CELLULAR AUTOMATA DYNAMIC CONTROL FOR SOUND DESIGN WITH HISTOGRAM MAPPING SYNTHESIS AND THE MULTITYPE VOTER MODEL

Jaime Serquera and Eduardo R. Miranda

ICCMR - Interdisciplinary Centre for Computer Music Research
University of Plymouth, UK
{jaime.serquera, eduardo.miranda}@plymouth.ac.uk

ABSTRACT

This paper presents a mechanism for controlling over time an automaton with sound design purposes. The process is developed in the context of Histogram Mapping Synthesis and it is based on a DSP monitoring of the automaton's evolution. Such dynamic control is suitable to design sounds with dynamic spectrum and controlled complexity. We present a new textural concept based on dynamic and complex sound beats.

1. INTRODUCTION

Cellular automata (CA) are simple mathematical / computational models that can exhibit complex behaviour. CA are normally implemented as a regular grid of cells in one or more dimensions. Each cell may assume any state from a finite set of *n* values. CA evolve in successive generations at every time unit. For each generation, the values of all cells change simultaneously according to a set of transition rules that takes into account the states of the neighbouring cells. The transition rules can be deterministic or non-deterministic. With a deterministic rule, for a given configuration of cell states, the updated cell state is always the same. With non-deterministic rules the next state is not only dependent on the neighbourhood but also on some random inputs and/or probabilistic components. A probabilistic rule gives the probabilities that each cell will or will not transition to the next possible state. The states of the cells may represent different colours and therefore, the functioning of a two-dimensional cellular automaton may be displayed on the computer screen as a sequence of images, like an animated film.

Cellular automata have been of interest to computer musicians [1] because of their emergent structures – patterns not created by a single rule but through the interaction of multiple units with relatively simple rules. This dynamic process leading to some order allows the musician to explore new forms of organization. In sound synthesis, CA are normally used for controlling over time the parameters of a synthesis instrument. Many of the synthesis techniques demand enormous amounts of control data for obtaining interesting results, making it difficult to be controlled manually. CA represent a solution to this problem because with few parameter specifications it is obtained massive amounts of structured data. The goal is to transfer the structured evolution of CA onto the sound synthesis domain. This is always done through a mapping, a set of correspondences between different domains.

The aim of our research is to develop a sound synthesis technique based on CA capable of allowing a sound design process. The major criticism against sound synthesis techniques based on CA concerns the sound design limitations experienced by the users, which fundamentally stem from the unpredictable and autonomous nature of these computational models. In order to achieve our goal we have developed a new approach which considers the output of CA as digital signals and uses DSP procedures in order to analyse them. This approach opens a great variety of possibilities to better understand the self-organization process of CA with the view of identifying mapping possibilities for sound synthesis and control possibilities for sound design. Histogram Mapping Synthesis (HMS) is a fruit of such an approach.

In this paper we present a control mechanism for sound design with HMS and the multitype voter model.

2. HISTOGRAM MAPPING SYNTHESIS

HMS is a recently new sound synthesis technique based on a statistical analysis of CA evolutions. The functioning of a two-dimensional automaton is considered as a sequence of digital images, each of which is analysed by means of histogram measurements. Such a DSP analysis gives a histogram sequence that can be displayed in a 3D plot.

The histogram of a grey level digital image is a graphical representation of the number of occurrences of each grey level[1] in the image. By dividing the number of occurrences by the total number of pixels of the image, the histogram is normalised and expressed in probabilistic terms giving an estimate of the probability of occurrence of each grey level in the image –the sum of all the histogram bins is equal to one.

From certain appropriate automatons, in the histogram sequences, which can be seen as temporal structures, it is possible to identify time-varying

[1] Apart from this definition, in this paper we refer to colours instead of grey levels because we usually display the CA in the computer screen using a palette of different colours.

structural elements resembling spectral components of sounds such as sinusoidal components, noise components, transients, spectral envelopes, formants, etc. With these structural elements we can design the time-varying frequency content of sounds; we can build spectrograms.

Seen from another perspective, these structural elements can be used as control data to drive different synthesis techniques. Depending on the resemblance of the structural elements, different mappings onto appropriate synthesis parameters can be established. These possible mappings are distinctive because in most other synthesis techniques based on CA there is not an intuitive correspondence between the components of the automaton and the components of a sound.

HMS is also remarkable due to its controllability for sound design, as it is the topic of this paper. Although most of the HMS control possibilities depend on the automaton being used, an important aspect that can be generalised is the possibility of developing a sound design process from the structural elements of the histogram sequences. Virtually infinite sounds can be designed from a single histogram sequence by manipulating its structural elements. The fact of being operating in the frequency domain facilitates the control over all the sound attributes: intensity, duration, pitch and timbre.

3. THE MULTITYPE VOTER MODEL

In 1953, geneticist Kimura introduced the stepping stone model. This process was studied extensively by other geneticists over twenty years before being rediscovered by probability theorists Clifford and Sudbury in 1973 being named the invasion process and by Holley and Liggett in 1975 under the name of the voter model [2].

The voter model is interpreted as a model of opinion formation. A collection of individuals is defined, ‘each of which has one of two possible positions on a political issue. These possible positions are denoted by 0 and 1. Periodically, […] an individual reassesses his view in a rather simple way: he chooses a “friend” at random with certain probabilities and then adopts his position’ [4]. It can be also seen as a model of competition. The interpretation is clear from the point of view of the invasion process; two species compete for the territory by invading each other’s sites. When the voter model is generalized to more than two opinions it is known as multitype voter model (MVM).

The MVM can be simulated by means of a probabilistic cellular automaton in two dimensions. We have implemented the following transition rule on the basis of the stepping stone implementation found at [3]: At every time step, a coin with prescribed Update Probability of tails is tossed for each cell in the grid. If the coin comes up heads, the state of the given cell will be replaced by the state of one of its neighbours selected uniformly at random from the specified neighbourhood. The coin-tossing process can be implemented by defining such a prescribed Update Probability as a CA input parameter. It will hold a number chosen by the user between 0 and 1. Then, the coin will come up heads if a random number between 0 and 1 generated for each cell and time step is higher than the Update Probability.

From a uniform random distribution of cell values, or colours, as the initial configuration, the automaton self-organizes in clusters or areas of single colours (Figure 1). As the rule is iterated, some areas will increase their space while others will decrease to the extent that they can disappear. In the end, one colour will prevail over the others when, according to the voter interpretation, consensus occurs.

The random inputs and probabilities in the rule make that different runs with the same settings result in different evolutions.

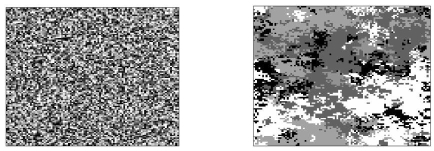

Figure 1. Two configurations of a MVM evolution. From a random input (left) it self-organizes in coloured areas (right).

4. FEATURES OF THE MULTITYPE VOTER MODEL HISTOGRAM SEQUENCES

The histogram sequences of the MVM present interesting features that make them suitable for sound synthesis. Here we summarise some of them; for a dedicated study, see [5]. Figure 2 shows the histogram sequence of an automaton with 70x70 size, 20 colours, and Update Probability = 0.5 through 4000 iterations.

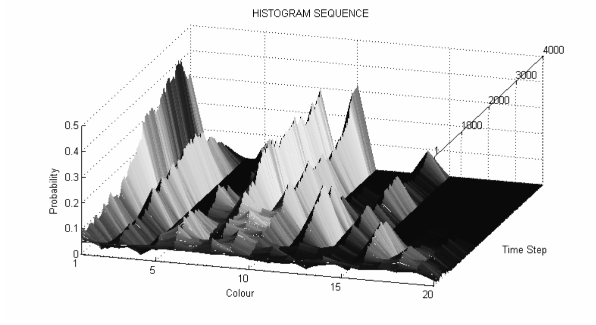

Figure. 2. A histogram sequence of a MVM evolution.

The first important characteristic is that the bins of the histogram sequence may represent the time-varying amplitudes of sound partials[2]. In addition, note that the multitype voter model allows us to work with as many colours (i.e., partials) as we want.

Secondly, note that since the automaton has a finite size and all the cells are occupied, when the total area covered by one colour increases then it means that the areas of other colours have decreased. In the histogram

[2] Assuming this premise, for the rest of the paper we may use the term “partials” for referring to histogram bins.

sequence it means that when some partials grow, other partials decrease. This is appealing for the synthesis of long sound textures with interesting internal opposite movements. The long duration is achieved by a large interpolation and it is desirable in order to synthesise the amplitude movements slowly enough so as to be perceptually noticeable. The problem is that the textures can not be endless because of the extinction of colours. Moreover, close to the end, when just few partials remain, the spectral complexity of the textures could be no longer interesting.

In other respects, the disappearance of colours during the run of the automaton is attractive for the synthesis of note type of sounds because it parallels the behaviour of sounds produced by acoustic instruments; they usually produce more partials in the attack than in the rest of the sound. In Figure 2 the automaton goes from having 20 colours to 5 in nearly 4000 time steps. We can favour this phenomenon if we work with a smaller automaton. According to the invasion interpretation, the twenty species would compete for less territory, a fact that would provoke a sooner extinction of many species. From another point of view, in such a smaller automaton there will initially be less exemplars per species (i.e., cells per colour), a fact that propitiates their extinction. The problem now is that the automaton can achieve consensus very soon after a quick disappearance of most of the partials, i.e., there can be too much extinction not under control. This usually leads to an inconsistent structure with very few partials for the sustain of the sound (Figure 3).

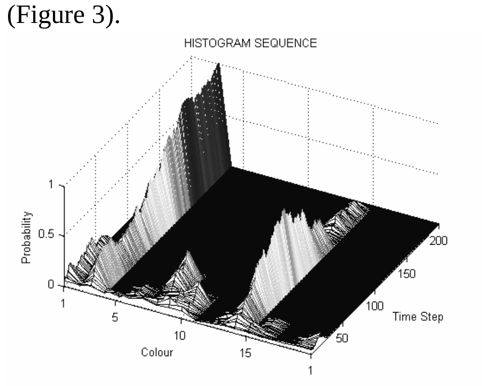

Figure 3. Histogram sequence from an automaton with the same settings as for Figure 2 except for the size which has been reduced to 10x10.

In summary, the MVM is interesting for producing dynamic spectra but we need, for both, note type of sounds and sound textures, a control mechanism for enabling the coexistence of partials when desired. In this respect, in [5] we developed a control mechanism based on the relationship between the size and the number of colours. When extinction was desired we used a small automaton (relatively to the number of colours) and when coexistence was desired we enlarged its size. The problem of this approach is that working with large sizes is not computationally efficient and the coexistence, although propitiated, is not totally guaranteed.

5. SELF-REGULATED MODEL

We have devised a control mechanism for enabling coexistence being largely independent of the CA size. It is based on an extended version of the MVM and a DSP monitoring of its evolution within a feedback algorithm. Originally, in the MVM the Update Probability (UP) is assigned equally to all the cells. The extension of the MVM, which was developed in [6] in order to design damped sounds, consists on the assignment of colour-dependent UPs. From the point of view of the invasion process, that means having different species with different degrees of vulnerability. For instance, a colour with a low UP has a high probability of “being eaten” by a neighbour, whereas a colour with UP=1 represents an invincible species.

This control over the vulnerability can be applied to prevent the extinction of species; when a species were in danger of extinction, we would protect it by assigning a high UP. Therefore, in order to enable coexistence the automaton must be monitored and the UP colour dependence must be variable in time. There are different ways to implement this paradigm. In this research we have established that the UP of each colour will be equal to 1 minus the value of its histogram bin at every time step. With this, the new model self-regulates in such a way that the more a partial decreases, the less vulnerable becomes while, at the same time, the more any other partial grows, the more vulnerable becomes.

The self-regulation process can be activated when desired, for instance, in the sustain of a sound in order to guarantee the permanence of a number of partials. To illustrate this possibility we will work with the same automaton settings as for Figure 3. Then, when there remain a determined number of partials (determined in advance for the sustain), the automaton will automatically enter self-regulation mode (Figure 4).

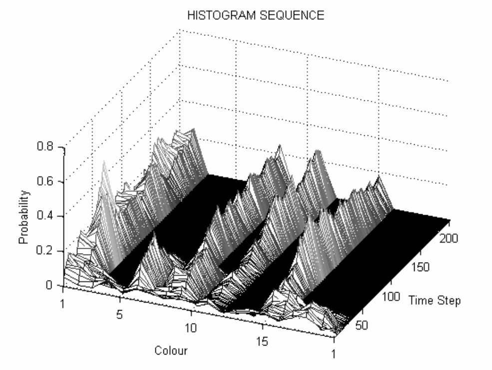

Figure 4. Self-regulation automatically activated when remaining five partials for the sustain.

As a result we obtain a transient attack in which many partials fade out. Thereon, we obtain a sustain where the permanence of the remaining partials is ensured. We can see that during self-regulation, the permanent partials undergo a process of equalization. In this respect, if desired, we can always modify the overall relative amplitudes by manually scaling any partial before rendering the sound.

The self-regulated model is also suitable for the synthesis of endless dynamic textures. We have focused our experiments on the textural effects of sound beats. Earlier experiments in this line were carried out with histograms sequences similar to Figure 2. We synthesised an interesting type of texture by extracting a number of the permanent partials, typically four, and assigning to them slightly different frequencies. The constant changes in the relative amplitudes of the four beating partials made the interference patterns change over time continuously, gradually and non-deterministically. We obtained dynamic complex beats – complex in the sense of using more that two beat frequencies. These beats are perceived as a bumpy texture in contrast to the invariable rhythmic sequence that common synthetic beats produce. To the best of our knowledge this textural concept is novel.

There were, however, a series of problems that hindered its practical use. Firstly, the overall amplitude was not constant. That is because although the sum of all the histogram bins is equal to one at every moment, we only extracted four out of many other partials. Also, as we have seen before, due to the disappearance of colours the texture could not be endless. Finally, real-time was not possible because we could not know in advance which would be the partials that would remain.

With the self-regulation algorithm activated from the very beginning, so as to ensure that all the partials can coexist, we have been able to produce this kind of textures with constant overall amplitude equal to one, which then can be modified at will by the user. Endless duration and real-time are potentially possible. In order to produce notable amplitude fluctuations for each beating partial we have been working with very small CA (Figure 5) –with small sizes each cell contributes a high weight to the histograms which, consequently, vary more from one generation to the other.

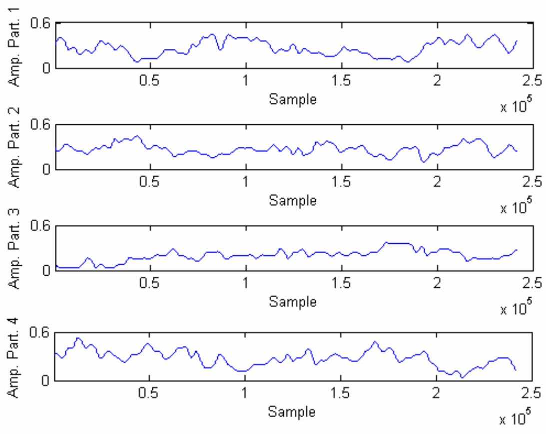

Figure 5. Time-varying amplitudes for four beating partials obtained from a 5x5 automaton. Note that the sum of the four amplitude envelopes is constant and equal to one. They are the result of a large interpolation of 100 time steps so as to produce a 30s. sound at Fs=8000Hz.

With such small sizes the system can be pushed to the limit and partials can exceptionally become extinct. We have included a mutation process to solve this problem. At every time step it is checked whether a partial has become extinct and if it is the case, one cell at random would change its colour to that of the extinct species. This mutation process would not normally work in the original MVM where just one species exemplar would be unlikely to survive. But in the self-regulated model such an exemplar will have a very high UP, so it will be virtually invincible.

6. CONCLUSIONS AND FURTHER WORK

We have controlled over time a cellular automaton for sound design practice with Histogram Mapping Synthesis. The basis of our method lies in a DSP monitoring of the automaton which informs us on when and how to alter its autonomous evolution.

Among the outcomes of this research we highlight a new textural concept based on dynamic complex beats. Sound examples can be found at:
http://www.youtube.com/watch?v=XADlXSP5TkQ

We are planning to further explore the textural effects of such beats. The first milestone would be to run our system it in real time, i.e., to develop a "dynamic beating oscillator". We are also considering interactivity enhancements by which some of the parameters of the system could be controlled by the user. Finally, we believe that our method could be also useful for improving chorus algorithms.

7. REFERENCES

[1] Burraston, D. and Edmonds, E. "Cellular Automata in Generative Electronic Music and Sonic Art : Historical and Technical Review", *Digital Creativity*, 2005.

[2] Cox, J. T. and Durrett, R. "The Stepping Stone Model: New Formulas Expose Old Myths", *The Annals of Applied Probability*, 2002.

[3] Fisch, R. and Griffeath, D. *WinCA: A Cellular Automaton Modeling Environment*. Available at http://psoup.math.wisc.edu/ftp/pub/winca_10.exe from www.archive.org. Version 1.0, 1996.

[4] Liggett, T. M. *Interacting Particle Systems*. Springer-Verlag, New York, 1985.

[5] Serquera, J. and Miranda, E. R. "Evolutionary Sound Synthesis: Rendering Spectrograms from Cellular Automata Histograms", *In Proceedings of EvoMUSART, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 2010.

[6] Serquera, J. and Miranda, E. R. "Cellular Automata Sound Synthesis with an Extended Version of the Multitype Voter Model", *In Proceedings of the 128th AES Convention*, London, 2010.