**Encoding Data into Sound and Music:**  
**A Live-Coding Approach**

**Takahiko Tsuchiya**  
Georgia Institute of Technology  
Center for Music Technology  
840 McMillan St., Atlanta, GA 30318  
takahiko@gatech.edu

**Jason Freeman**  
Georgia Institute of Technology  
Center for Music Technology  
840 McMillan St., Atlanta, GA 30318  
jason.freeman@gatech.edu

**ABSTRACT**

This study proposes the use of external data in live-coding performance as a mechanism to achieve constraint- or theme-driven creativity. It discusses the analysis of data structures and transforming them to various musical structures, while addressing common issues such as the choice of data-processing techniques in time-critical context, handling of previously unseen data, aesthetic presentation of data, and comprehension of data through sound.

1. INTRODUCTION

Live coding provides a unique experience where the artist’s design process is shared with the audience in real time. It exposes the generative mechanism of the sound (and/or visuals) and, to various extents, the dynamics of decision making and thought processes of the performer. This display of “creative” processes, however, can sometimes merely be a replication of past practice where the performer already knows what processes work. Even in improvisational or collaborative performances, the performer may resort to safe or memorized expressions more than engaging in real-time musical composition. To provoke a more live and unpredictable outcome, we propose the introduction of “data sonification” factors in live-coding performance. Sonification is a use of non-speech audio to represent data (Walker and Nees 2011), which is parallel in many ways to data visualization.

A live-coding performance of sonified data may seem challenging, as it adds layers of technical complexity and also may impose constraints on the live-coding practice, in that the coder somehow has to utilize the elements of data for effective musical expressions. Constraints are, however, not creative limitations. Magnusson, for example, discusses the creativity driven by the constraints of culture, physicality (e.g., instruments), and technology (2010). External data in live coding can also provide compositional materials or even high-level themes to the performance.

In this paper, we address the process of real-time transformation of data to music as “live encoding.” We will describe several simple and generalized analysis and mapping techniques for connecting data to musical structures in live encoding, while discussing technical and aesthetic challenges such as the control of dynamics and preserving the characteristics of data. We will first review related work, followed by an overview of our custom live-coding environment, Data-To-Music (DTM). We then discuss the use of the statistical information and structural components of data in musical mapping, and discuss the conflicting problems of aesthetic and functional data sonification. Example code listings using DTM, indicated with the (*) mark, are hosted online in the following website: https://gtcmt.github.io/DataToMusicAPI/iclc2016.

2. RELATED WORK

The use of external data in a real-time audio programming environments has been well-explored by both live coding and sonification researchers. For example, a live coder may design an instrument on-the-fly while receiving sensor data from a mobile phone controlled by another performer (Lee and Essl 2013). BRAID combines live coding (Gibber: Roberts and Kuchera-Morin 2012) with graphical-user-interface development, where the user interaction and audio processing are connected in real time in a model-view-controller paradigm (Taylor and Allison 2015). Threnoscope also employs a graphical or tangible user interface together with a terminal live-coding window, providing access to higher and lower abstractions for generating musical textures (Magnusson 2014).

For development of sonification systems, Rohrhuber et al. argue the advantages of text-based just-in-time (JIT) programming, such as the communicative value of dynamically swapping instruments, iterative scaling of
mapped data, and parameter tunings (2005). SonEnvir, a generalized sonification design system, indeed embraces a JIT code interface for its precision over GUI and explicit tuning processes being easy to be documented (De Campo et al. 2004).

3. LIVE-ENCODING UNSEEN DATA

3.1 Data-To-Music API

A “live encoding” of data typically involves multiple (often cyclic) steps, including analysis, transformation, and mapping. We experiment with these processes using the Data-To-Music (DTM) application programming interface (API), a browser-based musical sonification library. Unlike general-purpose audio languages (such as SuperCollider and Csound) with unit generators for constructing a complex audio graph, DTM primarily manipulates data arrays as a unified interface to synthesize and sequence audio events. This allows low- and high-level musical expressions such as wavetable, additive, or FM synthesis, filtering, time and pitch quantization, rhythmic interpolation, and time stretching*. With the data-dependent nature of synthesis and sequencing, slight changes to mapped data can propagate through the structured sound, like a “ripple” effect (DeLine and Fisher 2015).

In a typical live-encoding procedure in DTM, we would first acquire a data set or subset (e.g., CSV file, web-socket stream), preprocess the (sub)set with statistical and structural analysis while finding candidate musical structures such as rhythmic patterns and articulations, transform the data, and map to one or many synthesis parameters. In such a workflow, the authors find that the most basic technical requirements are the availability of computationally low-cost analysis and transformation methods and a mapping system for naturally and gradually injecting data to sound.

3.2 Handling Unseen Data

Any new data usually come with different forms than others (e.g., type, range, dimensionality, cardinality, rate), and it is an unavoidable step in live encoding to make them usable for musical mappings. Our previous research describes the adaptive mapping interface for the synthesis and sequencing parameters in DTM, in which any size or range of numerical array can be mapped to any synthesis parameters (Tsuchiya et al. 2016). Similar to the data-stream pipelining in UrSound (Essl 2010), this reduces the concern of having to align the sampling rate between different parameters and even between data dimensions. For non-numerical or discrete data, DTM facilitates quick computation of the distribution of symbols (i.e., unique values) or sorting with certain criteria such as character/symbol length and text-edit (Levenshtein) distance to a reference symbol. The use of distribution, mapped back to each data point, correlates well with the proportional aspects of musical structure. For instance, a distribution can be used for creating timbres with additive synthesis by mixing harmonic partials*. Other direct use of distribution values may be such as micro-rhythmic or articulation modulation by controlling the duration of note sequences*. Taking the idea from entropy coding, such as Huffman and Tunstall coding (Sayood 2012), where rarer symbols get longer code words, distribution values can be mapped inversely to musical parameters (such as volume), where rarer symbols get more presence in music while frequent symbols stay less “annoying.”

Similar to proportions, a particularly important musical quality is the balance of repetition and variation. Many live-coding performances employ the musical style with repeating patterns, gradually evolving or introducing new patterns as the piece grows. The amount of repetitions and variations, often related to expectation and (pleasant) surprise for the listener (Huron 2006), may be manually controlled by the performer, but also can be mapped from the repeating symbols of data. One way of quickly estimating the balance of repetition and variation in a given distribution is computing the information entropy (Shannon 1951), which indicates skewed distribution or repeating nature of certain symbols as the value is comparatively lower*.

3.3 Structural Analysis and Aesthetic Presentation of Data

Even when we adapt the range and type of the input data, it is easy to imagine mapping unseen sequences of values to synthesized voices in real-time causing a musical havoc. The majority of real-world data (e.g., biosensors measuring muscle or brain activities) are noisy and not suitable for creating natural musical expressions. Improvisational or experimental live coding in general already has the relatively high risk of the performer facing “failures and ugly moments” with the lack of control of musical quality or organization (Collins et al. 2003). With data-driven live coding, the risk becomes much bigger with the reduced control by
the human composer. How can we shape, or use data itself to control, otherwise highly unpredictable behavior of data-driven sound?

One approach is to systematically reduce the potential noise by extracting and isolating the “structural” and random components of the data. The above-described statistical analysis, such as the distribution of symbols, does not provide any structural information, as they assume no causal relationships between the sampled values. However, we may come across sequential, periodic, or grammatical relationships between data points, which can provide high-level characteristics of the data. To estimate and extract such structural components in real time, we may employ computationally low-cost analysis techniques such as low-order linear regression or periodicity detection with average magnitude difference function (AMDF)”. Once we separate the structural and random component, such that

\[ xn = fn + err, \]

where \( x \) is the original data sequence and \( n \) is the sample or time index, we may map them individually; for example, the structural part purely as a framework (e.g., rhythmic or melodic layout), and the random component as the actual trigger or exciter. Similar “structure + random” models are used in various applications, such as speech coding with linear prediction (Vaidyanathan 2007), in which the random component is often creatively replaced with simplified or more characteristic signals. The random component, sometimes called an “innovation” or “excitation” signal in audio processing, may be used as the amplitude or triggering control for a probabilistic sequence. As we successfully separate the structural and random components, the random part becomes closer to a white noise with a fixed amplitude. With its fixed and controlled range, we can actively scale the signal to gradually and naturally introduce new musical materials with data”.

3.4 Data Comprehension Through Sound

Lastly, we will consider the challenge of making sense of the data in live-encoding performance. As we analyze, transform, and map data to musical dimensions, we are likely to distort or lose the details of the original data. The more we process, the more it becomes indistinguishable from randomly-generated or arbitrarily-chosen sequence of values, especially for the audience. Indeed, many musical sonification compositions involve arbitrary elements or oversimplification of data. Dodge, the composer of Earth’s Magnetic Field (1970), for example, describes how he "freely" chose the timbre and used high-level interpretation for the spatialization (Thieberger and Dodge 1995). Perhaps conflicting with the human control of musical quality by, for instance, removing the unwanted randomness, but how can we preserve the details of the original data and present them effectively as a musical structure?

One systematic approach is keeping track of the difference or distortion incurred by the transformations and minimizing them. For example, we may choose to pitch quantize continuously-valued data to a known musical scale (e.g., major, minor, the church modes, etc.). Since the musical pitch quantization is usually a non-uniform or nonlinear process (i.e., the major scale, for example, is an asymmetric scale), we may write a function that iteratively applies each of the candidate scales to the whole or sampled data points and measures the signal-to-noise (SNR) ratio of the quantization process, with

\[ \text{SNR}_y = i=1Nxi2/i=1Nerr2, \]

where \{xi \( i=1,2,3,...,N\)\} is the original data sequence. Finally, we can select the musical scale that gives the minimal distortion”.

4. CONCLUSIONS

In this paper, we described real-time integration of external data in live-coding performance, and some of the common challenges and strategies for translating data to musical structures. The examples are implemented using the DTM API, available online at https://github.com/GTCMT/DataToMusicAPI/. Although the techniques that we introduced may be applied for other musical mappings that were not covered in this discussion, they may be useful for guiding creative decisions in live-coding performance, as we explore new data with the audience.
REFERENCES


Tsuchiya, Takahiko, Jason Freeman, and Lee W. Lerner. 2016. "Data-Driven Live Coding with DataToMusic API." In Proceedings of the 2nd Web Audio Conference (WAC-2016), Atlanta.
