Algorithm as Determinant of Form in Music

Halley Young
Computer Science
Department University of Pennsylvania
Philadelphia, PA USA
halleyy@seas.upenn.edu

Abstract

In this creative submission, algorithmic processes are explored as ways of establishing form in music.

Introduction

In addition to having different harmonies, instruments, and rhythmic styles, different cultures are associated with different ways of organizing music. In the 17th/18th-century common practice style, there are well-known conventions for developing both small-scale and large-scale patterns of sound; for instance, the antecedent-consequent structure of the period establishes order on the range of 5-10 seconds, while sonata form is a highly involved formal device for ordering entire movements (or even suites of movements) (Caplin 2000). The 20th and 21st centuries are associated with new formal devices, including process-based music in which events occur such as "reaching a limit beyond which the preceding process cannot continue" or "the arbitrary stopping of a process" (Tenney 1973). The processes used in this way can often be described mathematically, such as the use of rhythmic cannons by Messiaen (Carl Bracken 2009).

Simultaneously, since at least the 1950's, another type of sonic art has become popular - that of sonification. Sonification is the practice of converting data to sound (Fry 2005). It has been described by some as having actual scientific value for apprehending patterns in scientific data. However, its status as music is up for vigorous debate, as the title "Sonification ≠ 6 Music" in the Oxford Handbook of Algorithmic Music suggests (Scaletti 2018). This has to do with the fact that the most literal sonifications often willfully ignore the principles of what a Western audience finds pleasant (as well as features which are arguably more universal) in favor of keeping the meaning as explicit as possible.

I am interested in work at the intersection of these two areas. Specifically, I am exploring how to structure music using algorithmic processes. The distinction between this enterprise and traditional explorations of form is that when form is determined by a computational process, it might take a shape that was not envisioned (or even imaginable) to the composer. However, this enterprise is also different from traditional sonification, in that I explicitly want to generate output that is heard as music, which translates into a much less literal interpretation of the algorithmic process.

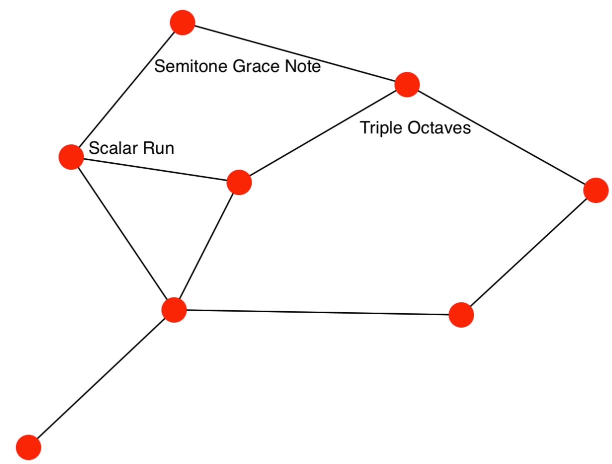

Figure 1: The structure underlying the convolutional graph network composition

Sonification of Convolutional Graph Networks

One of the most recent trends in deep learning is Convolutional Graph Networks (Defferrard, Bresson, and Vandergheynst 2016), which allows for the application of neural networks to data structures in the form of graphs. While I highly suspect that in the future music will be made through the usage of CGN's, here I attempt to illustrate the essential functioning of CGN's - how they "smear" the vector values of each node around the graph progressively through local averaging of the nodes. The graph structure illustrated in Figure 1 is sonified. Each node is assigned a different instrument and a different musical idea. The musical ideas are each not traditional motifs, but rather ideas that can be implemented to varying degrees - for instance, it is possible to play none, half, or all notes with a semitone-grace-note before it. At each time step, the value of each node is updated by the mean of the value of its neighbors - for instance, the node that is associated with semitone grace notes will to also be affected by the ideas of scalar runs and triple octaves, which are associated with two neighboring nodes. As the activation spreads, musical ideas become more homogeneous.

Sonification of Convolutional Graph Networks

In the second example of such a piece, I try to describe in music the execution of a short imperative program. This is “program music” in two senses - it fits the traditional definition of “program music” in that it is defined by a narrative, but the narrative is a piece of code rather than a natural English story! I invented ways of representing the specific concepts seen in an imperative program. For instance, I model a “while-loop” by repeatedly changing a motif, but stopping the process only when the final note finally changes. Motif transformations represent function application on variables. Under the melody is a constant, fast-paced “clock”, which is ticking both before the program initiates and after it terminates.

Figure 2 shows the code of the sonified algorithm.

```
x = f()
y = g()
h = stutter(x)
g = stochInvert(y)
while (g[-1] != "D") {
  g = perturb(g)
}
z = h()
for (i = 0; i < 2; i++) {
  bool_var = "G"
  if (z)
    bool_var += 1
  print(bool_var)
  z = h(z)
}
x = perturb(x)
print(x[-1] == "F#")
exit()
```

Figure 2: The code of the sonified algorithm

Sonification of Towers of Hanoi

The Towers of Hanoi is a classical game involving rings being moved between pegs which is used to teach about recursion. The way to optimally play is simple to describe if you know the algorithm, but very difficult to arrive out without that knowledge or the ability to carry out the recursive procedure it involves. The algorithm has some symmetrical and some asymmetrical qualities - while at every level the task is to move the top $n - 1$ rings twice, moving the nth ring once in the middle, the exact order of what is being moved there follows an interesting fractal-like pattern. As such, it became an excellent target for algorithmic composition. I assigned the flute a different melody at every time step according to the type of movement, and contracted or expanded each time step according to the size of the ring being moved. The instrumentation of the rest of the instruments was related to the teleological movement being effected at higher levels due to the actual movement (e.g., the ring of size 4 was being moved to the right in order to move the ring of size 3 to the left, which was done in order to move the ring of size 2 to the right, etc.) In post-processing, after running the Python script, I changed the dynamics and some of the pitch material of the inner voices (while keeping timbre, attack points and rhythm intact), which gave it more of a directed feeling than would be possible otherwise using such an algorithm.

References

Caplin, W. 2000. *Classical Form: A Theory of Formal Functions for the Instrumental Music of Haydn, Mozart, and Beethoven*. Oxford Press.

Carl Bracken, Gary Fitzpatrick, N. M. 2009. Tiling the musical canon with augmentations of the ashanti rhythmic pattern. *Bridges: Mathematics, Music, Art, Architecture, Culture*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc. 3844–3852.

Fry, S. P. 2005. A brief history of auditory data representation to the 1980s. In *First Symposium on Auditory Graphs.*

Scaletti, C. 2018. Sonification ≠ 6 music. *Oxford Handbook of Algorithmic Music*.

Tenney, J. 1973. Form in 20th century music. *Dictionary of Contemporary Music*.