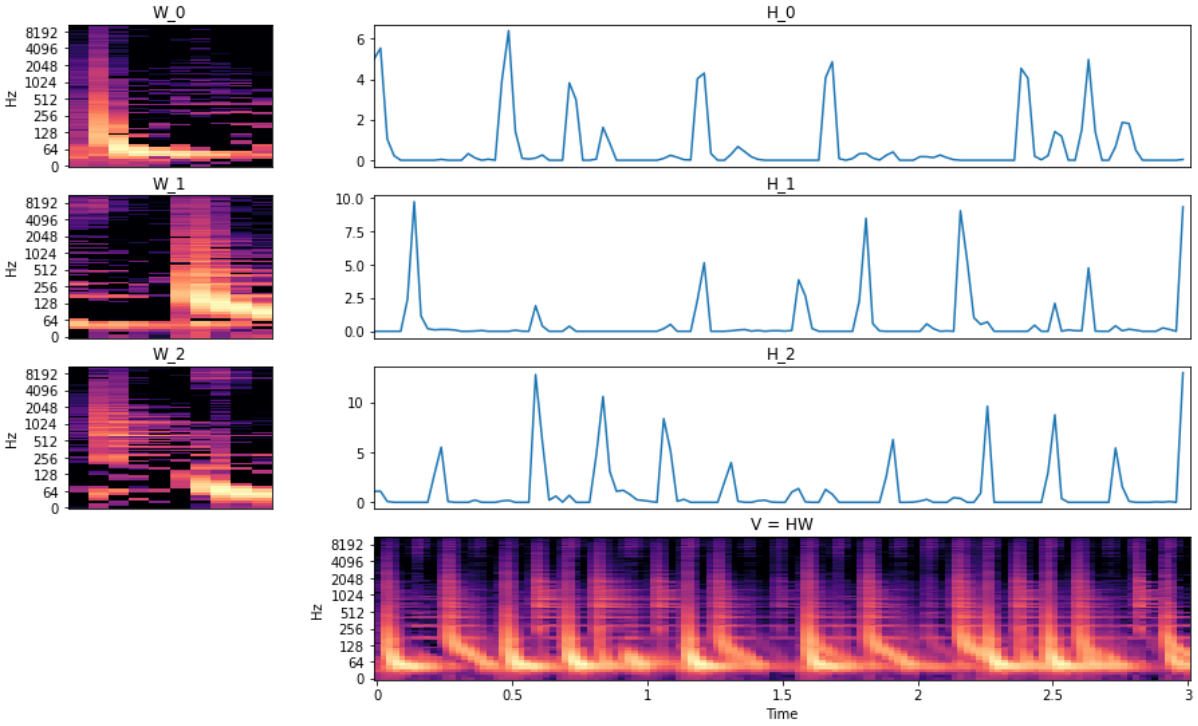
- Initially work on the rhythm aspect of the audio, that is the percussive instruments
- Rhythm is a key component in musical classification, and strongly associated with musical and cultural identities.
- Drums is the main focus of this prototype, as development work in other projects

  Pipeline:

  1 Input Audio -> 2 Extract Drums -> 3 Decompose Drums (NMFD) ->
  4 Transform Activations -> 5 Recombine Components ->
  6 Remix Components

1. **Input Audio** The user uploads a recording of their music. This can be a fully mixed track.
2. **Extract Drums** The audio is separated into individual tracks for each type of instrument in isolation. The percussion track is sent forward for further processing, while all the other tracks are stored to be recombined in the final step. This is achieved with an off-the-shelf model named Demucs.
3. **Decompose Drums (NMFD)** The drum track is decomposed and separated even further using a technique called Non-negative Matrix Factor Deconvolution. This has proven to be a very powerful method to decompose (and re-synthesise) percussive audio. See here for an example of a deconstruction of a famous drum recording.
   Essentially, this algorithm takes the spectrogram and factors it into two components, called the *activations* and *template* matrices, such that when they are multiplied together they recombine to reconstruct the spectrograms (and hence the audio). The template matrix contains information

about the sound of a particular element of the drum kit, whereas the activations matrix encodes the time these elements happen.
Below is an illustration of this step on a short drum sequence. The left column are the templates of the drum sounds (kick drum, snare drum, high hat), the right column are the activations of each of these sounds, with the bottom plot being the reconstructed spectrogram.



4. **Transform Activations** By manipulating these matrices and recombining, the original audio is transformed. For example, replacing the template matrix from one track with that of another drum kit will replace the sound of the drums in the input recording as if it was played on a different drum kit (e.g. resynthesise the drums of a rock track with Roland 808 drum computer sounds). Alternatively, by changing the activations matrix, the timing and the rhythm of the audio recording can be altered. In the initial stage of the project we will focus on this aspect only.
5. **Recombine Components** Recombination of the activations and template matrices is a simple step to reconstruct a transformed spectrogram. This spectrogram is then reverted back to an audio signal using the *Inverse STFT* operation to produce the transformed drum track.
6. **Remix Components** The isolated instrument tracks from **2** are brought back and mixed together with the transformed drum track to produce the final output delivered to the user.

Tasks

- **Transforming the activations.** Still need to develop a model that can take an activation matrix and output a new matrix that conforms to the target rhythm. This will perhaps be the bulk of the work of this project. A basic encoding model could learn the activation matrices for many examples of different genres, and learn to reproduce similar matrices that are in the target genre that are still close to the input track. This could be an opportunity to use an adversarial model?

- **Spectrogram reconstruction.** The spectrogram of an audio signal is a matrix of *complex numbers* in polar form which encodes both the frequency amplitude and phase. Proper reconstruction requires both of these components, otherwise the output audio sounds unpleasantly distorted (metallic and robotic). The formulation of the NMFD that has been implemented so far has only used the amplitude component, with the phase component being

re-used from the original spectrogram of the input. This is not a problem if the activations are not shifted, however if the rhythm of the drum hits are changed then the phase information will be mis-aligned. There are several ideas to resolve this problem that need to be explored.

- **Definition and classification of “genre” and Labelled data** As genre is a very loose and blurry categorisation schema, transforming a track to a target genre is a subjective outcome. Nonetheless, unsupervised learning of rhythmic styles does not require labelled data and allows the model to make distinction between styles in a novel way. From here, it is easy to see how the boundaries of genre *could* be defined by observing the regions in which tracks that are deemed *typical* (or even defining) of a genre are embedded into.

Risks & Mitigation

- Activations generator model: Outputs might not “feel” like the target genre, even if the activations do match songs in the target area of the music space.
  → Hand curate a selection of different activations that are typical of certain genres/styles and allow the user to select from this list. This is a compromise, but would work for at least a very early version and demonstration of the resynthesis techniques described above.

- There might not exist (or at least difficult to identify) a clustering of songs that is “natural” or corresponds to what we intuitively call *genre*.
  → Even if we cannot capture closely the notion of *genre*, clusterings in musical spaces exist (as has been shown by other work) and at least some of them will lead to “interesting” transformations. This is highly dependent on parameters (training, dimensionality reduction, etc.), which could even be controlled by the user in later versions, for them to explore different notions of *genre*.

Outcomes

- Timeline: July 1st; 10 days each working on the project, for a total of 20 days
- Deliverables: basic interface to transform input audio; data story on process and insights; research and programming infrastructure for further iterations on this (or similar) project
- Fee: €3000

Future Goals

- Apply rhythmic transformations to all other instrumentations
- Apply pitch transformations to harmonic elements
- Apply timbre transformations to all elements