Evaluating Mappings for Cellular Automata Music

Alexis Kirke[1], Eduardo Miranda[1]

[1] Interdisciplinary Centre for Computer Music Research (ICCMR),
University of Plymouth, Plymouth, UK {Alexis.Kirke, Eduardo Miranda}@plymouth.ac.uk

Abstract. We discuss the importance of choosing the right mapping for music composition generated from underlying ALife and emergent algorithms. Emergent algorithms are popular with composers both because their visual beauty inspires the composer, and because they can generate complexity from simple algorithm rules. Simple mappings to MIDI are preferred as they give a composer more control and predictability. However the wrong simple mapping may produce trivial music, and can fail to capture the visual aesthetic of the emergent phenomena in the music. Furthermore the mappings need some method of evaluation. To illustrate these issues we use the example of the Game of Life (GL). A polar co-ordinate mapping is introduced and we argue it is superior to previous GL mappings when both its simplicity and its visual-aesthetic capture is considered. This mapping is evaluated by comparing it to linear mappings using Zipf's Law and using a basic measure of structurality.

Keywords: Cellular Automata, Artifical Life, Game of Life, Algorithmic Music Composition, Zipf's Law, Entropy

1 Introduction

Most generative compositional systems based on emergence consist of two elements [1]:

a. An underlying emergent algorithmic process
b. A mapping which takes turns the results of the process into MIDI notes

Examples of these are given in [2] – including music generated by: chaos, genetic algorithms and cellular automata. When evaluating the properties of such compositional systems, there is sometimes a tendency to attribute more of the evaluation to (a) rather than with (b). Statements such as "there has been some success with using fractals/chaos/cellular automata in generating music" (i.e. labeling the generative process by its underlying algorithm) perpetuate this tendency. This is particularly the case when working with emergent systems where the focus is often on the level of complexity which can be generated by the underlying algorithm. However (b) can actually be adjusted in such a way as to vastly change the properties of any generative compositional system. Looking at, for example [3] a great deal of the work which went in to the system was in developing a mapping which allowed such a system to generate interesting music. There have been studies which discuss mapping

[1][4]. However they do not focus on examining the effects of different mapping types in emergent systems and how they may be objectively evaluated.

In our study we will look at the effects that changing the mapping can have on the output of a generative compositional process in emergent systems. To do this we will focus on the example of Cellular Automata, and introduce a novel mapping for the CA Game of Life. In this way we will investigate the effects of various mappings, and ways of evaluating those mappings. To examine the effects of the mappings, we will use Zipf's Law [5] and a structurality measure based on entropy.

2 Game of Life

Game of Life (GL) is a two-dimensional automaton invented by John Conway. "Conway was fascinated by the way in which a combination of a few simple rules could produce patterns that would expand, change shape, or die out unpredictably. He wanted to find the simplest possible set of rules that would give such an interesting behaviour." [6] The automaton consists of a finite $[m \times n]$ matrix of cells, each of which can be in one of two possible states: alive represented by the number one, or dead represented by the number zero; on the computer screen, living cells are coloured black and dead cells are coloured white (Figure 1).

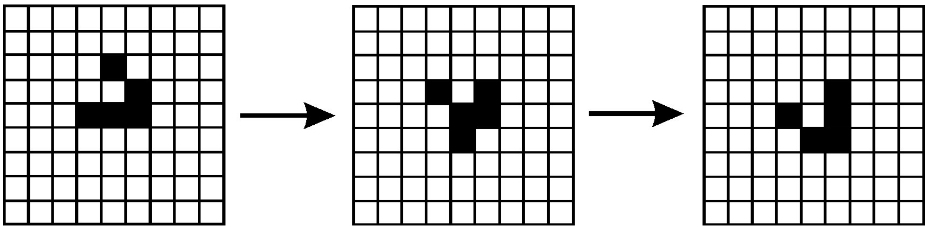

Fig. 1. Game of Life is a two-dimensional cellular automaton where each cell can be in one of two possible states: alive or dead.

The state of a cell as time progresses is determined by the state of its eight nearest neighbouring cells, as follows:

- Birth: A cell that is dead at time t becomes alive at time $t + 1$ if exactly three of its neighbours are alive at time t
- Death by overcrowding: A cell that is alive at time t will die at time $t + 1$ if four or more of its neighbours are alive at time t
- Death by exposure: A cell that is alive at time t will die at time $t + 1$ if it has one or no live neighbours at time t
- Survival: A cell that is alive at time t will remain alive at time $t + 1$ only if it has either two or three live neighbours at time t

In other words, considering that E represents the number of living neighbours that surround a particular live cell and F defines the number of living neighbours that surround a particular dead cell, the life of a currently living cell is preserved whenever $2 \leq E \leq 3$ and a currently dead cell will be reborn whenever $3 \leq F \leq 3$. A general form for representing transition rules is (E_{min}, E_{max}, F_{min} and F_{max}) where $E_{min} \leq E \leq E_{max}$ and $F_{min} \leq F \leq F_{max}$. The original Game of Life rules are represented as (2, 3, 3, 3). Clearly, a number of alternative rules other than (2, 3, 3, 3) can be set.

The original GL is also characterised by a number of interesting initial cell configurations that have given raise to intriguing emergent behaviour.

3 Generative Music Composition and GL

Since CA such as GL produce large amounts of patterned data and if we assume that music composition can be thought of as being based on pattern propagation and the formal manipulation of its parameters, it comes as no surprise that composers started to suspect that CA could be related to some sort of music representation in order to generate compositional material. One of the first composers to use CA was Iannis Xenakis, who used them in the mid of the 1980s "to create complex temporal evolution of orchestral clusters" for his piece Horos [7]. A number of pioneering experiments on using GL for generating music followed by composers such as [8], [9] and one of the authors of this paper [3]. There are two main reasons why GL has been at the centre of a significant number of algorithmic composition methodologies:

a. Comtivity - The generation of complexity in music [10], [11]
b. Inspiration - A desire to capture the aesthetic of GL in a non-visual art form [12]

Much previous work states (a) is the motivator. However they are also clearly motivated by (b) - the beauty of GL. The analysis of such motivations should be taken into account when choosing an appropriate mapping. We introduce a new mapping for GL based on (b) - a *Radial* mapping.

3.1 Radial Mappings

Examining the rules of GL described earlier, it can be seen that a cell's state is determined by the state of the cells around it. The neighbourhood could be rotated around the centre cell by 90, 180 or 270 degrees, and it would still have the same effect on the cell's state. This is a key reason that such attractive symmetric patterns emerge during iterations of GL. A number of researchers have discovered what are now well-known symmetrical life evolutions, and there is some discussion of the symmetrical tendencies of GL [13]. And even for those evolutions which are non-symmetrical, the emergence of complex broken symmetries from simple symmetrical rules is one element that appeals to our aesthetic sense during GL iterations.

However, the mapping rules used for music in previous investigations have involved some form of linear mapping of cells. For example [1] and [3] discuss the use of

Cartesian (*x,y*) co-ordinates of a live cell in generating musical notes. We proposed a mapping which better captures the radial symmetry inherent in the aesthetic of the GL –polar co-ordinates. In this mapping, the origin is placed at the centre of the GL board, and each live cell is mapped into the musical domain based on its (r,θ) co-ordinate rather than its *x* or (*x,y*) co-ordinate. This mapping approach is illustrated in Figure 2. Once this radial mapping is chosen a decision needs to be made as to what to map (r,θ) on to.

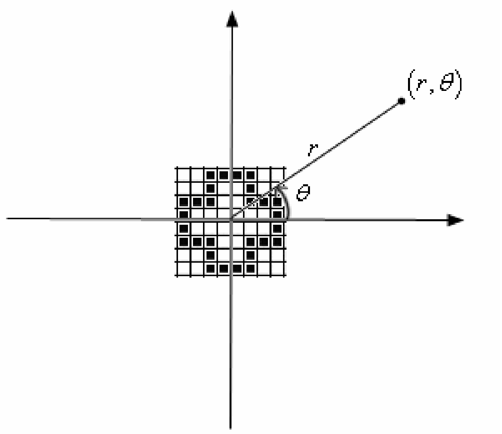

Fig. 2: A radial mapping for GL

For the purposes of this investigation the following algorithm was used:

1. Choose a *BPM* (beats per minute) value. Initialise a variable *baseBeat* to 0, choose a fixed note duration *D*
2. Run a generation of the GL
3. Iterate *r* from the centre of the board to the board's edge. For each of the 127 values of *r*, iterate θ around the board from 0 to 2π in 127 steps. The quantization into 127 steps is done for MIDI compatibility.
4. For each of the iterated values of θ, examine the cell at (r,θ). If it is alive generate a MIDI note with pitch proportional to θ, i.e. pitch = 128θ / (2π) and with duration *D*. And locate the MIDI note at beat = *baseBeat*+ $16rD/H$ where *H* is half the grid width in cells.
5. After completing the nested iterations of (r,θ) over the whole grid, update *baseBeat* to the beat after the last generated MIDI note.
6. Go back to (2) and repeat.

To illustrate this algorithm, let us consider the cell with (*x,y*) co-ordinate (2,2) in Figure 2. By performing the standard transformations from Cartesian to Polar co-ordinates (which will not be detailed here for space reasons) we get a quantized (r,θ) coordinate of (2, $\pi/4$). This is an eighth of the way ($\pi/4$) around the GL grid in Figure 2 (an eighth of the way up the MIDI pitch range of 0 to 127). So the value of the MIDI pitch will, by step 4 above, be $128(\pi/4)/(2\pi) = 16$. Looking at step (1) above, if we assume a duration *D* of 0.5 beats and a starting value of *baseBeat* of 0, then the start of the beat for the MIDI note calculated by step 4 is: 0 + 16x0.5(2/4) = 4. So (2,2) will generate a MIDI note of pitch 16 at beat 4. This algorithm has been kept as simple as possible, on the basis that the simpler a mapping is, the more control a composer has over the final output. It is possible to have such control and still generate complexity.

One final point needs to be highlighted – in the GL process in this paper we do not use the normal toroidal mapping, i.e. the "wrap-around" effect used in, for example [3]. The GL Universe is "clipped" at the edge of the grid, any cell coming to life outside of the grid is discarded. The rationale for this was that a toroidal mapping reduces the effects of the radial symmetry - a toroid has toroidal symmetry instead.

3.2 Linear Mappings in GL

The most common forms of Linear Mappings used in GL are Cartesian and Binary. Both forms have been discussed in past work on GL generative music already cited in this paper. Examples of these types of mappings are given below which only change steps (3) to (5) of the radial algorithm.

Cartesian Mapping

3. Iterate *y* down the GL grid from top to bottom (the Cartesian origin is fixed in the top left hand corner of the Grid). For each of the values of *y*, iterate *x* from the left of the board to the right.
4. For each of the iterated values of *x*, examine the cell at (*x*,*y*). If it is alive generate a MIDI note with pitch proportional to *x*, and of duration *D*. Locate the MIDI note at beat: *baseBeat*+ *D*+16*y*/*G*. *G* is width of the grid.
5. After completing the nested iterations of (*x*,*y*) over the whole grid, update *baseBeat* to the beat after the last generated MIDI note.

Binary Mapping

3. Iterate *x* across the GL grid from left to right. For each of the values of *x*, convert the column at *x* into a binary number. The Least Significant Bit is at the top of the grid and Most Significant at the bottom. Dead cells convert to a 0 bit and live cells to a 1 bit.
4. For each of the iterated values of *x* generate a MIDI note of duration *D* with pitch proportional to the binary number modulo the pitch bounds (e.g. modulo 128 if using the whole MIDI note range). Locate the MIDI note at beat: *baseBeat*+ *D*+16*x*/*G*
5. After completing the iterations of *x* over the whole grid, update *baseBeat* to the beat after the last generated MIDI note.

The reason for using a modulo calculation in step 4 in the binary mapping is that it prevents a large number of low MIDI pitch values. Unfortunately a pure binary mapping will often lead to a composition where few pitches are used and where they are all very low pitches. This is because mapping the binary number onto a proportional pitch involves dividing by a very large scaling binary value (of the order of 2^{20}).

4 Evaluating Mappings

As explained earlier, a key element of this paper is to demonstrate how focusing on the success of a mapping can be used to control the development of a generative musical process. To demonstrate this we need at least the following:

1. A set of GL initial states (“seeds”) which are used to generate music for evaluation
2. A set of mappings for evaluation
3. A methodology for relative evaluation of the musical results of the generative process for each mapping and seed.

For each mapping we will generate music for every seed, and then attempt to evaluate the music. The mappings we will use are the radial, Cartesian and binary mappings already described. The seeds we will use are:

1. 21 cell row – common GL seed (symmetric)
2. 11 cell row – common GL seed (symmetric)
3. Exploder – common GL seed (symmetric)
4. Circle radius 9 – geometric figure (symmetric)
5. Square size 9 – geometric figure (symmetric)
6. Concentric circles, max 14 – circles embedded within each other, with a circle of dead cells in between each circle of live cells. Outer circle has radius 14 (semi-symmetric)
7. Concentric circles max 6 (semi-symmetric)
8. 4 gliders – Made up of 4 copies of a common GL seed, placed apart in lower left quadrant
9. random 5x5 density 33% - area at centre of grid size 5 by 5 cells, with approx 33% of cells alive in random positions.
10. random 15x15 density 33%
11. random 25x25 density 33%
12. random 25x25 density 15%
13. random 25x25 density 50%

The methodology used to evaluate the music generated from these seeds is now described in more detail.

4.1 Evaluating Generated Music

A measurement process was needed which could aid in the relative evaluation for the generated music. The chosen measurement process was based on two metrics: Zipf’s measure of music “pleasantness” [5], [14], and an entropy-based measure.

4.1.1 Zipf's Law

In 1949 Zipf refined a statistical technique which has become known as "Zipf's Law". It is designed to measure the scaling properties of natural, including human, phenomena. The studies in [5], [14] demonstrate the idea that some aspects of proportion in music, as it relates to aesthetics in music, can be captured using measurements based on Zipf's Law.

To explain Zipf's Law, let us take a particular type of musical event or measure. We will use the measure of frequency of *intervals between successive MIDI notes*. (This is very similar to the first metric used to explain Zipf's Law in [5], and will be demonstrated later to be the most useful out of three metrics considered here.) We will notate measure as $N_{mi}(I)$, where I is the number of semitones in an interval. To calculate Nmi(I), go through the MIDI notes in a piece and calculate the number of semitones between successive MIDI notes. Then count up the number of successive pairs of MIDI note intervals which contain I semitones. Let $N(r)$ be the value of $N_{mi}(I)$ for the r-th most common interval. So if the most common interval contains 3 semitones, and the second most common interval contains 1 semitone, then $N(1) = 3$ and $N(2) = 1$. For our purposes Zipf's Law in music can then be summarized as: *$N(r)$ is proportional to $1/r^n$* (where n is very close to 1). Although we have used the example of successive MIDI intervals, Zipf's Law has been proposed as being applicable to a wide range of musical properties in human generated music. The practical effect of this proportionality is that if we plot $\log(N(r))$ against $\log(r)$, we should get something approximating a straight line. So if we perform a linear regression, the R^2 should be close to 1.

So to summarize, for our purposes, Zipf's Law says that if we take a log-log frequency plot of *intervals between successive MIDI notes* and rank it by decreasing frequency, then perform a linear regression, we should get an R^2 close to 1.

4.1.3 Using Zipf's Law to measure Aesthetics

In previous work referenced above, a link has been drawn between the aesthetic or "pleasantness" of a piece, and how close R^2 is to 1. However evaluation of a piece of music by a single numerical measure is a complex matter, and needs to be carefully qualified. [5], [14] support the idea that Zipf's Law may represent certain necessary but *not sufficient* conditions for aesthetically pleasing music. Furthermore, it is stated that "simple metrics could be easily fooled in the context of, say computer-aided music composition, where such metrics could be used for fitness evaluation." Also the Zipf measure has only so far been applied to very conventional, predictable, and conservative "culturally sanctioned" music, using the very limited representation medium (MIDI) not often experienced by real music listeners (most people do not listen to MIDI files), and discarding from aesthetic evaluation a wide range of musical parameters of potential significance (timbre, rhythm, timing, proportion, performance nuance, intonation, etc.) So the evaluation of music using Zipf's Law does not guarantee aesthetic superiority. But there is enough evidence to support its use as part of a combined relative measurement process. It gives a comparative numerical measurement of the relative fulfillment of sufficient conditions between different pieces of music.

On that basis, as one part of comparatively evaluating the radial mapping, it was decided to test for Zipf's law in the musical output of radial mapped GL. The values of R^2 generated from this data would be compared to values of R^2 generated for a pair of linear mappings – a Cartesian and a Binary mapping. The hypothesis is that the Zipf's Law R^2 values for the music generated by the radial mapping would be higher than by the non-radial mapping.

Three possible metrics were considered for measuring their R^2:

1. Distribution of pitch intervals between successive MIDI notes (N_{mi})
2. Distribution of the 12 pitch classes (*DPC*)
3. Distribution of the 128 MIDI pitches (*DMP*)

Previous papers have included metrics which are the result of duration and melodic/harmonic analysis. However the algorithm described in Section 3.1 generates notes of a fixed duration. Also the music analysed in previous papers has been human generated and is more susceptible to melodic/harmonic analysis. So we have limited ourselves to the above subset of metrics.

An experiment was run to find the R^2 values of these for Chopin's Revolutionary Etude, Op. 10 No. 12. A similar experiment was run using the radial, Cartesian and binary algorithms but randomly selected whether a GL cell was on or off during the process. This has the effect of essentially randomly generating music. The results are shown in table 1 (averaging over all randomly generated music).

Table 1. Comparison of R^2 values for Chopin and Random music.

Measure/ R^2	Chopin	Random
N_{mi}	0.88	0.54
DPC	0.83	0.76
DMP	0.75	0.83

It can be seen that the measure which most clearly differentiates the Chopin piece from Random music is the N_{mi}, and hence it became the Zipf-based metric chosen for this paper.

4.2 "Melodic" Entropy

It was considered to be important to also include some basic measure of "structurality" in evaluating an Alife musical mapping. This is because in theory complexity can be increased by simply generating a "chaotic" interface between the Alife process (GL) and the music. If the mapping and the Alife process are sufficiently at odds, then a great deal of chaotic complexity may be generated. However, a balance between structure and chaos is often considered an important part of music.

To measure how structured a piece is, we use Shannon's Information Entropy [15]. This measure can be interpreted as how much "structure" there is in a distribution. This is because the more structure there is in the distribution, the less data is required to represent it, and hence the lower its "information content". So one interpretation of a *higher* entropy value for a set of MIDI notes, is that it suggests a *less structured* piece of music in note to note terms. Once again we need to make the qualification that using such a metric to evaluate musical aesthetics can only be indicative. For example, Information Entropy measures in music are based on the assumption that music is ergodic. However by combining this measure with the Zipf measure discussed earlier, and using it only as a *relative* measure, we then have a useful indicator for relative evaluation.

We will focus on the Melodic Entropy, since the generated pieces can have extremely complex harmonic progressions, and so there is no simple way of measure harmonic entropy. However there is a heuristic way of approximating melodic entropy. This involves evaluating the "melody line" separately, as well as evaluating the full piece. The mappings to be demonstrated here do not explicitly generated melody lines. So to measure melodies the following approximation is used: the generated MIDI file is filtered so that for each tick, only the highest playing note is kept. Listening to such filtered files, it can be heard that this is the line of notes which will be most often associated with "melody" by a listener. These lines are then measured by taking the MIDI beat and pitch data for the "melody line" and generating a vector with one element for each note. Each element in the vector is the MIDI pitch data for the note multiplied by its beat location. The Shannon Entropy of this vector was then calculated. This approach is to ensure the melodic entropy takes into account timing as well pitch value.

5 Results

Each of the 13 seeds is allowed to grow for 15 generations (approximately 1.5 minutes of music), and music is generated using all three mappings: Radial, Cartesian and Binary. The average R^2 was calculated for each and the results are shown in table 2. We have included in these results the Zipf R^2 for the approximated melodies as well (using our melody stripping algorithm).

Table 2. R^2 Results of the various GL mappings for different initial seeds, 15 generations run.

Mapping	Zipf R^2	Melodic Entropy	"Melodic" Zipf R^2
Radial	0.9	6.8	0.91
Cartesian	0.84	7.6	0.88
Binary	0.7	7.6	0.7
Random	0.55	9.7	0.54

5.1 Analysis

The first thing to note in Table 2 is the close correlation between Zipf R^2 and the approximated "melodic" Zipf R^2. This supports the idea that the heuristic melody stripping algorithm is in fact capturing an important aesthetic of the music. And given the nature of this algorithm's output (a monophonic stream of the highest notes), it suggests that the algorithm may be a reasonable approximation of melody. Also in table 2 it can be seen that the lowest Zipf R^2 is given by the random music, as would be expected. Similarly the random music gives the highest melodic entropy, which is compatible with it having the least note "structure", also as would be expected. It can also be seen the highest R^2 and melodic R^2 occur for the Radial mapping, supporting the idea of using polar co-ordinates in an ALife system with radial symmetry. This is further underlined by the fact the Radial mapping has the lowest melodic entropy. The linear mappings (cartesian and binary) are both better than random, but still have higher entropy and Zipf R^2 than the radial mapping.

It was decided to examine the effects of the life evolution on the Zipf R^2 value, to increase confidence that the Zipf R^2 value was a result of the mapping. To do this two measures were defined: the Average Density of Cells (ADC) and the Average Change of Cells (ACC). For a particular seed and mapping, the ADC is simply the average percentage of cells in the grid which were alive over the whole 15 iterations of the game. The ACC is a similar measure over all 15 iterations but measured the absolute difference in number of cells alive between generation i and generation i-1. This difference is averaged over all 15 iterations to give the ACC. A regression was run between the ACC and the Zipf R^2 for each seed/mapping combination. A similar regression was run for ACC vs. the Zipf R^2s. The R^2 for these two regressions were 0.0002 for ADC and 0.0001 for ACC. This regression was also run against the melodic entropy, and the R^2 values were 0.27 for ACC and 0.0002 for ADC. These results support the idea that the Zipf R^2 and entropy values were determined more by the mappings than by the nature of the evolutions.

In spite of this it is also worth noting that if the average Zipf R^2 is found for each initial seed (across all mappings, but not including the randomly generated tunes) we get the following results in Table 3. The top 4 results (i.e. with the highest Zipf R^2) are the four of the *most symmetric* seeds. The bottom 5 results are five of the *least symmetric* seeds, with the lowest Zipf R^2. These R^2 values of the mappings support the importance of considering symmetry when choosing a mapping for the creation of music from GL.

It is worth noting that the radial mapping discussed here is actually an approximation, in the sense that the "radial aesthetic" of GL, upon closer examination, is actually a *local* phenomenon. It is quite common for GL seeds to eventually generate a number of islands of activity across the GL grid. Each island of activity often has a local symmetrical characteristic in itself (due to the radial symmetry of the GL rules). But there is not always a "grid-wide" or global symmetry in the behaviour of the whole board. It is interesting to note that when designing the Radial Mapping approach, one of the approaches considered by the authors was a mapping made up of multiple local

Radial Mappings across the board. Each local radial mapping would have its own polar co-ordinate origin, and methods were considered to self-organise these origins into appropriate positions. However it was felt that to take such a complex step without first evaluating the basic radial approach may add too much complexity at too early a stage.

Table 3. R^2 Results of the various GL seeds.

Seed	Zipf R^2
Concentric circles max radius 6	0.815
Circle radius 9	0.7975
11 cell row	0.765
21 cell row	0.755
25x25 random, density 15%	0.74
Square side 9	0.7225
Concentric circles maximum radius 14	0.7025
Large Exploder	0.6975
4 gliders	0.6975
25x25 random, density 33%	0.685
5x5 random, density 33%	0.685
15x15 random, density 33%	0.6725
25x25 random, density 50%	0.6675

6 Conclusions and Further Work

In this paper we have discussed the importance of choosing the right mapping for music composition generated from underlying emergent and ALife algorithms. To illustrate these issues we used one of the most basic emergent ALife systems - Cellular Automata, specifically Conway's Game of Life (GL). A simple mapping for GL was introduced, based on a polar co-ordinate system. We argued it was superior to previous GL mappings when both its simplicity and its visual-aesthetic capture was considered (since the visual symmetry in GL is a result of radial symmetry in its rule set). This mapping was evaluated by comparing it to linear mappings using Zipf's Law and using a measure of melodic entropy. Based on certain limiting assumptions about the application of Zipf's Law and Shannon's Information Entropy to music – the evaluation supported the idea that the radial mapping provided some improvement. Thus we have demonstrated the process of looking at the effects that changing the mapping can have on the output of a generative compositional process in emergent systems, and shown ways of investigating the effects of various mappings, and basic ways of relatively evaluating those mappings.

Further work could include revisiting the set of measures. For example the melodic stripping algorithm, and the structurality measure were very basic and could be improved. Also the number of elements against which a Zipf R^2 value was calculated could be extended from N_{mi} to include others which measure further important musical factors such as rhythm. Looking at the mappings themselves, the issue of non-local and multiple mappings could be addressed. For example having multiple polar co-ordinate origins to capture local symmetry. And further investigations could be done into perhaps dynamically adjusting mappings based on various objective measures – for example the ADC and ACC are actually part of the visual experience of GL, so could be incorporated into an extended mapping system. These are higher order mappings, and thus capture higher order experiences of GL. Overall it is hoped that this work has highlighted the importance of selecting and evaluating the correct mapping for turning a complexity-generating process into a musical process.

References

1. Beyls, P.: Cellular Automata Mapping Procedures. Proceedings of the International Computer Music Conference, Miami USA (2004)
2. Miranda, E., R.: Composing Music with Computers. Focal Press Oxford (2001)
3. Miranda, E.,R.: Cellular Automata Music Investigation, MSc. in Music Technology final project report. University of York UK (1990)
4. Burraston, D., Edmonds, E., Livingstone, D. and Miranda, E. R.: Cellular Automata in MIDI based Computer Music. In: Proceedings of the International Computer Music Conference, Miami USA (2004)
5. Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., Davis, R.B.: Zipf's Law, Music Classification, and Aesthetics. Computer Music Journal 29:1 (2005) 55-69
6. Wilson, G.: The Life and Times of Cellular Automata. New Scientist October (1988) 44-47
7. Hoffman, P.: Towards and Automated Art: Algorithmic Processes in Xenakis' Compositions. Contemporary Music Review 21 (2002) 121-131
8. Beyls, P.: The Musical Universe of Cellular Automata. Proceedings of the International Computer Music Conference (1989)
9. Millen, D.: Cellular Automata Music. Proceedings of the International Computer Music Conference (1990)
10. Cope, D.: Computer Models of Creativity. MIT Cambridge (2005)
11. Burraston, D., Edmonds, E.: Cellular Automata in Generative Electronic Music and Sonic Art: A Historical and Technical Review. Digital Creativity 16 (2005) 165-185
12. Wolfram, S.: A New Kind of Science. Wolfram Media Inc, USA (2002)
13. Reiners, P.: Cellular automata and music. developerWorks, IBM (2004)
14. Manaris, B., Machado, P., McCauley, C., Romero, J., Krehbiel, D.: Evolutionary Music and the Zipf-Mandelbrot Law: Developing Fitness Functions for Pleasant Music. In: Lecture notes in computer science. Springer Berlin (2003) 522-534
15. Shannon, C.,E.,: A Mathematical Theory of Communication. The Bell Technical System Journal 27 (1948) 379-423, 623-656