SONIFICATION AND VISUALIZATION OF A GENETIC ALGORITHM TO APPROXIMATE ACOUSTIC TEXTURES

Jessica Thompson
Dartmouth College
jessica.thompson@dartmouth.edu

Michael Casey
Dartmouth College
mcasey@dartmouth.edu

ABSTRACT

A simple genetic algorithm learns the short-time fourier transform of a target static sound texture. The approximation gradually acquires information about the target sound via repeated semi-random modifications to the spectrogram. The learning process is sonified by inverting the estimated spectrogram at each iteration of the algorithm. The visualization is calculated by taking the inverse two-dimensional Fourier transform (2D ifft) of the spectrogram at each iteration. This sonification and visualization allows for the gradual evolution of the sound to be seen and heard. The goal is not to perfectly model the target sound, but rather to hear and see the learning process, whose path is unpredictable due to the random mutations to the spectrogram.

1. PREVIOUS WORK

Genetic algorithms (GA) have previously been used for sound synthesis [1, 2]. Usually a GA is used to solve for optimal synthesis parameters which are then used, after the learning process is complete, to synthesize a sound [4]. The present work is distinct from previous uses of GAs for synthesis in that the evolution of the GA itself is sonified and visualized.

2. SONIFICATION

The GA operates in the time-frequency domain by repeatedly setting sections of the spectrogram to semi-random values. The fitness function simply compares the euclidean distance between the section of the spectrogram that was set in the estimation and that same section of the target spectrogram. If the mutations do not reduce the distance, they are abandoned. Otherwise they are passed onto the next iteration where the same sections can be reset again. Thus at every iteration of the algorithm, the estimation will either take a step toward the target sound or it will not change at all. Magnitude and phase are learned independently. At each iteration, the estimated spectrogram is inverted to audio and concatenated to the final output file. This inverted spectrogram includes the mutations of that iteration, even if they will not be passed onto the next generation. Several parameters to the algorithm control the average size and variation in size of the sections of the spectrogram to be set, the length of the final output file, the number of iterations to skip when inverting to sound.

3. VISUALIZATION

The accompanying visualization is generated by taking the two-dimensional inverse Fourier transform (2D ifft) of the estimated magnitude spectrogram at each iteration (or at regular intervals) of the algorithm. Note that unlike the one-dimensional Fourier transform, which finds periodicities in a one-dimensional signal (e.g. audio), the two-dimensional Fourier transform finds periodicities in a two-dimensional signal (e.g. an image), in which the periodicities are in space rather than time. Since the 2D ifft receives only real values, the resulting output image is symmetric. At each iteration (or at a regular interval of iterations), an image is saved that corresponds to the audio segment that was inverted at that iteration. The resulting images are saved in an image sequence that is time-aligned with the corresponding sonification and combined in a video file.

4. DISCUSSION

The randomness in the GA means that the output of this process is unpredictable. Although the starting point (silence) and the goal end point (target sound) are known, the path from start to finish is not predefined. This is related to the work of David Dunn who discusses autonomy and emergent behavior in musical systems [3]. Is unpredictable behavior sufficient to label a musical system as "autonomous"? We argue that the tools of music information retrieval can help to probe questions such as these by providing artists with the means to create intelligent, sound generating systems.

5. FUTURE WORK

We would like to implement this process in an interactive environment such that the target sound material could come from the immediate environment. An interactive version would be ideal for an installation piece. Multiple instances of the algorithm could run simultaneously using different target materials. Additionally, feedback networks could be put in place in which multiple instances of the

the algorithm could trade targets after a certain threshold was achieved. Paths through many target sound textures could be defined. By switching targets partway through the learning process, the previous experience of any given instance of the algorithm would bias it's ability to approximate subsequent targets. Such a networked installation could generate unpredictable content indefinitely.

6. REFERENCES

[1] Batuhan Bozkurt and Kamer Ali Yüksel. Parallel Evolutionary Optimization of Digital Sound Synthesis Parameters. *Lecture Notes in Computer Science: Applications of Evolutionary Computation*, 6625:194–203, 2011.

[2] Michael Chinen and Naotoshi Osaka. Genesynth: Noise band-based genetic algorithm analysis/synthesis framework. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 179–182, 2007.

[3] David Dunn. Thresholds and Fragile States: A Descriptive Score for a Generative Sound System Using Networked Autonomous Chaotic Oscillators. 2011.

[4] Adam Johnson. Sound Resynthesis with a Genetic Algorithm. Technical report, Imperial College London, 2011.