Evolutionary Sound Synthesis: Rendering Spectrograms from Cellular Automata Histograms

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. In this paper we report on the synthesis of sounds using cellular automata, specifically the multitype voter model. The mapping process adopted is based on digital signal processing analysis of automata evolutions and consists in mapping histograms onto spectrograms. The main problem of cellular automata is the difficulty of control and, consequently, sound synthesis methods based on these computational models normally present a high factor of randomness in the output. We have achieved a significant degree of control as to predict the type of sounds that we can obtain. We are able to develop a flexible sound design process with emphasis on the possibility of controlling over time the spectrum complexity.

Keywords: Sound Synthesis, Cellular Automata, Histogram Mapping Synthesis, Additive Synthesis, Multitype Voter Model.

1 Introduction

A number of musicians, in particular composers, have started to turn to evolutionary computing for inspiration and working methodology. This paper focuses on cellular automata (CA), a class of evolutionary algorithms widely used to model systems that change some feature with time. They are suitable for modelling dynamic systems in which space and time are discrete, and quantities take on a finite set of discrete values. CA are highly suitable for modelling sound and music, which both are fundamentally time-based and can be thought of as systems in which a finite set of discrete values (e.g., amplitudes, frequencies, musical notes, rhythms, etc.) evolve in time [10].

In the 1950s several different kinds of systems equivalent to CA were independently introduced. The best-known way in which CA were introduced (and which eventually led to their name) was through work by John von Neumann in trying to develop an abstract model of self-reproduction in biology –a topic which had emerged from investigations into Cybernetics [13].

Cellular automata 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

C. Di Chio et al. (Eds.): EvoApplications 2010, Part II, LNCS 6025, pp. 381–390, 2010.

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 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 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.

There have been different mapping attempts [1] ranging from direct assignments of CA values, like in Lasy [2], to higher-level approaches intending to map the overall CA behaviour, like in Chaosynth [9]. We are interested in the second type of approach. Our research strategy is based on the analysis of CA evolutions by means of digital signal processing techniques in order to discover structural information of their organization. Then we proceed with the mapping of the analysis results onto appropriate synthesis parameters.

This paper is organized as follows. In Section 2 we present the automaton chosen for this study, which is based on the multitype voter model. In section 3 we explain the mapping process adopted, which is based on CA analysis with the histogram technique and, aims at the design of sound spectrograms. In section 4 and 5 we describe the musical features revealed from the histogram analysis of the multitype voter model and, we suggest solutions for the drawbacks found. As the control is the main problem of CA, in section 6 we comment controllability aspects of the sound design process. Sounds are rendered using additive synthesis. Finally, Section 7 concludes this paper.

2 The Multitype Voter Model

The automaton chosen for this research is based on the voter model. In 1953, geneticist Kimura introduced the stepping stone model [7]. This process was studied extensively by other geneticists over twenty years before being rediscovered by probability theorists Clifford and Sudbury in 1973 [3] where it was called the invasion process and by Holley and Liggett in 1975 [6] under the name the voter model [4]. Nowadays, the voter model is considered one of the standard models of interacting particle systems [8].

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 opinions on a political issue. These possible opinions are denoted by 0 and 1. An individual reassesses his view in a rather simple way: he chooses a "friend" at random with certain probabilities and then adopts his opinion [8]. When the voter model is generalized to more than two opinions it is known as multitype voter model.

It can be also seen as a model of competition. The interpretation is clear from the point of view of the invasion process; different species compete for the territory and the result of conflict is the invasion by one of the species of territory held by the other.

These models can be simulated by means of a probabilistic cellular automaton in two dimensions. The multitype voter model can be implemented with the following transition rule: a number between 0 and 1 is chosen as to be the update probability for all cells. Then, for each cell in the grid, a random number between 0 and 1 is generated at every time step. If the random number generated for the given cell is higher than the update probability, then the state of the cell changes to that of one of its neighbours selected uniformly at random. (Neighbour is defined as the four orthogonally adjacent cells: north, east, south, west) [5].

From a uniform random distribution of cell values, or colours, as the initial configuration, the automaton self-organizes in 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.

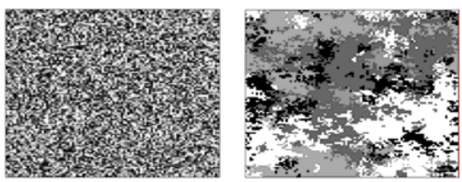

Fig. 1. Two configurations of a multitype voter model evolution. From a random input (left) it self-organizes in coloured areas (right).

3 Mapping Process: From Histograms to Spectrograms

The mapping process adopted in this study is based on a statistical analysis of the CA evolution. The functioning of a two-dimensional automaton is considered as a sequence of digital images and it is analysed by histogram measurements of every CA image. Such a CA analysis gives a histogram sequence.

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 expressed in

[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.

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.

The mapping process is very simple; the bins of the histogram sequence are considered to be bins of a spectrogram. With an appropriate automaton, in the histogram sequence it is possible to find structural elements resembling spectral components of a sound. For example, from a histogram analysis of the hodge podge automaton there were discovered structural elements similar to sinusoidal components and others similar to noise components such as noise bands and transients [12]. This makes such mapping process distinctive; in most other cases there is not an intuitive correspondence between the components of the automaton and the components of a sound.

With these structural elements we can design the time varying frequency content of a sound; we can build a spectrogram. This spectrogram can be rendered into sound using different synthesis techniques –the structural elements of the histogram sequences become control data for the synthesis program.

4 Features of the Multitype Voter Model Histogram Sequences

We have found several music-like features that make the histogram sequences of the multitype voter model interesting. Figure 2 shows the histogram sequence of an automaton with 100x100 size and 20 colours through 12000 iterations.

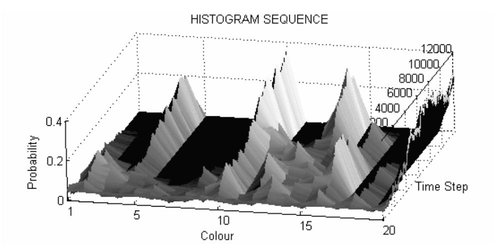

Fig. 2. A histogram sequence of the multitype voter model

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, the disappearance of colours during the run attracts our attention. The sounds of acoustic instruments present a similar behaviour; they usually produce more partials in the attack than in the rest of the sound. In the previous example the automaton goes from having 20 colours to 4 in 12000 time steps. We can favour this phenomenon if we work with more colours in a smaller automaton. According to the invasion interpretation, there would be more species competing for less territory, a fact that will provoke a sooner extinction of many species just at the beginning of the run. Figure 3 shows how, in a 30x30 automaton with 50 colours, there is a disappearance of many colours in only 200 time steps.

---

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

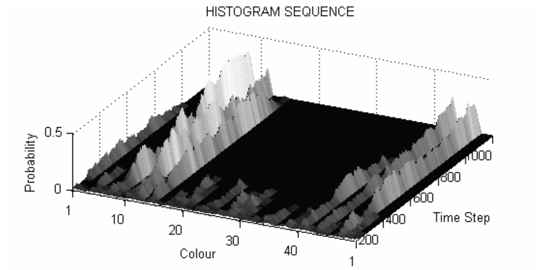

Fig. 3. Inducing disappearance of partials at the beginning of the CA run

The problem now is that the automaton can achieve consensus very soon after a quick disappearance of most of the partials. With this, there will not be an interesting structure for the sustain of the sound. We have devised a solution to this problem. When there remain a determined number of partials (determined in advance) to constitute the sustain structure, the automaton is automatically replicated several times in order to build a bigger one (which will have the same histogram). With more exemplars per species and, what is more important, having provided more space, the remained species can coexist for longer (Figure 4).

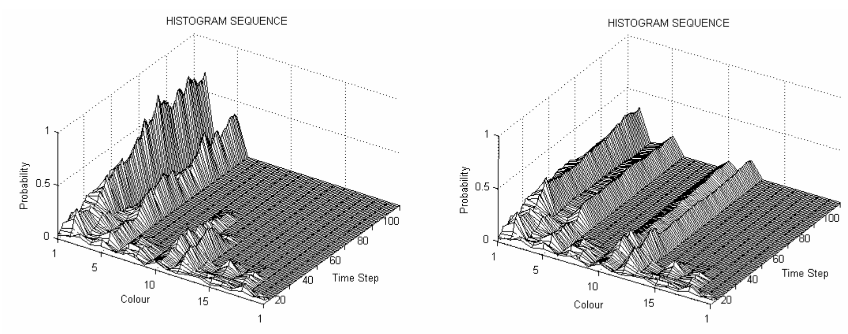

Fig. 4. Controlling spectrum complexity by controlling extinction and coexistence. Histogram sequence of a 10x10 automaton (left). Same evolution with replication in the 30^{th} generation to build a 40x40 automaton (right).

Finally, 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 sequence it means that when some partials grow, other partials decrease (Figure 5). This is of interest because it reminds us of opposite movements typically fond in polyphonic music. When some voices in the background become important they go to the foreground increasing their intensity, while at the same time, previous foreground voices become less important and go to the background decreasing their intensity.

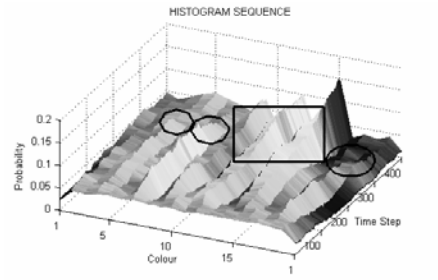

Fig. 5. Opposite movements. The circles show decreasing partials and the square shows increasing partials at the same moment.

Because of the characteristics described above, we find the multitype voter model suitable for rendering single sounds and also polyphonic sound textures.

We still identify two drawbacks. Firstly, before the automaton reaches consensus there will be an increasing prominence of one colour over the others, which may be not desirable. Normally, before this happens we will have enough structure for rendering a sound. Otherwise, as we will see later, it will be possible to exclude a prominent partial or modify its amplitude before rendering the sound.

The second drawback lies in the fact that the histogram sequences do not provide patterns of sound attacks and releases. We treat this matter in the following section.

5 Attacks and Releases

In Figure 6 we can see that the histogram sequence does not start from zero and therefore the synthesized sound does not have an attack pattern. It is also clear that the automaton does not provide either release patterns for all the partials.

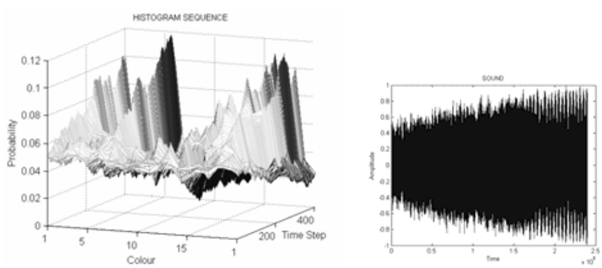

Fig. 6. No attack/release patterns in the histogram sequence (left), neither in the sound (right)

We can always apply external envelopes for creating attacks and releases. But we are interested in giving the automaton complete control over the time varying amplitudes and we have developed a solution by extending the model.

We start the automaton with just one instance of each colour, placed at random locations. With this we guarantee a beginning of the histogram sequence that can be approximated to zero. We define empty cells for the rest of the automaton. The functioning is the same (the same rule), but in order to ensure a growth (it could be a growth model) we impose that occupied cells can not become empty. With this, the occupied cells will expand covering the whole automaton (Figure 7). At that point the attack is finished.

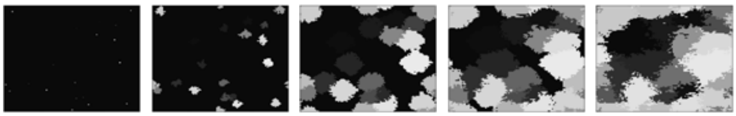

Fig. 7. CA evolution for creating attack envelopes

In Figure 8 we can see, in the histogram sequence, the beginning from zero and the attack patterns. The synthesized sound has as a result a sigmoidal-like attack.

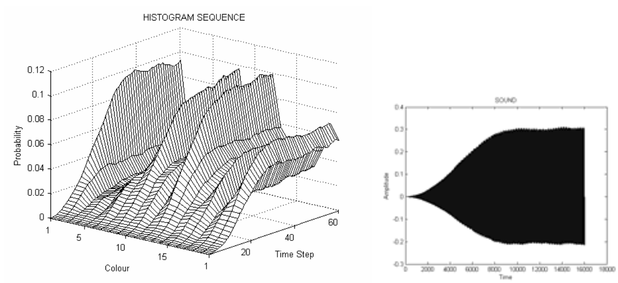

Fig. 8. Attack pattern in the histogram sequence (left), and in the sound (right)

An interesting characteristic of these attacks is that each partial reaches a different amplitude value. This is because the initial colours are located at random locations, and therefore they start to compete for the space at different times (the competition for the space occurs when different colours collide and has as a result a decrease of their growth).

Also, working with relatively big CA we observed that with this solution, partials present more stable amplitudes. This is probably so because the CA start with already established areas of colours, and then during the run, it is more difficult for established areas to experience changes in their sizes. We find this behaviour very interesting for sound synthesis, hence the reason we attempted to capture it in our system.

We have devised an alternative solution that, starting from single dots, and regardless of the amount of them, creates increasing areas but not with specific colours, but with random colours. With this, we fill the automaton with a random distribution of colours.

In order to create releases we have devised a method based on the opposite idea. We introduce sources of “epidemics” at random locations, which will expand “killing” all the cells. The curves obtained look like sigmoidals, with different release times for each partial (due mostly to the random locations of the epidemics), having as a result that the strongest partial is not necessarily the last one that disappears (Figure 9).

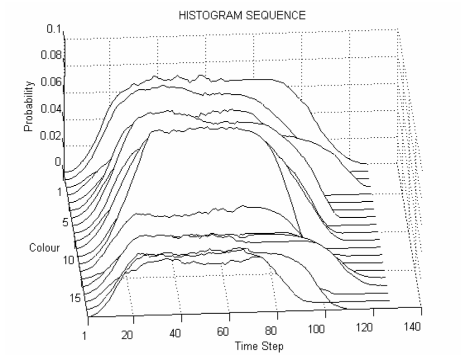

Fig. 9. Histogram sequence of a 100x100 automaton, with attacks and releases. Note that there is not disappearance of colours in the attack due to the relatively big size of the automaton.

In terms of implementation, the empty cells of the attacks and the dead cells of the releases may correspond to negative cell values not considered in the computation of the histograms.

6 Control

The multitype voter model is controlled with four input parameters: the size, the number of colours, the initial configuration, and the update probability. However, the predictability of the outcome of CA is an open problem; it is not possible to predict the value that a specific cell would hold after a number of generations [14]. This is even more obvious if we have random inputs and probabilities in the rules. Therefore, although a level of unpredictability is accepted and often desired in systems for generating music and sound, being under unpredictability conditions implies limited controllability. A lack of a reasonable level of control restricts the music or sound design process [11]. Our work alleviates this limitation in many respects.

Firstly, it is possible to find direct and intuitive relations between the multitype voter model input parameters values and their effects in the histogram sequence. For example, in section 4 we have seen how to cause the disappearance of partials and enable coexistence by controlling the relationship between the size and the number of colours. It is also clear that the update probability controls the amount of cell-colour updates that occur at each generation. Therefore, it controls the rate at which the automaton and thus, the histogram sequences, evolve towards consensus. The initial configurations provide control over the attacks. For example, starting with more than one instance per colour we can make the attack structure shorter –the same idea can be applied to the releases. With different number of instances of each colour we can control, to a certain extent due to the random evolutions, the relative amplitude of the partials (Figure 10).

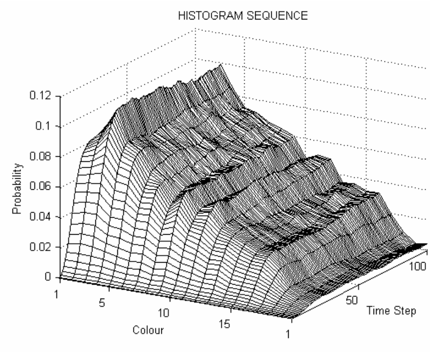

Fig. 10. Controlling relative amplitudes. The number of initial instances of each colour was inversely proportional to the colour value.

We can also model attack delays of the partials by introducing the different colours at different generations (Figure 11).

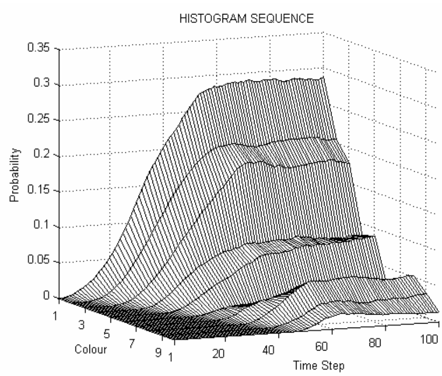

Fig. 11. Controlling spectrum complexity in the attack. Successive colours appear with a delay of 5 CA generations.

Finally, another important aspect of controllability is the possibility of developing a sound design process from the structural elements of the histogram sequences. Conceptually, the first steps to be performed are the assignment of frequencies to the partials and, the specification of the sound duration. From here, other spectral transformations are possible. Time stretching, pitch shifting and amplitude modifications of each partial are straightforward to implement. With all this we are able to design different spectrograms.

For this research we have chosen additive synthesis of sinusoidal components, but other synthesis techniques can be considered to be controlled with these histogram sequences. This is a venue we might explore in the future.

7 Conclusion and Further Work

In this paper we have reported on the synthesis of sounds by the computer simulation of natural systems with CA. The multitype voter model exhibits rich dynamics from a very simple rule and few input parameter specifications. From a histogram analysis we have obtained complex structures endowed with musical features, suitable for the design of spectrograms. A sound design process is possible thanks to the controllability achieved.

We have synthesized single sounds (with durations in the order of seconds) with dynamic spectra and controlled complexity, and also polyphonic sound textures (with durations in the order of tens of seconds) with interesting internal evolutions.

We are currently investigating the inclusion of a mutation process –mutations are often considered in genetic models. The possibility that new species can enter the system through genetic mutation suggests potential applications in the synthesis of polyphonic sound textures.

Examples of sounds synthesised by our new method will be (or were) played at the conference and they are available by request.

References

1. Burraston, D., Edmonds, E.: Cellular Automata in Generative Electronic Music and Sonic Art: A Historical and Technical Review. Digital Creativity 16, 165–185 (2005)
2. Chareyron, J.: Digital Synthesis of Self-Modifying Waveforms by Means of Linear Automata. Computer Music Journal 14, 25–41 (1990)
3. Clifford, P. and Sudbury, A.: A Model for Spatial Conflict. Biometrika. 60, 581–588 (1973)
4. Cox, J.T., Durrett, R.: The Stepping Stone Model: New Formulas Expose Old Myths. The Annals of Applied Probability 12, 1348–1377 (2002)
5. Cox, J.T., Griffeath, D.: Recent results for the stepping stone model. In: Percolation Theory and Ergodic Theory of Infinite Particle Systems. Springer, New York (1987)
6. Holley, R.A., Liggett, T.M.: Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. The Annals of Probability 3, 643–663 (1975)
7. Kimura, M.: Stepping-stone Model of Population. Annual Report of the National Institute of Genetics 3, 62–63 (1953)
8. Liggett, T.M.: Interacting Particle Systems. Springer, New York (1985)
9. Miranda, E.R.: At the Crossroads of Evolutionary Computation and Music: Self-Programming Synthesizers, Swarm Orchestras and the Origins of Melody. Evolutionary Computation Journal 12, 137–158 (2004)
10. Miranda, E.R., Biles, J.A.: Evolutionary Computer Music. Springer, London (2007)
11. Miranda, E.R., Wanderley, M.M.: New Digital Musical Instruments: Control and Interaction beyond de Keyboard, A-R edn., Middleton, WI (2006)
12. Serquera, J., Miranda, E.R.: Spectral Synthesis and Control with Cellular Automata. In: Proceedings of the International Computer Music Conference ICMC, Belfast, UK (2008)
13. Wolfram, S.: A New Kind of Science Online, `http://www.wolframscience.com/reference/notes/876b` (accessed on February, 2009)
14. Wolfram, S.: Computational Theory of Cellular Automata. Communications in Mathematical Physics 96, 15–57 (1984)