Evolving Complex Sounds with Cellular Automata: an Approach to Granular Synthesis

Eduardo Reck Miranda
Sony Computer Science Laboratory - Paris
6 rue Amyot
75005 Paris - France
miranda@csl.sony.fr

Abstract

In this paper we introduce a system in which a cellular automaton is employed to control the evolution of synthesised complex sounds. One of the most interesting sound synthesis techniques available to musicians today is the *granular synthesis* technique. This technique involves the production of thousands of tiny sound granules (e.g. 35 milliseconds long) that combine to form complex and dynamic sounds. The main difficulty with granular synthesis is the specification of the nature of each sonic granule and how it will affect the overall result. Traditionally, granular synthesis systems have used probabilistic methods to control this. In this paper we describe an alternative method based on cellular automata. Our method has proved to produce sounds of a noticeably better quality than probabilistic methods in terms of the dynamics of unfolding and the naturalness of the overall form.

1 Introduction

Computer sound synthesis technology has enabled musicians to control sound at its most fundamental levels, from the fine-tuning of its spectral components to the dynamics of unfolding. Perhaps one of the most interesting sound synthesis techniques available today is the *granular synthesis* technique (Miranda, 1998). Granular synthesis of sounds involves the production of thousands of tiny sound granules (e.g. 35 milliseconds long) that combine to form complex and dynamic sounds (Figure 1).

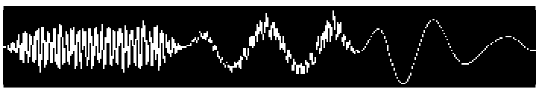

Figure 1:
A sequence of 3 different tiny sounds.
A rapid succession of thousands of tiny sounds forms larger complex sounds.

The sounds from granular synthesis tend to exhibit a great sense of movement and flow. This synthesis technique can be metaphorically compared with the functioning of a motion picture in which an impression of continuous movement is produced by displaying a sequence of slightly different images at a rate beyond the scanning capability of the human eye.

The composer Iannis Xenakis is commonly cited as one of the mentors of granular synthesis. In the 1950s, Xenakis (1971) developed important theoretical writings where he laid down the principles of the technique. The first computer-based granular synthesis systems did not appear, however, until Curtis Roads (1991) and Barry Truax (1988) began systematically to investigate the potential of the technique.

The main difficulty of granular synthesis for musicians is the specification of the nature of each individual sonic granule and how it will affect the overall result. So far granular synthesis systems have used stochastic formulae (i.e. probabilities) to control the evolution of the granules; for example, to control the waveform and the duration of the individual granules as they evolve in time. In this paper we present an alternative method based upon cellular automata.

The rest of this paper is structured as follows: the next section introduces the basics of cellular automata (here we only give the background needed the to follow the rest of the paper), followed by an introduction to ChaOs, the cellular automata used in our system. Next, we present the synthesis engine we have developed. Here we explain how ChaOs controls the granular synthesiser and demonstrate the reasons for using this particular automaton. Then we comment on the sounds the system can produce and end the paper with a conclusion and final remarks.

2 The basics of cellular automata

Cellular automata (CA) are computer modelling techniques widely used to model systems in which space and time can be represented discretely, and quantities take on a finite set of discrete values.

CA were originally introduced in the 1960s by von Neumann and Ulan as a model of biological self-reproduction (Cood, 1968). They wanted to know if it would be possible for an abstract machine to reproduce; that is, to automatically construct a copy of itself. Their model consisted of a two-dimensional grid of cells (i.e., variables), each cell of which had a number of states (i.e., values for the variables), representing the components out of which they built the self-reproducing machine. Controlled completely by a set of rules designed by its creators, the machine would extend an "arm" into a virgin portion of the grid, then slowly scan it back and forth, creating a copy of itself - reproducing the patterns of the cells at another location in the grid.

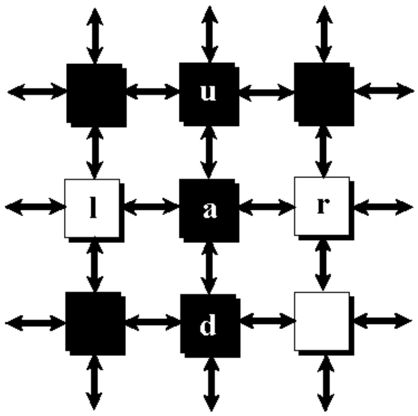

Figure 2:
An example of a matrix of cells where the transition function computes the value for a cell at time t+1 based upon the values of its four neighbours.

Since then CA have been repeatedly reintroduced and applied for a considerable variety of modelling purposes; see, for example, Wolfram (1994). Many interesting algorithms have been developed during the past thirty years. In general, CA are implemented on a computer as a regular array or matrix of cells; they normally can have one, two or three dimensions. Each cell may assume values from a finite set of integers and each value is normally associated with a colour. The functioning of a cellular automaton is displayed on the computer screen as a sequence of changing patterns of tiny coloured cells, according to the tick of an imaginary clock, like an animated film. At each tick of the clock, the values of all cells change simultaneously, according to a set of transition rules that takes into account the values of their neighbourhood. Figure 2 portrays an example of a matrix of cells where the transition function computes the value for a cell a at time $t+1$ based upon the values of its four neighbours: u, d, r and l; in this case, an example of a rule could be: "a cell will value 1 (black) at $t+1$ if at least 2 neighbours value 0 (white) at time t".

The set of the values of all cells at a specific lapse of time t is called a configuration c^t. Given an initial configuration c, a global transition function F determines a sequence $c, c^1, c^2, \ldots, c^n$, called propagation, where $c^{t+1} = F(c^t)$ for all t.

3 The cellular automaton used in our system

The cellular automaton used in our system is called ChaOs (an acronym for Chemical Oscillator). ChaOs is an adapted version of a cellular automaton that has been used to model the behaviour of a number of oscillatory and reverbatory phenomena such as Belouzow-Zhabotinsky-style chemical reactions (Dewdney, 1988).

ChaOs can be thought of as a matrix of cells containing identical electronic circuits. At a given moment, cells can be in any one of the following states: quiescent, depolarised or collapsed. The automaton tends to evolve from an initial random distribution of states in the grid of cells towards an oscillatory cycle of patterns (Figure 3).

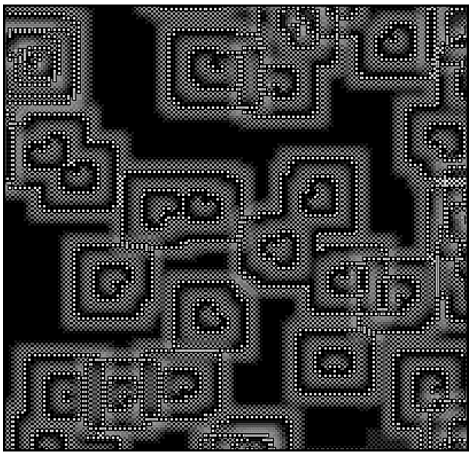

Figure 3:
The ChaOs CA in action.

The metaphor behind the ChaOs model is as follows: A cell interacts with its neighbours (normally 8 neighbours) through the flow of electric current between them. There are minimum (V_{min}) and maximum (V_{max}) threshold values which characterise the state of a cell. If its internal voltage (V_i) is below V_{min}, then the cell is quiescent. If it is between V_{min} (inclusive) and V_{max} values, then the cell is being depolarised. Each cell has a couple of resistors (*R1* and *R2*) which is aimed at maintaining V_i below V_{min}. But when it fails (that is, when V_i reaches V_{min}) then the cell becomes depolarised. There is also a capacitor (*k*) which regulates the rate of depolarisation. The tendency, however, is to become increasingly depolarised with time. When V_i reaches V_{max}, the cell collapses. A collapsed cell at time *t* is automatically restored to a quiescent state at time $t + 1$.

Assuming that *M* is a discrete set of *n* states m_p, such that $n >= 3$, then we can define the states of the cells as follows:

if $V_i < V_{min}$ then m_0 (i.e., quiescent)
if $V_i >= V_{min}$ and $V_i < V_{max}$ then m_p (i.e., depolarised), where $p > 0$ and $p < n-1$
if $V_i >= V_{max}$ then m_{n-1} (i.e., if collapsed then restore it)

In practice, the states of cells are represented by a number between 0 and *n-1* (*n* = amount of different states). A cell in state 0 corresponds to a quiescent state, whilst a cell in state *n-1* corresponds to a collapsed state. All states in between exhibit a degree of depolarisation, corresponding to the number of their state. The closer the cell's state value gets to *n-1*, then the more depolarised it becomes.

The global transition function *F* is defined by three rules simultaneously applied to each cell, selected according to its current state: quiescent, depolarised or collapsed. The rules are as follows:

if quiescent: it may or may not become depolarised at the next tick of the clock (i.e., *t+1*). This depends upon the number of quiescent cells Q_{cells} in its neighbourhood (normally 8 neighbours), the number of collapsed cells C_{cells} in its neighbourhood and the resistance of the system to the depolarisation of its cells (*R1* and *R2*):

$$m^{t+1}(a) = int(Q_{cells}/R1) + int(C_{cells}/R2)$$

if depolarised: the tendency is to become more depolarised as the clock t evolves. Its state at the next tick of the clock *t+1* depends upon two factors: the capacitance *k* of the nerve cell and the degree of depolarisation of its neighbourhood. The degree of depolarisation of the neighbourhood is the sum of the numbers that correspond to the states of the neighbours S divided by the number of quiescent neighbours Q_{cells}:

$$m^{t+1}(a) = int((S/Q_{cells}) + k)$$

if <u>collapsed:</u> a collapsed cell at time t generates a new quiescent cell at time $t+1$:

$$m^{t+1}(a) = m_0$$

Once implemented on a computer, the user can specify the behaviour of ChaOs by setting up the following parameters:

the number of n cell values (or colours), such that $n >= 3$
the resistances $R1$ and $R2$
the capacitance k
the dimension of the grid

4 The synthesis engine

4.1 The behaviour of the model

The CA described above is interesting because its behaviour resembles the way in which most of the natural sounds produced by acoustic instruments evolve: the automaton tends to evolve from an initial wide distribution of cell's states in the grid towards oscillatory cycles of patterns. Acoustic sounds also tend to converge from a wide distribution of their partials at the onset, to periodical oscillations. The organisation principle of this CA intuitively suggests that it could be applied to control the production of a large number of sonic granules which together form a complex sound event: the overall sound would begin with a highly disorganised sequence of granules that would gradually settle into an oscillatory pattern sequence. To find an effective way to map the behaviour of the CA onto the parameters of the synthesis algorithm has not been, however, a straightforward task. We devised and tested several methods; but only a few have produced interesting sounds. We introduce below the technique which has been adopted.

4.2 The mapping technique

Each sonic granule produced by the system is composed of several partials; each partial is a sinewave produced by an oscillator (in theory, we could use any other waveform here, but for the sake of simplicity of the first experiments with the system we have used only sinusoids). An oscillator needs three parameters to function: frequency, amplitude and duration (in milliseconds) of the sinewave (Figure 4).

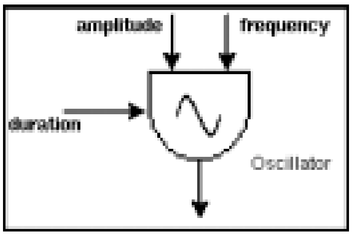

Figure 4:
Each component of the spectrum of a granule is generated by a different digital oscillator.

The CA controls the frequency and duration values of the components of each sound granule, but the amplitude values are pre-set by the user beforehand; see Miranda (1998) for a comprehensive introduction to sound synthesis techniques.

Each possible cell state m_p is associated beforehand to different frequency values (e.g., $M = \{m_0=110Hz, m_1=220Hz, m_2=440Hz, \ldots, m_n=N\}$ and oscillators are associated to a group of cells.

Each sound granule is in fact the product of the additive synthesis of sinewaves (Figure 5): at each configuration c, c^1, c^2, ..., c^n of cells, all oscillators simultaneously produce sinewaves, whose frequencies are determined by the arithmetic mean over the frequency values associated to the states of their corresponding cells. Thus, the frequency values of the components of a granule at time t are established by the arithmetic mean of the frequencies associated with the states of the cells allocated to different oscillators. As an example of a grid of 400 cells associated to 16 oscillators of 25 cells each is shown in Figure 6.

The duration of a whole sound event is determined by the total number n of configurations c, c^1, c^2, ..., c^n and the duration of the individual granules; for example, 100 configurations of granules of 40 milliseconds would result in a sound event of 4 seconds duration.

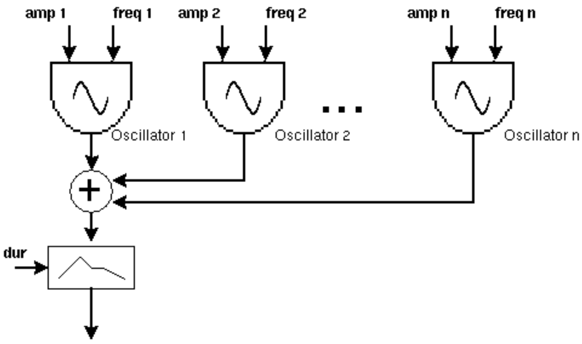

Figure 5:
The additive synthesis of sinewaves.

In order to operate the system, the user specifies beforehand: the dimension of the grid, the amount of oscillators, the allocation of cells to oscillators, the frequencies associated to the states, duration of the granules, and the parameters of ChaOs, that is, the number of states in set M, the resistances $R1$ and $R2$, the capacitance k, and the number of iterations (i.e., maximum value for t).

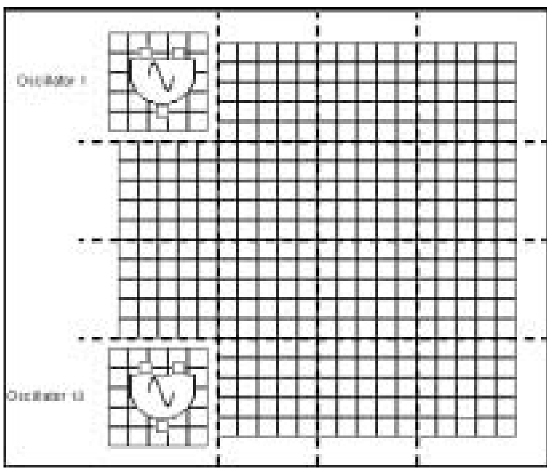

Figure 6:
An example of a grid of 400 cells allocated
to 16 digital oscillators.

5 Commentary on the results

The mapping method described above is interesting because it explores the behaviour of ChaOs in order to produce sounds in a way which resembles the evolution of sounds produced by most acoustic instruments during their production; their harmonics converge from a wide distribution (as in the noise attack of a sound) to oscillatory patterns (the characteristic of a sustained tone). The random initialisation of states in the grid produces an initial wide distribution of frequency values, which tend to settle to a periodic fluctuation.

We have synthesised sounds using up to 64 different states (that is up to 64 different frequency values) and up to 64 oscillators, on grids of up to 4,000,000 cells (2,000 x 2,000). The results have tended to exhibit a great sense of natural movement and flow, and yet most of these sounds cannot be found in the "real" acoustic world. Some results, however, resemble the sounds of flowing water, bird calls and insects.

Variations in tone colour can be achieved by varying the frequency values associated with the states of the cells. For example, if the set of frequencies contains values lower than 220 Hz, then the result will be a dull sound in the lower band of the audible range. Conversely, if the set contains frequencies above 880 Hz, then the results will be a brighter sound in the higher band of the audible range.

The size of the grid and the amount of cells per oscillator are largely responsible for the degree of granularity of the spectral variation: a larger amount of cells per oscillator produces finer granulation effects, whilst a shorter amount produces coarser granulation effects.

The length of the individual granules also plays a key role in the overall result. The acoustic effect of the variation of the length of the individual granules can be summarised as follows: very short lengths (e.g., 35 milliseconds) produce textures of sparkling bubble-like cloud of sounds, whereas larger lengths (e.g., 800 milliseconds) produce sequences of sound "strokes"; lengths above 1 second produce sequences of notes of different timbres.

Different rates of transition from noise to oscillatory patterns are obtained by changing the values of $R1$, $R2$ and k.

6 Conclusion and final remarks

In this paper we have presented a cellular automata-based approach to model the evolution of the complex sounds produced by the granular synthesis technique. On the whole, our system has demonstrated that CA is a plausible solution to control the evolution of highly dynamic and complex synthesised sounds.

Our method proved to be more efficient than other methods for controlling the production of the individual granules of a granular synthesiser because it drives the evolution of the synthetic (artificial) sounds in a way which resembles the functioning of most acoustic instruments: the random initialisation of cells in the grid of the CA produces an initial wide distribution of frequency values, which tends to settle to an oscillatory cycle. However, the mapping between the behaviour of the CA and the synthesis parameters is not always readily apparent. We have come to conclude that those alternatives which explore time-based representations of sounds tended to work better; hence the granular synthesis approach.

This author has used the system to generate sounds to compose *Olivine Trees*, a prize-winning electroacoustic composition (Luigi Russolo 1998 competition, Italy), and *Electroacoustic Samba X*, recently released on CD (Miranda, 1998b).

Acknowledgements

This work started in 1994 as a TRACS research project at Edinburgh Parallel Computing Centre (EPCC) and at the Faculty of Music of Edinburgh University. This author is grateful to Peter Nelson (Faculty of Music), and Robert Fletcher and Martin Westhead (EPCC) for their support and valuable contributions during the early design and engineering stages.

A working version of this system, designed and programmed in collaboration with Joe Wright, is available on the accompanying CD-ROM of this author's book *Computer Sound Synthesis for the Electronic Musician* (Miranda, 1998a).

References

Cood, E. F., *Cellular Automata*, London (UK): Academic Press, 1968.

Dewdney, A. K., "Computer recreations: The hodgepodge machine makes waves", *Scientific American*, August:104-107, 1988.

Miranda, E. R., *Computer Sound Synthesis for the Electronic Musician*, Oxford (UK): Focal Press, 1998a.

Miranda, E. R., “Electroacoustic Samba X”, *Electroacoustic Music from Latin America*, CD oo45, Black Rock (USA): O.O. Discs Inc., 1998b.

Roads, C., “Asynchronous Granular Synthesis”, De Poli, G. *et al*. (Eds.), *Representation of Musical Signals,* Boston (USA): The MIT Press, 1991.

Truax, B., “Real-Time Granular Synthesis with a DSP Computer”, *Computer Music Journal*, 12(2):14-26, 1988.

Wolfram, S., *Cellular Automata and Complexity*, Reading (USA): Addison-Wesley, 1994.

Xenakis, I., *Formalized Music*, Indiana (USA): Indiana University Press, 1971.