ALLOY SOUNDS: NON-REPEATING SOUND TEXTURES WITH PROBABILISTIC CELLULAR AUTOMATA

Tiago Fernandes Tavares *

School of Electrical and Computer Engineering
University of Campinas
Campinas, Brazil
tavares@dca.fee.unicamp.br

Thales Roel P. Pessanha

Interdisciplinary Nucleus for Sound Studies
University of Campinas
Campinas, Brazil

Gustavo Nishihara

Interdisciplinary Nucleus for Sound Studies
University of Campinas
Campinas, Brazil

Guilherme Zanchetta L. Avila

Interdisciplinary Nucleus for Sound Studies
University of Campinas
Campinas, Brazil

ABSTRACT

Contemporary musicians commonly face the challenge of finding new, characteristic sounds that can make their compositions more distinct. They often resort to computers and algorithms, which can significantly aid in creative processes by generating unexpected material in controlled probabilistic processes. In particular, algorithms that present emergent behaviors, like genetic algorithms and cellular automata, have fostered a broad diversity of musical explorations. This article proposes an original technique for the computer-assisted creation and manipulation of sound textures. The technique uses Probabilistic Cellular Automata, which are yet seldom explored in the music domain, to blend two audio tracks into a third, different one. The proposed blending process works by dividing the source tracks into frequency bands and then associating each of the automaton's cell to a frequency band. Only one source, chosen by the cell's state, is active within each band. The resulting track has a non-repeating textural pattern that follows the changes in the Cellular Automata. This blending process allows the musician to choose the original material and the blend granularity, significantly changing the resulting blends. We demonstrate how to use the proposed blending process in sound design and its application in experimental and popular music.

1. INTRODUCTION

Creating diversity in music composition has largely been aided by algorithms throughout history. Algorithms have been used since the musical dice games from the XVIII Century [1], and were more formally explored using modern language concepts in work by Xenakis [2] and John Cage [3]. Digital computers have allowed implementing algorithms that run in real-time, which has lead to several modern applications [4, 5, 6].

Although many of these applications work on symbolic music representations [4, 5], it is undeniable that musicianship in the 20th and 21st centuries has progressively made sound qualities more important [7]. In this process, music-making is becoming deeply intertwined with detailed elaborations on timbre and extraordinary discoveries in instrumental techniques and orchestration. In turn, composers and music producers now face the challenge of finding meaningful paths towards new, diverse, characteristic sonorities to use in their work.

Likewise, many computer-based applications can aid musicians to manipulate sound qualities and timbres [8, 9]. Many of these algorithms rely on modelling specific sound sources such as rainfall [10], harmonic instruments [11], and percussive impacts [12]. These algorithms are used in applications such as automatic music mastering and mixing, automatic orchestration, and finding specific configurations for software synthesizers, and can be evaluated according to the similarity between their outputs and particular, desired targets.

Computers can also be used to generate entirely new timbres. In this type of application, the automatically-generated novelties can bring forward new, interesting perspectives for musical exploration. In early electronic music, these explorations have relied on low-frequency oscillators, statistical models, and chaotic series; more recently, they have been aided by modern data-driven models such as Convolutional Neural Networks [13, 14] and Adversarial Auto-Encoders [8].

In all of these cases, automatic sound novelty creation is related to a trade-off between the need to generate unexpected sound qualities and the need to control the results of the creation process. In these applications, the interaction between musicians and machines lies in a spectrum that goes from having no control over the creation process (the machine generates a new musical idea upon request) to having full control (the musician simply plays an entirely predictable instrument). The exploration of this trade-off often leads to systems in which the musician has control over some parameters related to their probabilistic behaviors, thus only controlling the emerging statistical properties of the final outcome, like in the well-known granular synthesis technique [15]. The search for more possibilities in this matter has inspired the use of population-based metaheuristics for computer-assisted music creation.

Population-based metaheuristics are algorithms that work by modeling local interactions between individual agents. Different behaviors emerge from these interactions [16]. This idea was implemented, among others, in models for genetic evolution [17],

* Thanks CNPq (Brazilian National Council for Research) for funding this research.

swarms of insects [18], ant colonies [19], and cell communication [20].

The models for cells are known as *cellular automata* (CA) [20]. They consist of discrete-time models in which each cell is in a single state at each time. The state of the cell in the following discrete-time depends on its state and the state of the neighboring cells [20].

CA are deterministic, and their behavior can be entirely modified by changing the cell state transition rules. Two interesting possible emerging results can be chaos, which allows its use in cryptography, or self-organization, which was the basis for the famous Game of Life. These emerging behaviors are hard to predict from a simple analysis of the rulesets, and have been studied using exhaustive simulation in different starting conditions [21].

CA is also a widespread process in the artistic field of Generative Art, which consists of art practices that use systems whose autonomy generates emerging behavior and contributes to the resulting work [22]. A common process in Generative Arts is to devise systems and then search for their emerging behaviors, for example in the artwork "Rule 30", by Myskja (2008), in which a machine punches holes on a sheet of paper using Cellular Automata [23]. These behaviors have been first investigated in the context of avant-garde art, but are currently being adopted by mainstream production software as methods to generate variety. They have been used in the music domain to map triggers to particular cells [24, 25], to create note or beat sequences [26], and to control filterbanks [27, 28, 29] and other audio transformation tools.

A variation of the deterministic CA is the probabilistic CA (PCA). In the PCA, cell state transitions are probabilities that depend on the cell's and its neighbors' states. PCAs can facilitate to predict the spread of fires [30] and that of diseases [31], among other applications. Up to the authors' knowledge, PCAs have not yet been used in the music domain. PCA as a probabilistic process generates a level of indeterminacy desirable in the artistic and creative field and unlike their deterministic counterparts, they allow results based on the control of different probabilistic parameters therefore producing more complex behaviors. In this article, we explore PCAs for audio transformation. We use the metaphor of blending two source sounds the same way that two minerals mix themselves to form marble, or metals form an alloy. The result is a family of sounds called Alloy Sounds, which are new, but still carry original material characteristics.

The proposed algorithm generates novel musical qualities based on blending previously-recorded sound sources. It allows the musician to define the original sound sources, the probabilistic blending rules, and the PCA's operation rate. The generated Alloy Sounds continuously change in time, but their transformation patterns are controlled by the musician.

The sound blending process relies on a PCA whose cells are organized in a ring neighborhood. The cells can assume states either 1 or 0. Each cell is related to a specific frequency band, and the automaton evolves in real-time intervals. As further explained in Section 2, the cell state information is used to select which of the original sources is active at each frequency band and at each time interval. This algorithm allows to create sounds that differ from the original material, but, as we discuss in Section 3, can be purposefully designed to retain some specific characteristics from the sources. Moreover, the retained characteristics can chance through time, hence increasing the variability of the sound results.

Importantly, PCAs can be seen as an extension of deterministic CAs, hence they provide a larger field for exploration. In special, we note that there are some patterns that cannot be achieved with CAs because of their deterministic nature. Among these patterns, as discussed in Section 2.3, we highlight changes that appear sporadically and long-term behaviors that rely on probabilistic state durations.

The idea of preserving different characteristics of two audio sources has been previously explored in cross-synthesis [32]. This technique consists of analyzing two audio samples to extract parameters and then synthesizing a new sample using a combination of these parameters. Differently, Alloy Sounds preserve characteristics of the original material in the form of parts of their spectral shapes [33], which appear along time following a probabilistic behavior.

Another similar idea is called timbre morphing [34, 35, 36, 37, 38, 39, 40], which aims at generating signals that sound like a perceptual interpolation between the sources. This perceptual interpolation has been shown to correlate with the interpolation of audio features, in special the spectral centroid, the spectral spread, the spectral kurtosis, and the spectral skewness [36]. The Alloy Sounds discussed in this article do not aim to provide perceptually linear interpolations; instead, they aim to generate a continuously and non-repeating transformation of the source material that preserves some nuances from the original sounds.

This article brings forward a novel method to generate new sounds. The resulting Alloy Sounds allow recognizing the original material, but the transformations are also evident. Therefore, the proposed method can be used to reach unexplored perspectives within sound design and foster creative processes in contemporary musicianship.

2. PROPOSED METHOD

The method proposed in this article aims at combining two different source signals into a resulting blend. For such, the source sounds are divided into equal frequency bands. The combination process uses the PCA to choose which of the sources sound in each frequency band in the final blend.

The PCAs evolve in time, hence changing which of the source sounds play in each frequency band at each time. This makes the sources' spectral contents blend in an ever-changing fashion, following patterns that depend on the PCA's transition probability rules.

Importantly, PCAs can produce more complex behavior than the well-known Markov Chains. This is because the transitions in Markov Chains are only dependent of the automaton's own state, whereas, in the PCA, transition probabilities also depend on the state of the neighboring cells. This allows each cell of a PCA to interact with the others, which cannot happen in traditional Markov Chain models.

This section is further subdivided into two subsections. Subsection 2.1 describes the PCA algorithm used in this work. After that, Subsection 2.2 shows how the PCA is used to perform the blending process.

2.1. Probabilistic cellular automaton

The cellular automaton implemented in this work has K cells c_k. They are disposed in a ring neighborhood, that is, cell c_k has the left neighbor c_{k-1} and the right neighbor c_{k+1}. The first and the last cells are also neighbor to each other, thus closing the "ring" neighborhood.

The state $s_{k,t}$ of each cell is re-estimated at each discrete time t. For such, it uses a probability distribution that depends on the states of the cell and its neighbors, that is, $P(s_{k,t+1} = 1) = f(s_{k-1,t}, s_{k,t}, s_{k+1,t})$. The probability distribution for the eight state possibilities related to a cell and its neighbors is manually configured.

This allows generating a grid $\boldsymbol{S}_{K,T}$, with the states of K cells through T discrete time steps, using the procedure:

1. Generate K random cells for the first generation;
2. Update each cell in the next time step using its state probability $P(s_{k,t} = 1)$;
3. Advance one time step;
4. If T generations have been created, stop;
5. Return to step 2.

This process allows creating different state grids even when the next state probabilities remain the same. Figure 1 illustrates this situation. Grids A and B were generated using the same PCA configuration. Likewise, grids C and D were generated using another configuration. It is possible to see that the types of patterns that appear in grids A and B are similar, whereas different patterns appear in grids C and D. However, none of the grids is strictly equal to each other. Hence we show that the PCA configuration allows changing the types of patterns in the state grids but do not imply generating entirely predictable behavior.

Next, we discuss how to use the grids $\boldsymbol{S}_{K,T}$ to blend two audio recordings.

2.2. Blending process

The blending process takes two audio tracks as inputs. They are adjusted to the same length by zero-padding the shorter one. Then, their Short-Time Fourier Transforms (STFT) are calculated using a user-defined frame size, a 50% overlap ratio and zero-padding to twice the frame's length. This process results in two STFT representations:

$$\mathbf{S_1}, \mathbf{S_2} \in \mathbb{C}^{NxM}, \tag{1}$$

where N is the STFT length and M is its total number of frames.

After that, the PCA grid $\boldsymbol{S_{K,T}}$ calculated in Section 2.1 is scaled to the dimensions of S_1 and S_2. This generates a mask $\boldsymbol{A}_{N,M}$, whose values $a_{n,m}$ are defined by:

$$a_{n,m} = s_{i,j} \tag{2}$$

where

$$i = \lfloor (n\frac{K}{N}) \rfloor \tag{3}$$

and

$$j = \lfloor (m\frac{T}{M}) \rfloor. \tag{4}$$

In Equations 3 and 4, the *floor* operation $\lfloor x \rfloor$ means "the largest integer lower than x".

Then, the mask is low-pass filtered along the time axis (that is, each row of $\boldsymbol{A}$ is filtered separately) to smooth the transitions between the source sounds, as shown in Figure 2. The filter can be configured by the musician: a higher cutoff frequency leads to more abrupt transitions, whereas a lower cutoff frequency leads to smoother transitions. Then, the blend is performed using the mask $\boldsymbol{A}$ to weight each of the sources in each frequency band at each time frame, generating the blend $\mathbf{S_3}$ by:

$$\mathbf{S_3} = \mathbf{S_2} \odot \mathbf{A} + \mathbf{S_1} \odot (1 - \mathbf{A}), \tag{5}$$

where the symbol $\odot$ is the element-wise multiplication.

Last, the blend $\mathbf{S_3}$ is converted to the time domain using overlap-and-add.

Figure 3 illustrates this process. It shows how the PCA grids are used as a mask to alternate between each of the sources. The spectrogram of the result retains characteristics from both the sources and the PCA used to create it.

In the next section, we show how to use our blending process in sound design applications.

2.3. Devising PCA rules

Probabilistic Cellular Automata present emergent behavior that is entirely defined by eight probability values, each related to the possible states of the cell and its neighbors. An exhaustive search in this parameter space is unfeasible, as it can span billions of possibilities.

One possibility to make rule sets is to start from deterministic cellular automata rule sets, which knowingly generate behaviors such as stability, oscillation, chaos, or self-organization [21]. By changing the deterministic rules to probabilistic ones, we can generate PCAs that behave similarly to the corresponding CAs, with the advantage of adding more variety to the transitions. Table 1 shows an example of this process starting from Rule 110 and adding uncertainty to all transitions, which results in behaviors such as the ones shown in Figure 4.

Also, some interesting behaviors can be obtained by carefully adjusting the PCA's rules. It is beyond the scope of this work to provide a complete catalogue of behaviors. Instead, we provide some examples and their underlying rationale, so that they can be used as a starting point to further explorations in this matter. The rules related to all patterns discussed here are shown in Table 2.

We start from simple case, namely "spots". This pattern, shown in Figure 5 (a), does not allow for any continuous black spots, because the zero probability related to all previous states in the form $\{0, 1\}1\{0, 1\}$ forcibly return cells to the 0 (white) state.

Increasing the probability related to the $\{0, 1\}1\{0, 1\}$ states, we can make the spots last longer. This leads to another pattern, namely "stripes", which consists of independent horizontal stripes. As shown in Figure 5 (b), adjacent stripes can only start simultaneously, that is, a new stripe cannot appear next to another existing one.

The probabilities associated with states 001, 100, and 101 control the possibility of stripes becoming wider, as if they spread to neighbor cells over time. This creates another pattern, shown in Figure 5 (c), which we call "stains".

As noted before, the possibilities shown in this section are not exhaustive. Nevertheless, they can be an useful starting point to create new patterns and pattern variations. In the next section we discuss the musical application related to designing sound blends.

3. DESIGNING SOUND BLENDS

Sounds commonly relate to perceptual characteristics, such as their pitch, timbre, and intensity. Sound mixing often causes conflicts in these characteristics, which is partly due to spectral overlap. Our technique aims to avoid this spectral overlap while also preserving spectral characteristics of both blended sounds. As a consequence, the sound design process uses musical characteristics

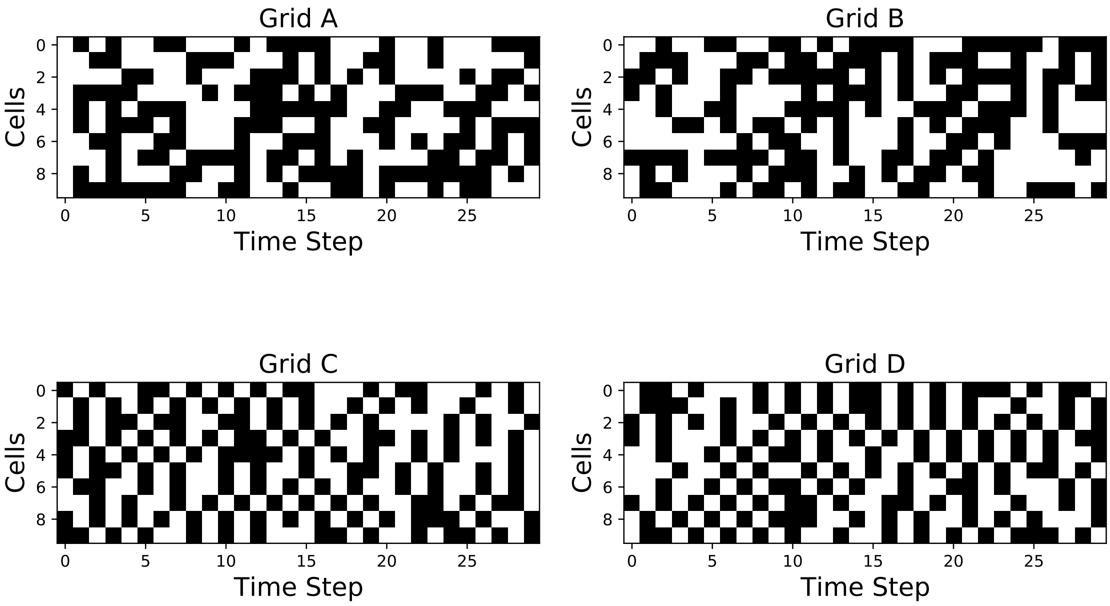

Figure 1: Example grids of the cellular automaton states through discrete-time. Grids A and B were generated using $f(s_{k-1,t}, s_{k,t}, s_{k+1,t}) = [0.5, 0.45, 0.8, 0.7, 0.45, 0.75, 0.2, 0.2]$, and the grids C and D were generated using $f(s_{k-1,t}, s_{k,t}, s_{k+1,t}) = [0.7, 0.9, 0.1, 0.2, 0.7, 0.9, 0.1, 0.2]$. The PCAs have a ring neighborhood, that is, the upper and lower cells are neighbors.

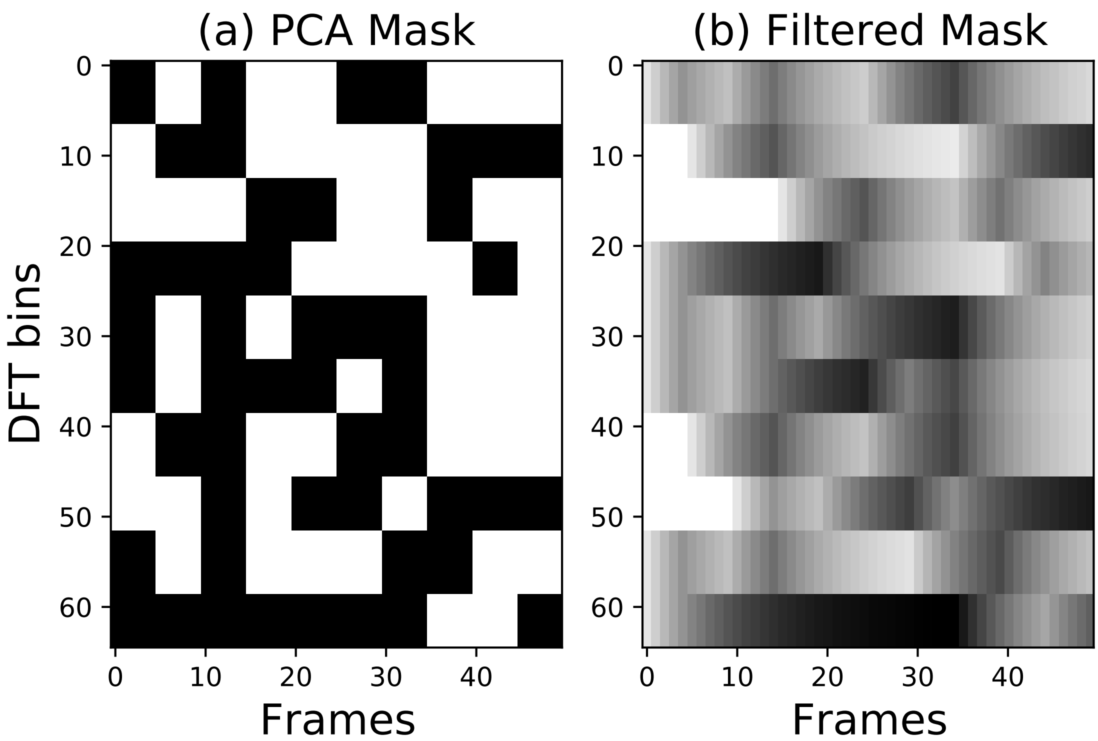

Figure 2: Mask resized to spectrogram dimensions (left) and low-pass filtered (right).

Previous States ($s_{k-1,t}, s_{k,t}, s_{k+1,t}$)	000	001	010	011	100	101	110	111
Rule 110 (deterministic)	0	1	1	0	1	1	1	0
Relaxed Rule 110	0.3	0.7	0.7	0.3	0.7	0.7	0.7	0.3

Table 1: PCA rules ($P(s_{k,t+1} = 1)$) related to the relaxed Rule 110.

Previous States ($s_{k-1,t}, s_{k,t}, s_{k+1,t}$)	000	001	010	011	100	101	110	111
Spots	0.1	0	0	0	0	0	0	0
Stripes	0.1	0	0.5	0.9	0	0	0.8	0.9
Stains	0.1	0.1	0.5	0.9	0.1	0.5	0.8	0.9

Table 2: PCA rules ($P(s_{k,t+1} = 1)$) for each of the patterns discussed in this section.

from the original material, but also adds its own behavior, as we will discuss in the next subsections[1].

3.1. Blending concrete sounds

The idea of combining concrete sounds has been greatly explored in early electronic music (the 1920s and 1930s), for example, in the *musique concrète* lead by composers like Pierre Schaeffer and Pierre Henry. Here, we show how our technique helps achieving semantically meaningful results to explore combinations between two concrete sounds coherently and consistently. For such, we experimented with blending two distinct sounds, as shown in Figure 6.

The first sound was retrieved from the BBC Sound Effects Library [41]. As shown in Figure 6 (a), its spectrum indicates that it is a harmonic sound with strong attacks, and it has harmonics close to 1kHz and 2kHz. The second sound is a recording of an audio amplifier's inactivity noise. It consists of colored noise with a more prominent range from 1.5kHz to 2kHz, as shown in Figure 6 (b).

We used the Relaxed Rule 110 probabilities shown in Table 1. Henceforth, the blend map has structures similar to those shown in Figure 4. The sound blend, shown in Figure 6 (c), preserves characteristics of amplifier noise, but there are some occurrences of bird sounds through the whole spectrum. This new sound can be used to foster the idea of a soundscape.

3.2. Alloy Chords

Another idea is to blend pitched sounds with different timbres and add diversity to its original timbre. The resulting blend is similar to a chord with a characteristic timbre that changes along time, and music producers can use it as a *pad* background sound.

To demonstrate this possibility, we used two samples. The first one consists of a cello playing a C2 note (64Hz). As shown in Figure 7 (a), its spectrum highlights its richness of partials, which is typical of the cello's sound in its low register. The second sound consists of a flute playing a C4 note (262 Hz). Its spectrogram, shown in Figure 7 (b), highlights its strong fundamental.

We used the Stains pattern probabilities shown in Table 2. Hence, the blend map has structures similar to those shown in Figure 5 (c).

The result's spectrogram is shown in Figure 7 (c). We notice that combining two sounds generates a more complex sound, whose timbre has a richness of partials. Also, we can see that the harmonic balance changes through time. This behavior indicates a constantly-evolving timbre that is hard to obtain using traditional linear filtering and mixing techniques.

This example shows how to create Alloy Sounds with interesting timbres from simple note samples. The newly generated sounds depend on the original samples and the musician-defined probability transition rules. This interplay between the randomness and the musician's definitions leads to many possibilities that can be explored and creatively used.

3.3. Musical Usage

The use cases discussed in Section 3 regard changing blending parameters as to generate novel sounds. These sounds can be further used in musical contexts. Next, we discuss two possible musical applications.

The blending process that generates Alloy Sounds transforms sound qualities, and this can be used to generate audio texture variety. As such, it can be used to create soundscapes with variations of concrete sounds. Henceforth, Alloy Sounds can be used to build specific musical discourses that refer to those of electroacoustics and musique concrète.

Another musical application regards popular electronic music. Nowadays, it is common to use low-volume sounds, called "pad"s, to make a mix sound more full. Alloy Sounds can be used as pads in electronic music, so that the sense of filling is preserved and the result is non-repetitive.

3.4. Further uses in sound design

The possibilities discussed in this article are not exhaustive, and other variations can be obtained by changing the algorithm's parameters and the source materials.

The probability distribution for the next state can be changed to help to generate different textures. These can be more granular, tend to draw diagonal lines, favor one state over the other, and other possibilities that can be cleverly developed. Impressive results can come from linearly interpolating two grids along time.

The number of cells is related to the number of bands in the frequency domain. A very high number of bands can lead to harder-to-perceive blends, whereas a meager number can harm the frequency blending characteristic.

The number of steps in the grid is related to the PCA's execution rate. Hence, the number of time steps can be adjusted to favor the desired texture blending.

[1] Audio examples using Alloy Sounds can be found in our repository: `https://github.com/tiagoft/alloy_sounds`

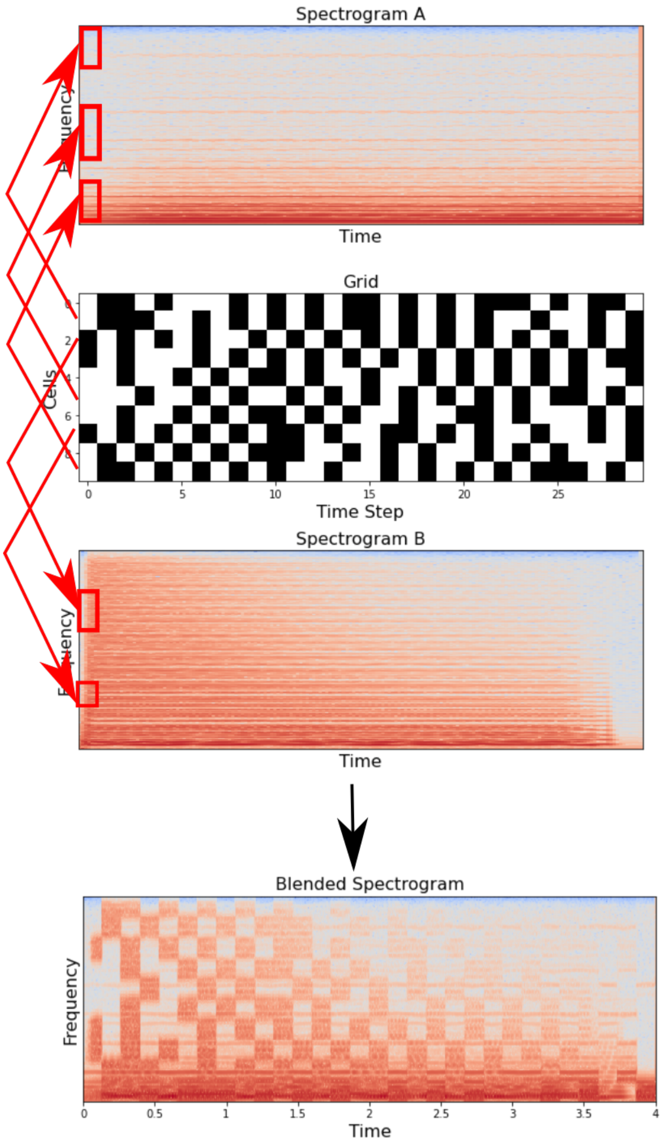

Figure 3: Blending process of two spectrograms with cellular automaton and output spectrogram. The original sources are two drone-style sounds with different spectral contents, chosen as to generate a clear visual contrast in the blend's spectrogram. The white cells and black cells represent the spectrograms A and B, respectively. On the blended spectrogram the segments of both input are discernible.

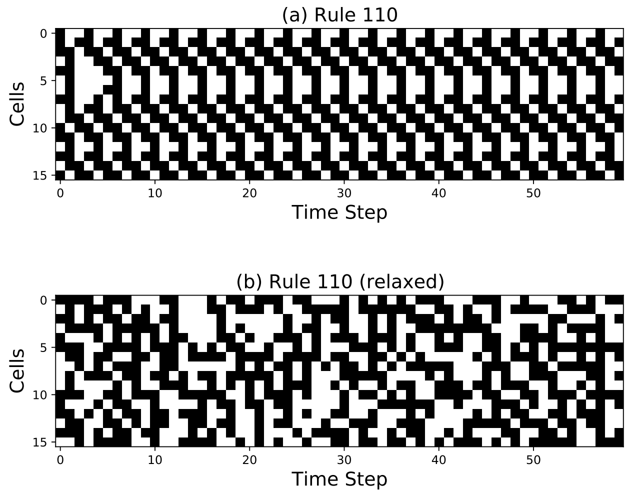

Figure 4: Grids generated using Rule 110 (a) and a PCA based on similar rules (b), as shown in Table 1. The PCA exhibits patterns that resemble its deterministic counterpart and increase variability.

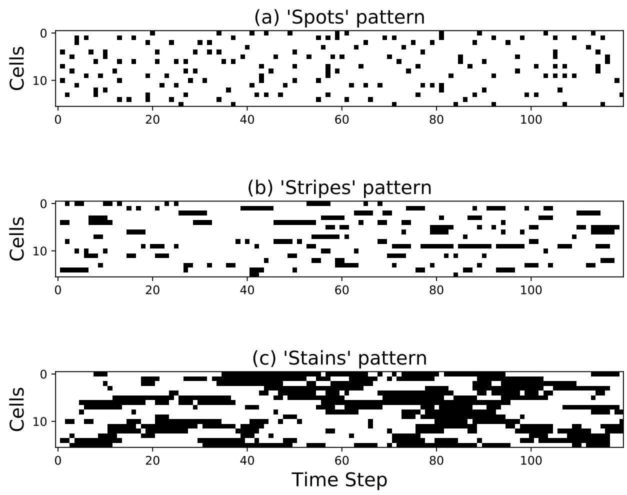

Figure 5: PCA grids created using each one of the patterns shown in Table 2.

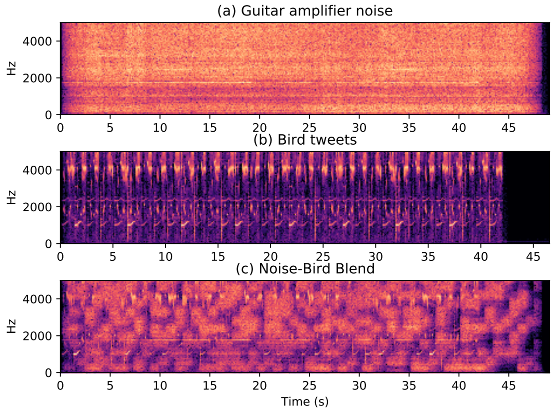

Figure 6: Spectrogram showing the blend of concrete sounds. The resulting spectrogram has characteristics of each of the sources, which evolve along time. The spectrograms were limited in band to improve readability.

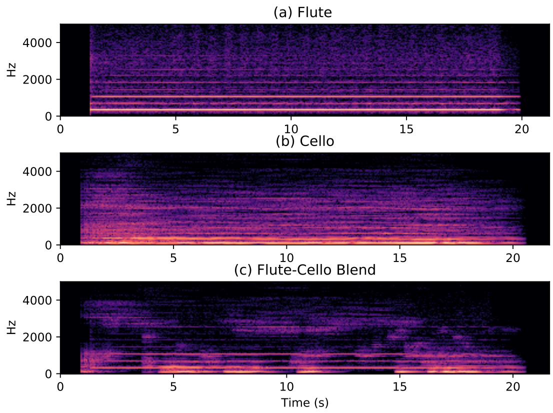

Figure 7: Blending chords. The resulting sound preserver the harmonic characteristic, but its quality (timbre) changes through time. The spectrograms were limited in band to improve readability.

Finally, the cutoff frequency of the low-pass filter must be configured. Lower cutoff frequencies cause the grid to be too smooth and can make the blending process imperceptible. On the other extreme, a very high cutoff can generate undesired clicks or attacks in the result. All the parameters are strongly dependent on the source material and the desired results. Henceforth, the Alloy Sounds are a rich domain for exploring different blends and their characteristics and can be an important tool for semantic-inspired sound design and other forms of generative art.

4. CONCLUSION

In this article, we discuss the use of Probabilistic Cellular Automata (PCAs) to create sound blends. The proposed blending process generates a family of sounds that is called Alloy Sounds in reference to the blends of materials that combine, creating textures such as marble. The proposed sound blends use PCAs to generate a similar behavior in the audio domain. Because of its exploratory purpose, the proposed method cannot be evaluated solely according to their results. Rather, it is presented as a creativity tool that favors both novel results and a different perspective for the creative process.

The proposed blending process uses a PCA to determine which of the original sounds is more prominent in each frequency and time interval. The estimated prominence evolves along time guided by the PCA rules creating an ever-evolving pattern of transformations. The musician can control the source materials of this blend and the PCA granularity in the time and frequency domains.

Alloy Sounds can be a starting point for a walk through a large domain of synthetic spectra. It offers a whole family of unusual sounds – limit sounds, paradoxical sounds, unstable sounds, complex sounds – that are hard to obtain using traditional techniques. We show experiments related to merging concrete sounds and to merging harmonically-related notes, and both of them allowed creating diverse and evolving sound qualities.

The proposed method enables us to test a myriad of semantically-meaningful possibilities to explore the combination between sounds and, as a consequence, to generate novel materials that allow composers to create a coherent and consistent musical discourse in sound-based music. For this reason, the convergence between Alloy Sounds and the composer's critical positioning about the generated results can be an important tool for creative explorations. Also, we speculate whether logarithmic-scaled or overlapping bands could be more effective in creating more natural-sounding results. These questions will be left for future work.

5. REFERENCES

[1] Gerhard Nierhaus, *Algorithmic composition: paradigms of automated music generation*, Springer, Wien ; New York, 2009, OCLC: ocn233933272.

[2] Makis Solomos, "Cellular automata in xenakis's music. theory and practice," 2005.

[3] John Cage, *Silence : lectures and writings*, chapter Composition as Process I: Changes, pp. 18–34, Wesleyan University Press, 1951.

[4] Jean-Pierre Briot and François Pachet, "Music Generation by Deep Learning - Challenges and Directions," *Neural Computing and Applications*, vol. 32, no. 4, pp. 981–993, Feb. 2020, arXiv: 1712.04371.

[5] Omar Lopez-Rincon, Oleg Starostenko, and Gerardo Ayala-San Martín, "Algoritmic music composition based on artificial intelligence: A survey," in *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, Feb. 2018, pp. 187–193, ISSN: 2474-9044.

[6] IEEE MultiMedia, "Playing With Virtual Musicians: The Continuator in Practice," *IEEE MultiMedia*, vol. 9, no. 3, pp. 77–82, July 2002.

[7] Makis Solomos, *De la musique au son. L'émergence du son dans la musique des XXe-XXIe siècles*, Presses Universitaires de Rennes, 2013.

[8] Adrien Bitton, Philippe Esling, Antoine Caillon, and Martin Fouilleul, "Assisted Sound Sample Generation with Musical

Conditioning in Adversarial Auto-Encoders," *Digital Audio Effects (DAFx) 2019*, June 2019, arXiv: 1904.06215.

[9] Shiguang Liu and Dinesh Manocha, "Sound synthesis, propagation, and rendering: A survey," 2020.

[10] Shiguang Liu, Haonan Cheng, and Yiying Tong, "Physically-based statistical simulation of rain sound," *ACM Trans. Graph.*, vol. 38, no. 4, July 2019.

[11] Zhimin Ren, Hengchin Yeh, and Ming C. Lin, "Example-guided physically based modal sound synthesis," *ACM Trans. Graph.*, vol. 32, no. 1, Feb. 2013.

[12] Changxi Zheng and Doug L. James, "Toward high-quality modal contact sound," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, vol. 30, no. 4, Aug. 2011.

[13] Jenelle Feather and Josh H. McDermott, "Auditory texture synthesis from task-optimized convolutional neural networks," in *2018 Conference on Cognitive Computational Neuroscience*, Philadelphia, Pennsylvania, USA, 2018, Cognitive Computational Neuroscience.

[14] Hugo Caracalla and Axel Roebel, "Sound texture synthesis using convolutional neural networks," *Digital Audio Effects (DAFx) 2019*, May 2019, arXiv: 1905.03637.

[15] Curtis Roads, "Introduction to Granular Synthesis," *Computer Music Journal*, vol. 12, no. 2, pp. 11, 1988.

[16] Christian Blum and Andrea Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, Sept. 2003.

[17] David E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Pub. Co, Reading, Mass, 1989.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Nov. 1995, vol. 4, pp. 1942–1948 vol.4, ISSN: null.

[19] Nicolas Monmarché and Frédéric Guinand, Eds., *Artificial ants: from collective intelligence to real-life optimization and beyond*, ISTE, London, 2010, OCLC: 699811270.

[20] Stephen Wolfram, "Universality and complexity in cellular automata," *Physica D: Nonlinear Phenomena*, vol. 10, no. 1-2, pp. 1–35, 1984.

[21] Andrew Ilachinski, *Cellular Automata: A Discrete Universe*, World Scientific, Singapore, 2001.

[22] Philip Galanter, "What is generative art? complexity theory as a context for art theory," in *In GA2003–6th Generative Art Conference*. Citeseer, 2003.

[23] Alan Dorin, Jonathan McCabe, Jon McCormack, Gordon Monro, and Mitchell Whitelaw, "A framework for understanding generative art," *Digital Creativity*, vol. 23, no. 3-4, pp. 239–259, 2012.

[24] Kenneth McAlpine, Eduardo Miranda, and Stuart Hoggar, "Making Music with Algorithms: A Case-Study System," *Computer Music Journal*, vol. 23, no. 2, pp. 19–30, June 1999, Conference Name: Computer Music Journal.

[25] Dale Millen, "Cellular Automata Music," in *Proceedings of the ICMC*, 1990.

[26] Peter Beyls, "Cellular Automata Mapping Procedures," in *Proceedings of the ICMC*, 2004.

[27] Eduardo Reck Miranda, "Cellular Automata Music: An Interdisciplinary Project," *Interface*, vol. 22, no. 1, pp. 3–21, Jan. 1993.

[28] Eduardo Miranda, "Cellular Automata Synthesis of Acoustic Particles," in *Proceedings of the ICMC*, 1995.

[29] Scott Mclaughlin and Pierre Alexandra Tremblay, "Spectral-conway: Cellular Automata Off The Grid," in *Proceedings of the ICMC*, 2010.

[30] J. Quartieri, Nikos Mastorakis, Gerardo Iannone, and Claudio Guarnaccia, "A cellular automata model for fire spreading prediction," 07 2010.

[31] S. White, Ángel Rey, and Gerardo Sánchez, "Using cellular automata to simulate epidemic diseases," *Applied Mathematical Sciences*, vol. 3, pp. 959–968, 01 2009.

[32] Juan José Burred, "Cross-synthesis Based on spectrogram Factorization," in *ICMC*, 2013.

[33] Denis Smalley, "Spectromorphology: explaining sound-shapes," *Org. Sound*, vol. 2, no. 2, pp. 107–126, Aug. 1997.

[34] Marcelo Freitas Caetano and Xavier Rodet, "AUTOMATIC TIMBRAL MORPHING OF MUSICAL INSTRUMENT SOUNDS BY HIGH-LEVEL DESCRIPTORS," in *International Computer Music Conference*, United States, June 2010, pp. 11–21.

[35] Marcelo F. Caetano and Xavier Rodet, "Evolutionary Spectral Envelope Morphing by Spectral Shape Descriptors," in *Proceedings of DAFx*, 2010.

[36] M. Caetano and X. Rodet, "Musical instrument sound morphing guided by perceptually motivated features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1666–1675, 2013.

[37] Duncan Williams, Peter Randall-Page, and Eduardo Miranda, "Timbre morphing: near real-time hybrid synthesis in a musical installation," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. June 2014, pp. 435–438, Zenodo.

[38] Marcelo Caetano and Xavier Rodet, "Sound morphing by feature interpolation," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 161–164, IEEE.

[39] Trevor Henderson and Justin Solomon, "Audio Transport: A Generalized Portamento via Optimal Transport," in *Proceedings of DAFx*, 2019, arXiv: 1906.06763.

[40] G. Roma, O. Green, and P. A. Tremplay, "Audio Morphing Using Matrix Decomposition and Optimal Transport," in *Proceedings of DAFx*, 2020.

[41] Sound Ideas, "The bbc sound effects library-original series," *Retrieved March*, vol. 1, pp. 2012, 2012.