Image2Music - Exploring Image Sonification

Aditya Adhikary

2015007

Indraprastha Institute of Information Technology

Delhi, India

aditya15007@iiitd.ac.in

Brihi Joshi

2016142

Indraprastha Institute of Information Technology

Delhi, India

brihi16142@iiitd.ac.in

Abstract

Data Sonification is a a popular paradigm that lets you 'hear' data. It is the use of non-speech audio to develop a visualisation or perception of the data. This data can be of various kinds - numbers, text, signals or even images. In this project, we provide a novel idea of sonifying images with the question - *Can we really hear what we see?* This project paves way for several ideas in the future. We can use it to sonify Comic strips, sceneries or even a collection of images like a collage or providing background scores for videos. Since this is our first experiment with Data Sonification, we introduce some key concepts, the tools used and the provide a detailed explanation to why we used them and finally, a taste of the experiments that we have conducted.

Introduction

Data inference and Data analysis has always been a saturated field, when it comes to innovative ways of data representation. Almost everyone has worked with bar charts, plot and other forms of visual representation of data. Data Sonification is an upcoming and emerging field in Computer Music that deals provides an audio-based representation of data. From a very broad perspective, the way a generic data sonification pipelines proceeds is as follows -

- The first step is to gather the data and organise it. After collating the data from any source – such as signals, images, text, etc, one needs to store it in an efficient manner. The more common way of organising the data is to store it in a CSV format.
- The next step is to preprocess the data. It includes normalising the values (for consistencies), removing unnecessary data, noise removal and also design-decision based preprocessing – such as converting the image pixels from RGB to HSV values.
- The last step or probably the most important step is to identify how to convert the data into audio. Generally, the data points are read and are use to control parameter of several UGens in SuperCollider. This is mostly a low-level generation. High level generation of audio requires one to conduct further analysis or modifications on the pre-processed data. Such as, converting all data points to MIDI sequences before playing them in a Pattern, etc.

Tools

Intuitively, SuperCollider seems to the be the perfect software that we can majorly use in any data sonification task. However, there are some drawbacks that we had

encountered while working on SuperCollider -

- There is absolutely no provision to combine image processing libraries and modules with SuperCollider, thus one needs to use extra softwares like MATLAB, Python, etc to perform the preliminary tasks.
- There are size and capacity limitations while loading large datasets onto SuperCollider, which then starts to crash.

As Python is the most library-rich resources, we lookout for some API can be call SuperCollider and control it from within. FoxDot is a python module that has been specially made to communicate between an external source like Python and SuperCollider.

FoxDot

FoxDot is a powerful tool that provides both the musical power of SuperCollider and the syntactic ease of Python. It functions in the following way -

- The coding and other analysis-heavy tasks are done on a Python Applet.
- FoxDot then connects to the SuperCollider backend, where it can process all the audio.

There are several reasons as to why we chose FoxDot as our main tool, despite there being other Python libraries like *Librosa* or *Py-Osc*. They are -

- The syntax is predominantly based on Python, which is extremely easy to use.
- Almost any Python library can be imported and used in the SuperCollider backend, which makes image processing a very easy task that can be done along with the sonification, rather than it being separately done.

Techniques

Broad Image Sonification Techniques

As described in "Application of Image Sonification Methods to Music" (Yeo et. Al, 2005), there are two major techniques for organizing time-independent data for auditory purposes, depending on whether they are pre-scheduled and fixed, or arbitrary and freely adjustable.

- Scanning - This refers to when the data is scheduled to be sonified in a fixed, non-modifiable order. The speed of scanning is usually fixed, and not allowed to be changed arbitrarily during sonification.
- Probing - If the speed and the location of sonification pointer can be varied by an operator arbitrarily during sonification, we classify this as probing. One may arbitrarily probe anywhere and anytime within an image. It requires some pre-assessment of the qualities or features of the image.

Till now, we have explored some techniques which are combinations of scanning and probing.

First Experiment - MNIST Data

We used a subset of the MNIST data, a commonly available dataset of handwritten images in grayscale (with pixel values ranging from 0 to 255) of a total of 350 images. You can find the jpgs here: https://www.kaggle.com/scolianni/mnistasjpg. We then read the images as 28x28 matrices, and stacked them from left to right, with a resulting matrix of 28x9800, and stored it as csv for ease of data manipulation.

- Simple Scanning of flattened images - The images are then flattened into a long array row-wise (274400 elements). A base frequency is decided, and an additional

"detune" frequency. We iterate through the flat array, and at each pixel value x, we calculate $frequency = (x * baseFreq) + detune$. This frequency is then played for a fixed period duration, such as 1 second. The resulting music is a steady stream of base frequencies (since most pixels are 0) interspersed with blips when the pixel values peak. This has been implemented using arrays and Tasks in SuperCollider.

- Scanning of flattened images, scale regulated - Using FoxDot, the same task is repeated, but the frequency is not required to be calculated individually for every pixel. FoxDot maps the integers to a pre-defined note on the chromatic scale of C, and the effect is different and more melodious than the previous part.

- Scanning of binarized images - In order to better understand and map the sounds from the lighter portions of the images to happier sounds and darker portions to more serious or ominous music, we binarized the stacked images based on a threshold to 0 or 1 pixel values, and then scanned through them. If the pixel value turned out to be 0, we assigned it randomly to a minor or a diminished chord in the scale, and if it turned out to be 1, we assigned it to a major chord. Hence, with the utilization of chords, the music sounded more relatable to the concepts of happiness and sadness, and sounded more sophisticated in general.

- Scanning of images, columnwise - With the advantage of Foxtrot playing multiple notes together as chords, we took the stacked images and iterated through every column (a vector of 28 integers) and played the values together as one chord. The result was often dissonant, and some notes stood out because they occurred multiple times in the column vector and their amplitudes added up. Adding advanced filters and VSTs resulted in more interesting musical effects.

- Experimental scanning using FoxDot's pre-defined sounds - FoxDot has a number of sounds which can be played by simply specifying a particular ascii character. For example, simple drum beats can be sequenced using characters like 'x' and 'o'.

In an attempt to make things more interesting, we map each pixel value encountered to a unique sound, and a 0 value to silence. This results in a cool concoction of beats and voices and various notes, and is the most rythmic out of all the experiments.

- Other various experiments - We also tried scanning the images columnwise, taking the mean of the first and last column, second and second-last, and so on. Hence, the pointers move from both sides of the stacked images. Also, we tried taking the absolute of the derivative along x and y directions. The gradient tells us about the rate of change of pixel intensity values and is useful in detecting objects and edges. We then flattened out the gradients and playing it using similar approaches as above. We shall extend these experiments with more approaches to processing the images before the due date demonstration.

Second Experiment - Coloured Images

For this experiment, we use two coloured images – One of a sunrise and other of a starry night. After exploring low level sonification in the first experiment, we moved on to learning higher level features from images – for example, emotions like happy or sad or descriptions like bright or dark. Like any coloured images, our images had three channels, the R, G and the B channel. However, we felt that in order to describe the contents of the image, we would need a more comprehensive representation of the image. This is achieved by the **HSV values**. Here, **V** is also called the **Value** or **Brightness** index of each pixel.

The procedure followed was as follows -

- After converting the image to an HSV format, all the V values were extracted and stored.

- A threshold was decided. All pixels that had a Brightness index less than this threshold were considered dark images and vice-versa.
- With each dark and bright pixel, we attached a Pattern that sounded different. For the bright pixels, we took major chords which felt very happy. On the other hand, for darker pixels, more grim sounding or minor scales were chosen.
- After adding additional FX like Reverb and Pass filters, the image was flattened and played pixel by pixel. Even though the result was not 'musical' per say, it did give a good distinguishing factor between the bright and the dark images.

Third Experiment - Creating more 'Musical' sounds

Till now, all the experiments that we conducted generated abstract sounds and audio pieces. In this attempt, we tried to create more 'musical' versions of Sonification of the data used in both parts above. Some techniques that we used were:

- to make extensive use patterns and in-built scales available in Catalog.
- Compose a percussion piece to go along with the compositional piece, which can be easily generated using *FoxDot Samples.*

We aim to improve upon this technique till our presentation due date.

Conclusion

In conclusion, we were absolutely delighted to explore this field, and we wish to continue on this field further by exploring concepts like sign post sonification, sonification of books, of comic books or of the mime/english videos in the near future!

References

- Application of image sonification methods to music - CCRMA

  https://ccrma.stanford.edu/ woony/publications/Yeo_Berger-ICMC05.pdf