LIQUIPRISM : GENERATING POLYRHYTHMS WITH CELLULAR AUTOMATA

Alan Dorin

Centre for Electronic Media Art,
School of Computer Science & Software Engineering,
Monash University, Clayton, Australia 3800
aland@csse.monash.edu.au

ABSTRACT

This paper describes a MIDI instrument for the generative composition of polyrhythmic patterns. Patterns are determined by the activation of cells in a non-homogeneous set of inter-connected cellular automata grids which form the faces of a cube in virtual three-dimensional space. This paper discusses the selection of the rules of the cellular automata which enable rhythmic outcomes to be generated by the system, and the motivation for organizing the cells in their current configuration. The fluid sonification and visualization of the cellular automata arrays was a specific aim of this work for gallery installation. Hence the means by which the rigidly-defined and deterministic automata behave was intended to be visualized and sonified in such a way as to convey a sense of emerging liquidity from the mechanism. A brief discussion of these issues appears in this paper.

1. INTRODUCTION

The generation of complex patterns from simple rules is a constant source of fascination for practitioners in the fields of computer generated imagery and sound [1,2,3,4,5,6,7,8]. This interest takes root in a deeper drive to comprehend the world around us. Where we may, we reduce complex outcomes to the combination of simple principles. We look for causal connections between otherwise mysterious and unpredictable events. We seek order in chaos and, in our own work, seek *complexity for free*.

This paper describes a musical event generator intended for gallery installation for the production of polyrhythms. The complex patterns it generates emerge from the neighbourhood interactions of cellular automata, simple machines arranged in grids and connected only to their closest neighbours (See [4,5,16,23] and below for a more detailed description). The tool may be used as an instrument for generative composition or as a stand-alone rhythm generator which runs without interference from a human user.

The combination of multiple independent and/or intertwined rhythmic patterns gives an emergent and intricate shimmer with a "life" of its own. This phenomenon is quite irreducible to its component parts. It is this aspect of rhythm which first attracted the author to the idea of generating interwoven processes in a manner reminiscent of the functioning of a living thing.

Steve Reich hits the nail on the head when he states "Sometimes everything just comes together and suddenly you've created this wonderful organism" [9, pp7] The resulting piece of music is alive, an organism, in the sense that a multitude of interacting processes combine to produce a transient dynamic entity with an energy not apparent on the dissecting table. Michael Nyman [10, p149] also uses the term musical "organism" in a discussion of the work of Phillip Glass.

Xenakis' percussion work *Pleiades* [11] and Reich's pieces (for example *Drumming*, *Music for 18 Musicians* and *Sextet* [12,13,14]) are good examples of the dynamism which may emerge from the careful orchestration of simple patterns. Of course Reich is best known for his use of *phasing* – a term coined to describe the procedure of playing repetitive patterns of different lengths against one another so that they gradually shift out of (and then back into) phase. Music from African, Chinese and Cuban sources is also frequently based around complex polyrhythmic structures [15, p35].

Reich was, in his earlier days at least, specifically interested in the *process* by which his music arose. "Once the process is set up and loaded it runs by itself" [10, p152] The simplicity of a process such as playing a repetitious drum pattern, or the operation of mechanical tape players, contrasts strongly with the complexity of the result. Here then the composer has demonstrated a kind of power over sound which is not a reflection of their ability to conceive of intricate rhythms, so much as it illustrates an understanding of the origins of complexity in simplicity. A fascination with these ideas has played a part in the production of the software described shortly in this paper.

The behaviour of Wolfram's one-dimensional cellular automata (CA) [16] is of relevance to the present discussion. Wolfram established a closed loop of automata which is usually depicted as a row of cells. The two end cells are assumed to be neighbours in the same way that adjacent cells within the line neighbour one another.

Wolfram's cellular automata are binary state machines – they may be in the state *on* or *off*. The state a given automaton will be in at a discrete time step immediately following the present one is determined by its present state and the present state of its two neighbours. All automata in a row are updated synchronously.

It is possible to exhaustively document the patterns Wolfram's connected machine produces by running through all the possible rules governing the state changes of its automata. Wolfram found four general types of behaviour in his machines. He found machines which may produce or move into:

(i) a homogeneous state (limit-point);
(ii) simple, separated, periodic structures (limit-cycle);
(iii) chaotic aperiodic patterns (strange attractors);
(iv) complex patterns of localized structures.

The relationship between these behaviours and the production of music or imagery using procedural means has been discussed in [17]. Various authors and composers have produced and exhibited systems which generate musical structures from cellular automata [18, 19].

Wolfram's CA's are, like any dynamical system, capable of producing patterns for use as rhythms in music. However in the work presented here, two-dimensional arrays of cells connected in a grid (like the familiar *Game of Life* cellular automata devised by Conway [4,5]) are employed. These finite grids are then connected along their edges into a cube, rather than employing the usual toroidal structure. The reasons for this are explained shortly.

Following this, the present system for generating musical patterns is explained in some detail. The system's visual interface is also described since it heavily influences the effective presentation of the concept behind the work.

2. THE LIQUIPRISM CONCEPT

The Boolean Sequencer is a previously published tool produced by the author [24]. This device utilizes Boolean logic gates arranged in tracks to generate multi-layered rhythms in an interactive fashion, hence the idea was related to the concept behind the Liquiprism. However, the sequencer was not designed so much as an artistic work but rather as a tool for the generation of musical structures. Therefore the interface design issues which arose previously were very different from those encountered here.

Liquiprism is a (toy) artistic work reflecting the author's fascination with Biology and its roots in Chemistry and Physics. The work is intended to focus on the seeming conflict between the determinism and regularity or cellular automata, and the apparently complex and fluid changes of form and pattern their interactions produce. Similarly, the rigid and clinical platonic structure of the cube on which Liquiprism is based, is representative of the ordered way in which molecules assemble themselves into crystal forms. The flickering and intricate patterns which slide across the surface of the prism so rapidly as to be impossible to clearly perceive, are the processes of life. From the static, comes the dynamic. Based on the simple, even geometric form, appears the fertile and liquid organism.

Taking these issues into account, it was important to the author that the grid be perceived as clinical and geometric. Hence, a rigid grey (or black) array of points in space was constructed. As for the life forms which flow across its surface, what better way than to represent these visually by pale luminescent light which weightlessly flickers and scatters? It is true that these design decisions are not a revolutionary means of portraying the changing states of a cellular automata array! However in this instance they seem to be most appropriate, even if they are not unusual.

As far as the sonic landscape produced by the Liquiprism is concerned, it needed also to work on this contrast of rigidity and fluidity. In exhibition, Liquiprism has been connected to a Novation Nova analogue modelling synthesizer. The palette of sounds the prism triggers typically varies between simple 'squelchy' (to employ technical jargon) tones and pure, clean sine tones. This choice again reflects the desire to present a flickering, shimmering organism which emerges from the interactions of simple tones, rather than to begin the construction from complex building blocks (such as might be created using temporally evolving tones or lengthy samples). Further details about the method by which the tones are controlled are given in the sections which follow.

3. CELLULAR AUTOMATA BEHAVIOUR

For the sake of simplicity, only binary-state cellular automata are considered here. (The discussion is easily generalized to cells with more than two states.) The state of a cell in a CA grid at a future time is governed by a transition table which operates on the cell's inputs – these being the state of its (eight or four) neighbouring cells, and its own current state. The set of states a system passes through as its cells are updated is known as its *trajectory*.

The first important concept to grasp is that any actual implementation of an automata grid has a finite number of cells. Therefore the machine as a whole has a finite number of possible states (for a binary CA, specifically 2^N where N is the number of cells). If the transitions are deterministic, if the system returns to a state from which it previously emerged (which it must do eventually since there are a finite number of possible states), the system must be in a *limit cycle.* The number of steps in a limit cycle may range from one (a limit point) to 2^N. The same is true of a subset of cells within the array provided it does not receive interference at its boundaries.

The set of machine states which lead the system to fall into a particular limit cycle is the *basin of attraction* for that limit cycle. The limit cycle is the *attractor* for the basin. A cellular automata grid must have at least one limit cycle but some array/transition combinations have many, each with its own basin of attraction. Some basins are very broad (a large number of machine states lead to their attractor), others are quite narrow. Whilst the machine is within the basin of attraction but not yet within its limit cycle the machine is said to be in a *transient*.

Besides basins of attraction, the other major property of interest is the susceptibility of a particular attractor to disturbance. Some limit cycles are more stable than others. That is, even if some cells in the system have their state flipped, the basin of attraction around the attractor ensures the trajectory of the system returns from its transient to the previous limit cycle. Sometimes, even major alterations to a system do not upset limit cycles. However in the case of cellular automata patterns, this is not often the case. These systems have patterns which are typically brittle, although in general the *kinds* of behaviour they produce after a disturbance may be predictable.

4. WHY USE CELLULAR AUTOMATA?

Of course multi-layered rhythms may be constructed and edited by hand in a conventional sequencer or using a tool like that described elsewhere [24]. There is no disputing the value of this technique. However the current task is the generation of such patterns under the control of the computer, possibly with some minor activity governed by a user.

Some CA rule sets are known to be effective producers of complex patterns and cyclic activity. For example Conway's *Game of Life* falls into this category [4,5]. Some researchers have used CA's like this for music event generation [18], however the mapping from CA to music is almost arbitrary. Why should the CA rules produce music? The researchers may as well revert to Cage's dice or coins [21, p132] or any of a myriad of other physical systems, and then attempt an arbitrary mapping from arbitrarily chosen system to musical events – an idea this author feels was exhausted in the sixties.

Effective transition rules for cellular automata are notoriously difficult to devise. Wolfram's studies and the studies of Sims [22] and Langton [23], as well as this author's own investigations, indicate that of the enormous range of possible rules, the majority reliably lead a system to a fixed point or short cycle, rather than to one in which lengthy cyclic or complex patterns emerge. It is this property which was *utilized* in the current work in order to avoid the arbitrary nature of CA to music mappings. Rather than depend on a single automata grid to produce a lifeless oscillating limit-cycle, six coordinated grids are used here in concert. The activity of these may be stimulated by a user, or under computer control, in order to 'stir up' the sonic and visual environment, or to let it rest in a desirable cycle/set of cycles.

5. LIQUIPRISM TRANSITION RULES

Liquiprism is a set of six cellular automata grids arranged over the surface of a cube. The cells count the eight nearest cells in adjacent rows and columns as their neighbours. The automata cells which border an edge of the cube, and thus border also another automata grid, count the closest cells on the neighbouring faces as their neighbours (as if the surface were flat). Hence, it is possible for the patterns of automata to move from one face of the cube to another across an edge. This contrasts with the usual torroidal configuration for cell neighbourhoods which causes patterns to wrap around a grid from left to right, top to bottom and vice versa.

Another aspect of the design aimed at altering the behaviour of the system from that typically constructed on a CA was to employ different update rates on each of the prism faces. Hence, although patterns may propagate from one face to the next across edges, these patterns change in speed as they do so. This also allows for simple limit cycles on different faces to phase against one another, even if their cycles are of the same length measured in updates.

The author has experimented (by trial and error) with numerous rule sets for updating the automata. Unfortunately, besides a 'gut feeling' about which rules will generate interesting results and which will not, there is little by way of guidelines to assist the programmer. What follows is a description of one means of updating the automata which is employed at the time of writing this paper.

Currently, two transition tables/rule-sets are employed simultaneously to update the system. The first of these is 'conventional' in that its updates are symmetrical about each cell. That is, it does not matter which neighbours of a cell are in which states, only the number of neighbours in a particular state are considered in determining a transition. This set is specified as follows:

```
if (currentCell is ON)
{
 if(#NeighboursOn=2)OR(#NeighboursOn=3)
 { currentCellNextState ->ON }
 else
 { currentCellNextState ->OFF }
}

else
{
 if (#NeighboursOn>=4)
 { currentCellNextState ->ON }
 else
 { currentCellNextState ->OFF }
}
```

Hence, this rule set ensures, like the rules of *The Game of Life*, that an ON cell remains ON if it has 2 or 3 neighbours ON. This may be considered a rule of 'mutual support' tempered by 'over-crowding'.

If a cell is OFF it will remain OFF unless it has sufficient 'start-up' support – namely 4 ON neighbours. This condition must change if the cell is to survive the next generation and not suffer from over-crowding. Hence, this rule set requires a new cell in an area to *replace* one of the neighbours which spawned it.

The second set of rules for the automata update has a set of conditions for handling cells in the ON state which is identical to the first set. I.e. an ON cell remains on if it has 2 or 3 ON neighbours. However, for handling OFF cells:

```
if (currentCell is OFF)
{
 if (belowNeighbour is ON) AND
    (1in3 chance)
 { currentCellNextState ->ON }
 else
 { currentCellNextState ->ON }
}
```

These rules introduce a stochastic element to the system and also a tendency for patterns to propagate 'up' the grid.

The first and second rule sets are employed depending on the overall level of activity on the prism. An active grid will be updated by the first rule set. When the grid settles into an inactive or stable state (measured against a threshold of the number of state changes per step), the second rule set is used to update cells in order to cause cascades of patterns moving across the prism's surface. When this triggers the prism into activity, the first rule set is returned to control.

In addition to these rules sets, input from the user may be collected from a sensor or the mouse. Currently this alters the viewing angle of the cube, and may trigger a random burst of activity on its front-most face. This allows the presence of a viewer to alter the cube's behaviour, to stir it if it is too subdued or to let it rest if its activity is, for whatever reason, interesting.

6. MAPPING CELLULAR TO SONIC ACTIVITY

As indicated earlier, the choice between different mappings from cell states to sonic events is somewhat arbitrary. In this case, the programmer had certain aesthetic criteria which needed to be fulfilled, namely the construction of complex rhythms from simple elements. Therefore at any time-step, only a few of the active cells generate sonic events. The cells which generate MIDI events are those which are changing from the OFF state to the ON state. To ensure there is not a jumble of sound from each face, a threshold is set (currently at two) to limit the number of notes which may be played simultaneously per face. In the event of note culling, 'lower' cells on the face are culled first.

Each face of the cube is assigned a different MIDI channel and each of these is assigned a different voice on the multi-timbral synthesizer. Each cell is assigned a different pitch according to its position. Currently, pitches rise a semi-tone for one step across a grid, and a fifth for one step up the grid. Hence, as the rules have been selected to frequently generate various short-cycle oscillators, there are frequent semitone and fifth intervals. Since different faces update at different rate and have different timbral qualities, the rhythms that arise from these may be quite complex.

Besides pitch, clearly there are numerous other MIDI parameters which might be controlled by the Liquiprism. Experiments with tone envelopes, velocity and duration for example have yet to be attempted.

7. SOME SAMPLE CONFIGURATIONS

Some simple configurations are tabulated here as an example of the type of patterns which the current implementation produces. An 'O' indicates a cell which becomes fixed in the ON state. Symbols '   ' and '⊕' indicate cells which oscillate in different phases.

Stable configurations include:

```
O O   O O O   O O
O O     O       O
```

Figure 1: Sample stable configurations

Two-step oscillating configurations include:

```
  O     O O     O
O   O     ⊕   O   O
O   O   O O     O
```

Figure 2: Sample two-step oscillators

7.1 A NOTE ON TRANSITION RULES

The description above details the mappings and transitions which *were* implemented in the Liquiprism. Before concluding this paper, here are a few words about rules which were *not* implemented on the system and some reasons for these decisions.

Clearly, if short limit cycles are the desired outcome, the rules for the update of the grid cells need to take this into account. Hence, systems which tend to give rise to a multiplication of the activity on the grid were not implemented. These rules can be detected since they usually require an OFF cell to have only a small number of neighbours (such as one or two) to trigger it to switch ON. Similarly, if an ON cell has many neighbours and the system does not implement a culling operation which will turn such a cell OFF, the amount of activity on the grid can run amuck.

There is also a balance to be struck between completely extinguishing activity on the grid and keeping short limit cycles alive. This may be obtained by ensuring ON cells with a few neighbours are kept ON despite fluctuations in the number of ON neighbours they have, whilst keeping in mind the requirements above to turn OFF cells which have too many or too few neighbours. Beyond these few basic principles and those outlined in the previous section, the set up of two dimensional cellular automata is best governed by chance, intuition, trial and error.

8. IMPLEMENTATION

The Liquiprism was implemented on a Macintosh G4 running MacOS 9.2. It was written in C++ using Metrowerks Code Warrior, the OpenGL library for Macintosh v1.1, and Opcode's OMS 2.3.8. OMS and OpenGL need to be installed to run the software. The software is available on the author's website for download (http://www.csse.monash.edu.au/~aland).

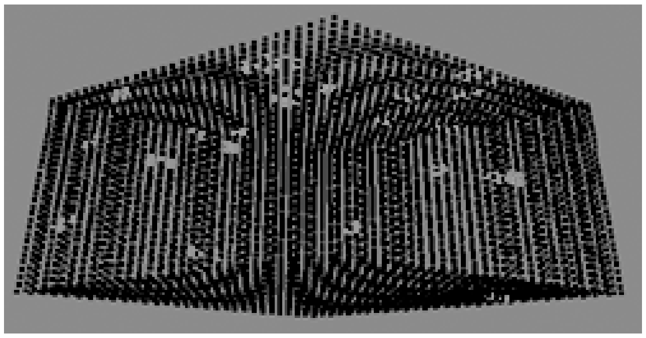

Figure 3: Liquiprism

9. CONCLUSIONS & FUTURE WORK

This paper documents an approach to generative electronic art which is based in an understanding of the underlying processes, and therefore stands as an attempt to give a visible life to them, that they may be appreciated by those not 'in the know'. Too much electronic art falls at the two extremes of the spectrum – either it is inaccessible to all but those who understand the underlying algorithmic processes, or it is beautiful to look at, but simplistic or uninteresting to those with insider knowledge. Hopefully this work may be appreciated at both levels.

In the future, it is hoped to further animate the behaviour of the cells in order to accentuate the lifelike properties of the patterns their state changes produce. Although the author is pleased with the current visual properties of the work, there is more which could be done to improve the system. Experiments with different MIDI parameters have also begun in order to improve the expressional capabilities of the prism and give it yet further 'life'.

10. REFERENCES

1 Prusinkiewicz, P., Lindenmeyer, A., "The Algorithmic Beauty Of Plants", Springer Verlag, 1990

2 Turk, G., "Generating Textures on Arbitrary Surfaces Using Reaction - Diffusion", SIGGRAPH 91, Computer Graphics, Vol. 25 No. 4, July 1991, ACM Press, pp289-298

3 Turing, A.M., "The Chemical Basis of Morphogenesis", Philosophical Transactions of the Royal Society, London, B, Vol. 237, 1952, pp37-72

4 Gardner, M. "Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life'", *Scientific American*, 1970, 223 (4), pp120-123

5 Gardner, M. "Mathematical Games: On Cellular Automata, Self-Reproduction, the Garden of Eden and the Game 'Life'", *Scientific American*, 1971, 224 (2), pp112-117

6 Dahlstedt, P., Nordahl, M. "Living Melodies: Co-evolution of Sonic Communication", in Proceedings, *First Iteration*, Dorin & McCormack (eds), CEMA, Monash University, Australia, 1999, pp56-67

7 Hiller, L., "Composing with Computers: A Progress Report", Computer Music Journal, Vol. 5, No. 4, 1981

8 Laske, O., "Composition Theory in Koenig's Project One and Project Two", Computer Music Journal, Vol. 5, No. 4, 1981

9 Schwarz, K.R., "Music For 18 Musicians, Revisited", liner notes in CD, *Music For 18 Musicians*, Reich, S., Nonesuch Records/Warner Music, 1997

10 Nyman, M., "Experimental Music, Cage and Beyond", 2nd edition, Cambridge University Press, 1999

11 Xenakis, I., "Pleiades", Grammofon AB BIS, 1990

12 Reich, S., "Drumming", Elektra / Nonesuch Records, 1987

13 Reich, S., "Music For 18 Musicians", Nonesuch Records, 1997

14 Reich, S., "Sextet . Six Marimbas", Elektra / Asylum / Nonesuch Records, 1986

15 Copland, A., "What To Listen For In Music", first published, McGraw-Hill, 1939, new edition, Mentor Books, 1999

16 Wolfram, S., "Universality and Complexity in Cellular Automata", in *Physica 10D*, North-Holland, 1984, pp1-35

17 Dorin, A., "Classification of Physical Processes for Virtual-Kinetic Art", in Proceedings, *First Iteration*, Dorin & McCormack (eds), CEMA, Monash University, Australia, 1999, pp68-79

18 McAlpine, K., Miranda, E., Hoggar, S.,"Making Music With Algorithms: A Case-Study System", Computer Music Journal, Vol. 23, No. 2, 1999, MIT Press, pp9-30

19 Dorin, A., "Liquiprism", in *Process Philosophies*, curated by Dorin & McCormack, First Iteration Conference, Monash University, Australia, 1999.

20 Kauffman, S.A., "The Origins of Order - Self Organization and Selection in Evolution", Oxford University Press, 1993

21 Ford, A., "Illegal Harmonies, Music in the 20th Century", Hale & Iremonger, 1997

22 Sims, K., "Interactive Evolution of Dynamical Systems", *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference of Artificial Life*, Varela & Bourgine (eds), MIT Press, 1992, pp171-178

23 Langton, C.G., "Studying Artificial Life with Cellular Automata", Physica 22D, North-Holland, 1986, pp120-149

24 Dorin, A. "Boolean Networks for the Generation of Rhythmic Structure", in Proceedings *Australian Computer Music Conference* 2000, Brown (ed.), ACMA, July 2000