Interfacing Sounds: Hierarchical audio content morphologies for creative re-purposing in earGram 2.0

Gilberto Bernardes
INESC TEC and University of Porto, Faculty of Engineering
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
gba@fe.up.pt

ABSTRACT

Audio content-based processing has become a pervasive methodology for techno-fluent musicians. System architectures typically create thumbnail audio descriptions, based on signal processing methods, to visualize, retrieve and transform musical audio efficiently. Towards enhanced usability of these descriptor-based frameworks for the music community, the paper advances a minimal content-based audio description scheme, rooted on primary musical notation attributes at the threefold sound object, meso and macro hierarchies. Multiple perceptually-guided viewpoints from rhythmic, harmonic, timbral and dynamic attributes define a discrete and finite alphabet with minimal formal and subjective assumptions using unsupervised and user-guided methods. The Factor Oracle automaton is then adopted to model and visualize temporal morphology. The generative musical applications enabled by the descriptor-based framework at multiple structural hierarchies are discussed.

Author Keywords

Hierarchical audio description, Concatenative sound synthesis, Temporal morphology, Re-purposing, Information visualization

CCS Concepts

•**Applied computing** → **Sound and music computing;** Performing arts; •**Information systems** → *Music retrieval;*

1. INTRODUCTION

Interacting with musical audio descriptions is a pervasive methodology across creative music workflows. Representative examples include beat-matching [21, 3], harmonic mixing [7], sound morphing [11], concatenative sound synthesis (CSS) methods [26], and the promotion of active listening [16]. Across commercial applications, descriptor-based processing typically runs on the back-end with little to no interference from the user. Yet, more challenging transformations require the user to select and manipulate from the available pool of audio descriptors. A limitation on the wide adoption of these descriptor-based transformations by the techno-fluent musicians has been pinpointed by their grounds on digital signal processing terminology and methods, rather than music theory and practice [8]. In this context, earGram [8] presented the first steps towards a musician-friendly framework for CSS. In what follows, a revamp of the long-outdated application towards a hierarchical enhanced descriptor-based framework is detailed.

Audio sources in `earGram 2.0` are the raw creative material which imprint a particular 'sound' and 'formal structure' to generative re-purposed (musical) audio. A diversity of audio sources from personal field recordings to commercial music releases can be adopted. Yet, conversely to most CSS applications, `earGram 2.0` does not target disparate audio source collections. Instead, a *corpus* of audio *units* result from a unique audio source (i.e., one user-defined audio input file), whose temporal morphology has been *intentionally* composed or recorded.

Three major contributions in `earGram 2.0` are reported. First, it features a more efficient computational framework for a musician-friendly descriptor scheme, grounded in prominent qualities of music notation at the rhythmic, harmonic, timbral and dynamic levels. Second, audio signals are described at multiple hierarchies of musical structure. Perceptually-guided spaces, where distances indicate similarity, are defined from low-level audio descriptions. Mid-level categories are then guided by user-input parameters of unsupervised clustering methods, towards contextual fine-tuned formal music categories. Third, flexible contexts from multiple audio viewpoint systems form a discrete and finite alphabet from which a Factor Oracle (FO) automata is drawn. Ultimately, FO reveals temporal morphological patterns in the audio source and finds optimal splice (looping) points from repeating sub-sequences. Generative music applications driven from the descriptor-based framework at various degrees of structural and sonic content re-purposing are discussed.

The remainder of this paper is structured as follows. Section 2 details the methodological rationale behind audio content-based processing for music generation and identifies related work. Section 3 outlines the `earGram 2.0` architecture, in particular the design, and content-driven information flux of the hierarchical audio description modules. Section 4 details the hierarchical description of rhythmic, harmonic, timbral and dynamic audio content. Section 5 builds on the proposed descriptors to define a discrete and finite alphabet from which the audio source temporal morphology is modelled and visualized using a Factor Oracle automaton. Section 6 discusses generative music applications which stem from the hierarchical audio descriptions. Finally, Section 7 summarizes the contributions of the work and avenues for future work.

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

2. RELATED WORK

This work touches upon several existing computational methods and technologies for structural modeling [13, 14, 5], audio description, and sound synthesis [26]. Moreover, it enables the generation of musical audio by re-purposing an audio source content with varying degrees of sonic and hierarchical-structure reattainment. [1]

The task builds upon the early days of music informatics [13] in modelling music structure manifested as symbolic information. Based on ideas from information theory [5, 12] and the postulate of music structure as a low entropy phenomenon, early algorithms converge to the notion that musical structure can be modelled and predicted algorithmically from a sequence of past events. To this end, probabilistic automata and Markov models were widely adopted to identify repeated sequences and variations. The operational property of such algorithms is to provide the conditional probability distribution or paths over an alphabet (i.e. entire collection of musical events), given a preceding sequence, i.e. a *context*. This distribution can then be used to unpack temporal morphology and generate new sequences while maintaining structural resemblance.[2]

An established representation of symbolic music is the multiple viewpoint systems [14]. It consists of independent views of the musical surface, each of which models a specific type of musical phenomena. It includes *basic* attributes such as duration and pitch, but also *derived* attributes such as intervals and longer-structural dependencies at a particular hierarchy, such as the first event in a bar or a phrase. When applied to musical content manifested as audio signals, the previous approach is non-trivial. The low-level and noisy sample representation of audio signals typically requires additional complex processing layers to increase the level of abstraction in the representation towards a discrete and finite alphabet. To this end, content-driven methodologies for audio segmentation, description and matching, as well as pattern recognition and structure discovery have been adopted [8, 4, 5, 26, 25]. On the other hand, the timbral dimension in audio signals and performative aspects, with its underlying layers of expression such as micro-timing, is greatly enhanced.

The Audio Oracle (AO), an extension of the Factor Oracle (FO) algorithm, is a representative state-of-the-art example of such predictive models for processing audio signals [5]. FO and AO rely on parsing the incoming signal into an alphabet of states from continuous ranges of audio descriptors. A useful property of AO is that it allows recombination of repeating sub-clips in a manner that assures continuity between splice ('looping') points. AO effectively accomplishes music generation tasks for style imitation and texture synthesis, where variations of audio source content are derived through the reshuffling of sub-clips while maintaining an overall *sonic* and *structural* resemblance to the source [4, 5]. AO uses an information rate measure to inspect the reduction in uncertainty about a signal when past information is taken into account [5]. Audio frames with an information rate below a given threshold are grouped into the same state of the oracle. Despite its robust analysis, this measure does not allow for a highly controllable reconfiguration of the states 'quality'. To this end, this paper revisits the main argument behind `earGram` [8] and presents a fluid method that enables musicians to select and parameterize multiple audio viewpoint descriptions in light of user preferences, application domains and the nature of the audio source, without *a priori* assumptions about its content.

[1] Due to space constraints, a comprehensive review of related work is beyond the scope of this paper, the reader is referred to [19, 8, 22].

[2] Please refer to [22] for a comprehensive literature review on music sequence modelling and generation.

3. EARGRAM 2.0 ARCHITECTURE

Fig. 1 shows the architecture of the content-driven audio processing in the `earGram 2.0` software. From left to right, there is an increased degree of abstraction on the audio signal description, from sample to macro levels of description.

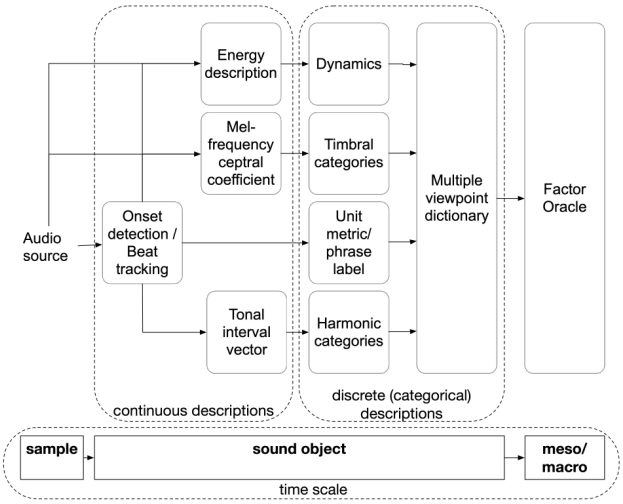

Figure 1: Architecture of `earGram 2.0` content-based description modules. Left to right layout denotes increasing structural hierarchies or time scales. Directional arrows indicate the data flux across the component modules of the system inside square bounding boxes.

From the audio source amplitude values, a novelty function from short-term features is summarized to segment the audio source into component audio units. At the unit level, continuous (e.g., signal energy) and categorical (e.g., dynamics) descriptions at the fourfold rhythmic, harmonic, timbral, and dynamic musical levels are extracted. A user-driven discrete and finite alphabet from flexible multiple audio content viewpoints is then constructed. It aims to group audio units into a smaller number of symbols, thus promoting higher degrees of redundancy (low-entropy). Finally, a FO automaton is drawn from the alphabet to model the temporal morphology of the audio source and expose repeating patterns of audio units. `earGram 2.0` further includes generative component modules, which enable musicians to *interface* with the audio source content. The generative music applications stemming from the descriptors framework are summarized in Section 6.

4. DESCRIBING AUDIO

Aiming to capture the nuances of musical structure morphology imprinted in an audio source, `earGram 2.0` adopts a small collection of four rhythmic, harmonic, timbral, and dynamic descriptors. They capture the most relevant dimensions of music–notably those enforced by musical notation and vastly manipulated by the art of musical composition. Each descriptor has a twofold continuous and categorical manifestation, which relates to a move from a low to mid abstraction level, i.e., from physical to formal attributes. The rhythmic description is responsible for defining the component audio units from the audio source, S. The rhythmically-aware audio unit segmentation informs

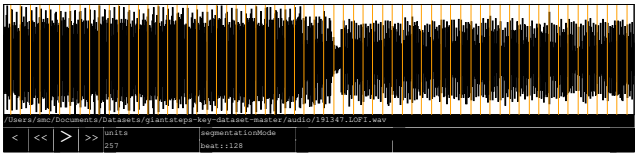

Figure 2: Audio source waveform visualization with overlaying unit boundaries in `earGram 2.0`.

remaining harmonic, timbre and dynamics descriptions. A single descriptor value per audio unit m is computed across the entire corpus of $1 \leq m \leq r$ audio units–where r is the index of the last unit; thus r equals the total number of units.

4.1 Rhythmic Description

Rhythm in `earGram 2.0` is the first description stage. It aims to extrapolate a collection of component audio units from the continuous audio source waveform. To this end, a spectral growth novelty function in the perceptually-aware Bark scale is computed using the `timbreID` library [9]. From the bark novelty function, two mid-level rhythmic information can be extracted: note onsets and beat locations, using algorithms from [10] and [15], respectively. Users define the temporal segmentation prior to the definition of the audio source. A collection of $m = m_1\, m_2 \ldots m_r$ audio units, defined by their onset time and duration $(m_{i+1} - m_i)$ is output to the remaining descriptive modules. Fig. 2 shows the interface in `earGram 2.0` which provides a visual feedback of the audio source segmentation to the user.

After the automatic segmentation of the audio source, a numeric label denoting temporal regularities at the bar or phrase level is assigned to each audio units across the entire corpus. Based on a user-defined value s, units are sequentially labelled by $\mod s$. Although the unit number may not align to metrical grids or phrase locations, it guarantees the same $\mod s$ label across sequential patterns. By default, if no user-defined value is defined, no regularities are assumed by adopting $s = 1$.

4.2 Harmonic Description

Harmonic relations across units are expressed in a perceptually-inspired continuous pitch space, rooted in the recent music theory literature on the DFT of pitch class sets [24, 32, 2]. The work has been expanded to the audio domain by mapping chroma vectors, $c(n)$, normalized to unity, $\bar{c}(n)$ into Tonal Interval Vectors, $T(k)$, using a distorted discrete Fourier transform [6, 7], such that:

$$T(k) = w_a(k) \sum_{n=0}^{N-1} \bar{c}(n) e^{\frac{-j2\pi kn}{N}}, \qquad k \in \mathbb{Z} \quad \text{with} \quad \bar{c}(n) = \frac{c(n)}{\sum_{n=0}^{N-1} c(n)} \tag{1}$$

where N=12 is the dimension of the chroma vector, $c(n)$, normalized to unity, $\bar{c}(n)$. k is set to $1 \leq k \leq 6$ for $T(k)$, since the remaining coefficients are symmetric. The weights, $w_a(k) = \{3, 8, 11.5, 15, 14.5, 7.5\}$, adjust the contribution of each dimension k of the space to comply with empirical ratings of dyads consonance for musical audio. Please refer to [7] for a comprehensive description of the weights values derivation. $T(k)$ has been shown to elicit many properties with music-theoretic value. Of note, $T(k)$ phases, $phases(k) = \angle T(k)$, revels aspects of tonal music, notably harmonic proximity across vectors, with unforeseen accuracy, in terms of voice-leading [29], tonal regions modeling and relations [30].

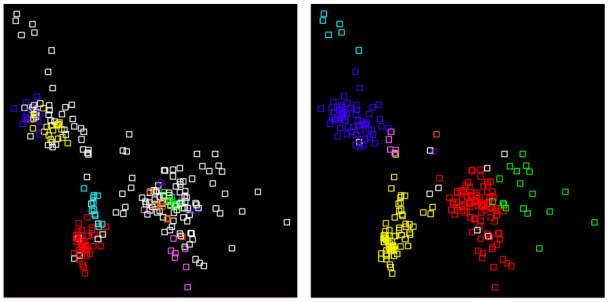

Figure 3: Timbre space visualization in `earGram 2.0`. The seven largest clusters are represented by colors other than white, which denotes the remaining (smaller) clusters and outliers. Both images adopt a minimum number of 4 units per cluster, and a quality threshold of 0.1 and 0.4 on the left and right images, respectively.

Stemming from early experiments with multiple clustering algorithms in `earGram` [8], QT-clustering [17] is adopted to partition the 12-dimensional (12-D) and continuous $T(k)$ space into (discrete and finite) harmonic categories. It discovers patterns without *a priori* domain knowledge, such as the total number of the clusters, and allows for the detection of outliers (e.g., units with percussive content). Furthermore, to comply with the subjective nature of the task, QT-clustering adopts two user-defined parameters to partition the elements: a cluster quality criteria defined as a distance threshold between elements and a minimum number of elements per cluster, which can be defined by the users. In the case of finding harmonic categories, elements are audio units defined by $T(k)$ and the cosine distance is adopted in capturing harmonic relations. The algorithm starts by considering all candidate clusters and retains the largest cluster at each iteration, thus being replicable on multiple runs and more predictable for different parameters. To guide the user-defined parameters of QT-clustering, the 12-D $T(k)$ is reduced to a two-dimensional (2-D) representation by assigning the axis to the largest weighted coefficients, $phases(4) = \angle T(4)$ and $phases(5) = \angle T(5)$. In the resulting representation, the seven largest clusters are coloured, in a similar fashion to the spaces shown in Fig. 3.

4.3 Timbral Description

The commonly-used 12 first Mel-Frequency Cepstral Coefficients (MFCC), excluding the DC coefficient, from a 38 mel-scaled filter bank, is adopted as a representation of the unit's timbre. MFCCs are computed via a discrete cosine transform of the log-transformed power of Mel spectra. They represent the shape of an audio signal's spectral envelope, encoding increasingly finer scales of spectral detail. MFCCs are standard in various tasks in audio content analysis and music information retrieval and have also been proposed as descriptors for timbre perception [27].

The same motivation behind the adoption of QT-clustering for defining harmonic categories is followed in the timbre domain. Cosine (or angular) distances across 12-D MFCC vectors define the distance metric used to partition the corpus into categorical timbre clusters, irrespective of harmonic and dynamics content [23]. To guide the user in assigning the cluster parameters, the 12-D MFCC are reduced to a 2-D visual representation using Star Coordinates [18]. As shown in Fig. 3, the resulting representation highlights the seven largest clusters in different colours.

4.4 Dynamics Description

From the audio units root mean square (RMS) energy averaged across audio units 2048 sample windows, a number of user-defined dynamics categories are computed. To this end, a context-aware division of the entire set of RMS values is done by iteratively dividing the ordered set of values into two halves according to their median value. A fourfold division equals to splitting the range into quartiles. This efficient computational split is contextual to the audio source signal properties. Yet, it considers static dynamics only, i.e., averages the RMS energy across the entire unit duration, disregarding morphological variations such as crescendos and decrescendos.

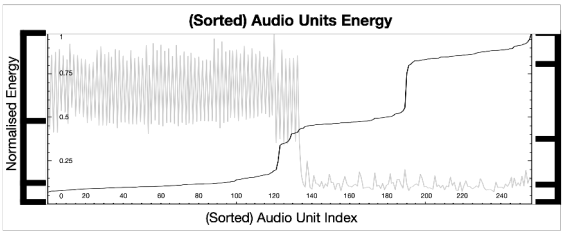

Figure 4: Audio units energy values in original (light gray) and ascending order sequence (black) from the audio source shown in Fig. 2 A threefold and fourfold interval division of the energy values are shown on the left and right sides of the plot, respectively.

5. TEMPORAL MORPHOLOGY

The finite-state automaton Factor Oracle (FO) P is a time- and memory-efficient string-matching algorithm, which is adopted in `earGram 2.0` to unveil an audio source's temporal morphology. In detail, FO has $r + 1$ number of states, whose links identify at least all factors of the structurally-segmented sequence of units S in the audio source. To this end, FO parses the original sequence of audio units, represented by a discrete and finite alphabet Σ, i.e. a set of symbols. For a full description of the FO construction, please refer to [1].

Repeating patterns in the FO are denoted by two types of links between states: factor links and suffix links. Factor links are forward transitions between all consecutive states $i - 1$ and i or between state i and a given forward state (where $1 < i < r + 1$). Forward links, shown as upper arcs in Fig. 5, indicate paths in the original sequence that can produce similar patterns with alternative continuations by moving forward. Suffix links, $SP(i)$, shown as lower arcs in Fig. 5, are backward transitions between state i and the 'leftmost' largest found similar sub-clip in P. Following [4], reverse suffix links, $rSP(i)$, are adopted to promote a greater number of paths across the states, while guaranteeing the same properties as the suffix links.

Audio units are represented in the FO by a discrete and finite set of symbols, from an alphabet Σ. The symbol σ attributed to each audio units is computed by parsing the multiple audio viewpoints for the entire unit collection. To each distinct multiple audio viewpoints, a new and unique σ symbol is created. A sequence of discrete symbols representing the temporal morphology of audio units $S = \sigma_1 \sigma_2 ... \sigma_r$ is then incrementally created. Audio units with equal multiple viewpoint descriptions adopt the same σ symbol and are considered to have a high degree of 'perceptual' affinity. The larger the number of all unique symbols σ, the larger

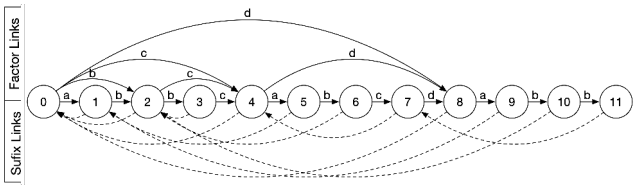

Figure 5: A Factor Oracle structure for the string abbcabcdabb. States are labeled by integer numbers and alphabet symbols by letters. Upper and lower arcs denote factor and suffix links, respectively.

the degree of novelty in the musical structure of the audio source is assumed.

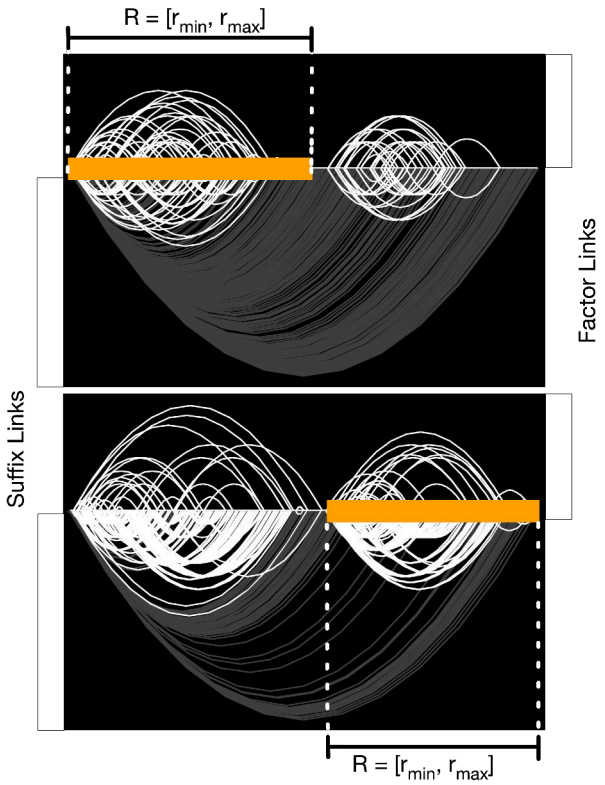

Figure 6: Visualization of the FO automaton in `earGram 2.0` of the audio source shown in Fig. 2. The original sequence of the audio units in the source runs from left to right. The two FO representation result from the different timbre space in Fig. 3. Remaining attributes are unchanged. The center (orange) bar allows the user to define the temporal range R of the audio source to be adopted in re-purposing applications.

The construction of the alphabet is instrumental in eliciting the temporal morphology of the audio source. It is ultimately driven by user parameterization of the categorical descriptors spaces, which are then concatenated as multiple audio viewpoints. Each viewpoint includes a description of the unit metrical position, harmonic, timbral and dynamic categories. The larger the number of categories in each descriptor, the larger the set of σ symbols. The user-defined parameterization in `earGram 2.0` avoids the formalization of subjective and contextual factors and prior assumptions on the audio content. The interactive navigable visualizations detailed in Section 4 aim to guide the user in defining

the descriptors parameterization. Yet, the FO arc visualization in `earGram 2.0`, of which a screenshot is shown in Fig. 6, provides the ultimate guide to the user. An optimal number of links results in a balanced model across the audio source complexity while capturing repetitions. A low number of links corresponds to a high amount of novelty and therefore a larger degree of variation and surprise. Conversely, a high number of links corresponds to a high degree of redundancy. Ultimately, the auditory feedback from traversing the FO links, detailed in Section 6, exposes the perceptual 'smoothness' of the automaton.

As shown in [31, 20], a balanced FO structure highlights structural boundaries across the audio source temporal morphology, by splitting the timeline across the x-axis into areas of dense repetition. These ought to have a high degree of perceptual agreement typical of a formal section, instigated by changes of instrumentation, harmony or other surface elements. In Fig. 6, a binary AB structure is highlighted in both representations.

6. GENERATIVE MUSIC APPLICATIONS

Drawing from the hierarchical descriptive layers in `earGram 2.0`, this section details generative music applications that re-purpose the audio source *sonic* and *formal* structure. They aim to expose the deployment potential of the minimal hierarchical audio descriptor scheme in interfacing with musical audio in a similar fashion to the manipulation of music manifested as symbolic notation. Use-cases for re-purposing audio at the multiple sound object, meso and macro time scales of music are detailed.

At the sound object level, `earGram 2.0` includes navigable interfaces in 2D spatial corpus visualizations, as shown in Fig. 3, in the style of CataRT [25]. Beyond the typical timbre space navigation found in CSS applications, a harmonic space, where key relations, as those expressed in the circle of fifths, and functional categories are expressed as distances. The smaller the distance across units, the greater their degree of perceptual affinity [6]. In navigating these interfaces, one can create unit sequences at various density rates, from highly overlapping *clouds* of audio units to isolated units with fine control over their affinity. Continuous position overlay and discrete pointer locations can be used as input instruction from user interface devices (e.g., mouse or joystick).

Mid-level categorical descriptions can explore sequencer-based interfaces with multiple degrees of automation. Euclidean rhythms [28] are adopted to sequence up to nine layers, each linked to a different timbre cluster. Three user-defined parameters control the generation: the total number of steps per cycle and, for each layer, the number of hits per cycle and a step offset value. Fig. 7 shows the interface of the Euclidean step sequencer in `earGram 2.0`, which provides a visual feedback of the resulting sequences by geometric representations of their active states as overlapping polygons.

At the long-term structure description, FO is adopted as a strategy to re-create unit sequences that guarantee some degree of sonic and formal structural retainment by traversing the FO links. In Algorithm 1, two long- and short-term forces contribute to the sequence generation. The user defines the long-term form as meso or marco structural section across the audio source duration in the FO arc visualization structure. A unit range $R = [r_{min}, r_{max}]$ to be adopted during generation is set by defining the orange bar size and start point in Fig. 6. The short-term surface sequence results from traversing the suffix, $SP(i)$ and reverse suffix $rSP(i)$ links in the FO, which promote smooth looping

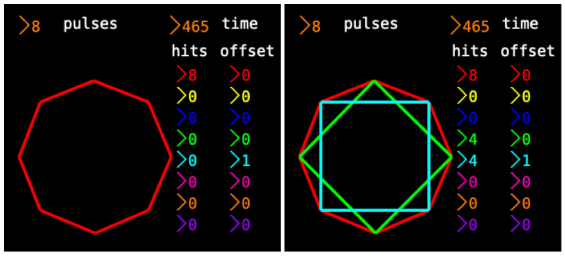

Figure 7: Interface of the Euclidean rhythms sequencer in `earGram 2.0`. Within a cycle of eight steps, left image has a single active layer with 8 steps, right image has three active layers with varying number of steps and offsets.

points across the temporal audio source. Algorithm 1 defines the steps of the iterative algorithm. At runtime, based on a preceding state i a random probabilistic decision which imposes different continuations paths that traverse the FO structure. traverse the links in FO are traversed according to a simple probabilistic decision which decides whether to move of rules, which mostly define the degree o units are selected

Algorithm 1: Function `onlineTransverseFO`

Require: Oracle P in active state i; a user-defined continuation parameter $q \in [0, 1]$ and a unit range R
set a list $l = \varnothing$
Generate a random value $u \in [0, 1]$
if $u < q$ **then**
  $i \leftarrow i + 1$
else
  while $SP(i) \geq 0$ **do**
    $l \leftarrow l \cup \{SP(i)\}$
    $l \leftarrow l \cup \{rSP(i)\}$
    $i \leftarrow SP(i)$
  end while
  $l \cap R$ ▷ exclude from l values outside R
  Randomly retrieve a value u from l
  $i \leftarrow u + 1$
  $v \leftarrow u$
end if
return Audio unit to be synthesized v

To explicitly observe and test the generative re-purposing applications in `earGram 2.0`, in particular, those detailed in this section, please refer to the supplementary material available online at `https://sites.google.com/site/eargram/` along with the software source code, tutorials and artistic work.

7. CONCLUSIONS AND FUTURE WORK

The paper presents `earGram 2.0`, a software for the hierarchical description of sound corpora based on salient attributes of music notation towards the fluid manipulation of audio units for the creative re-purposing of audio sources while retaining sonic and formal structure. Contributions include a minimal audio description scheme which targets musically trained users by focusing on primary dimensions of music notation. Descriptors have double representation in continuous and discrete spaces. The latter results from user-defined parameters guided by intuitive visualizations, thus promoting a highly contextual framework with minimal assumptions from the audio source.

In future, to assess the creative potential of `earGram 2.0`, hands-on workshop sessions and tests with different com-

munities of techno-fluent musicians are planned. Particular effort will be placed in learning optimal parameter ranges particular applications scenarios, such as game background music and sonic textures, active listening or even through-provoking variation creation within virtual assistant composers. Finally, a comparison with related methods, notably including the information rate measure in AO is under study.

8. ACKNOWLEDGMENTS

Research funded by the project "Experimentation in music in Portuguese culture: History, contexts and practices in the 20th and 21st centuries" (POCI-01-0145-FEDER-031380) co-funded by the European Union through the Operational Program Competitiveness and Internationalization, in its ERDF component, and by national funds, through the Portuguese Foundation for Science and Technology.

9. REFERENCES

[1] C. Allauzen, M. Crochemore, and M. Raffinot. Factor oracle: A new structure for pattern matching. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 295–310. Springer, 1999.

[2] E. Amiot. *Music through Fourier space*. Springer, 2016.

[3] T. H. Andersen. In the mixxx: Novel digital dj interfaces. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pages 1136–1137. ACM, 2005.

[4] G. Assayag and G. Bloch. Navigating the oracle: A heuristic approach. In *International Computer Music Conference*, pages 405–412, 2007.

[5] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov. Omax brothers: A dynamic topology of agents for improvization learning. In *Workshop on Audio and Music Computing Multimedia*, pages 125–132. ACM, 2006.

[6] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. Davies. A multi-level tonal interval space for modelling pitch relatedness and musical consonance. *Journal of New Music Research*, 45(4):281–294, 2016.

[7] G. Bernardes, M. E. Davies, and C. Guedes. A hierarchical harmonic mixing method. In *International Symposium on Computer Music Multidisciplinary Research*, pages 151–170. Springer, 2017.

[8] G. Bernardes, C. Guedes, and B. Pennycook. Eargram: an application for interactive exploration of large databases of audio snippets for creative purposes. In *International Symposium on Computer Music Modelling and Retrieval*, pages 265–277, 2012.

[9] W. Brent. A timbre analysis and classification toolkit for pure data. In *International Computer Music Conference*, pages 224–229, 2010.

[10] W. Brent. A perceptually based onset detector for real-time and offline audio parsing. In *International Computer Music Conference*, 2011.

[11] M. Caetano and X. Rodet. Sound morphing by feature interpolation. In *International Conference on Acoustics, Speech and Signal Processing*, pages 161–164. IEEE, 2011.

[12] J. Cohen. Information theory and music. *Behavioral Science*, 7(2):137–163, 1962.

[13] D. Conklin. Music generation from statistical models. In *AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.

[14] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[15] S. Dixon. An interactive beat tracking and visualisation system. In *International Computer Music Conference*, 2001.

[16] M. Goto. Active music listening interfaces based on signal processing. In *International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages IV–1441. IEEE, 2007.

[17] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Research*, 9(11):1106–1115, 1999.

[18] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium*, pages 9–12, 2000.

[19] N. Lee. *Digital Da Vinci: Computers in Music*. Springer Science & Business Media, 2014.

[20] B. Lévy, G. Bloch, and G. Assayag. Omaxist dialectics: Capturing, visualizing and expanding improvisations. In *International Conference on New Interfaces for Musical Expression*, pages 137–140, 2012.

[21] P. Molina, M. Haro, and S. Jordá. Beatjockey: A new tool for enhancing dj skills. In *International Conference on New Interfaces for Musical Expression*, pages 288–291, 2011.

[22] G. Nierhaus. *Algorithmic composition: paradigms of automated music generation*. Springer Science & Business Media, 2009.

[23] F. Pachet and J.-J. Aucouturier. Improving timbre similarity: How high is the sky. *Journal of Negative Results in Speech and Audio Sciences*, 1(1):1–13, 2004.

[24] I. Quinn. General equal-tempered harmony: parts 2 and 3. *Perspectives of New Music*, pages 4–63, 2007.

[25] D. Schwarz. Corpus-based concatenative synthesis. *Signal Processing Magazine*, 24(2):92–104, 2007.

[26] D. Schwarz and B. Hackbarth. Navigating variation: composing for audio mosaicing. In *International Computer Music Conference*, pages 604–607, 2012.

[27] K. Siedenburg, I. Fujinaga, and S. McAdams. A comparison of approaches to timbre descriptors in music information retrieval and music psychology. *Journal of New Music Research*, 45(1):27–41, 2016.

[28] G. Toussaint. The euclidean algorithm generates traditional musical rhythms. In *BRIDGES: Mathematical Connections in Art, Music and Science*, pages 47–56, 2005.

[29] D. Tymoczko. Set-class similarity, voice leading, and the fourier transform. *Journal of Music Theory*, 52(2):251–272, 2008.

[30] D. Tymoczko and J. Yust. Fourier phase and pitch-class sum. In *International Conference on Mathematics and Computation in Music*, pages 46–58, 2019.

[31] C.-i. Wang and G. J. Mysore. Structural segmentation with the variable markov oracle and boundary adjustment. In *International Conference on Acoustics, Speech and Signal Processing*, pages 291–295. IEEE, 2016.

[32] J. Yust. Stylistic information in pitch-class distributions. *Journal of New Music Research*, 48(3):217–231, 2019.