Francis Soddell

Jacques Soddell*

La Trobe University, Bendigo, Australia

f.soddell@latrobe.edu.au

* cajid media
28 Wirth St.,
Bendigo, 3550
Australia

jacques@cajid.com

Of Lindenmayer systems, Fungi and Music

Abstract

Lindenmayer systems are well known in computer science as a way of generating realistic images of plant-like structures such as trees, bushes and flowers. They have also been used to produce generative art and for algorithmic composition. Extending our work with modelling fungal growth using L-systems, we designed a MIDI file mapping suitable for interpreting these fungal models musically as well as graphically. This paper describes some of the problems we encountered in developing our MIDI mapping for L-systems and how we applied this mapping to create sound installations of plant and fungal models for a sculpture exhibition in an environment dominated by plants.

Introduction

Computer science borrows concepts, such as genetic algorithms, from nature to construct environments suitable for exploring virtual reality, artificial life, and complex systems. Biology makes use of such constructed environments, perhaps by manipulating data within them, to gain greater insight into the natural world. Musicians may borrow from both sciences and Supper (2001) suggested that those who apply extra-musical algorithms to simulate natural phenomena may secretly see algorithmic composition as a way of generating forms that justify themselves by their naturalness alone. He also pointed out that one algorithmic method, L-systems, is popular because it generates self-similar structures (Supper 2001). L-systems, which are generative systems in which complexity emerges from a simple set of rules, have been used to generate plant-like images, capturing the recursive nature of branching organisms such as trees, bushes, and filamentous microbes, all of which exhibit some self-similarity, reminiscent of fractals. They have aided investigations into artificial life because of their ability to represent the intrinsic nature of an organism, although they are not easily able to model environmental influences, which are better modeled with cellular automata. Prusinkiewicz (1986) first discussed score generation using L-systems, Mason and Saffle (1994) used them to produce relatively complex scores, and Nelson (1996) used L-systems to compose “Summer Song”.

In 1994 we investigated L-systems and found them useful for helping determine how fungal colonies may become circular (Soddell et al. 1994), we also found them useful for showing that the sonification of the fungal data we had collected conveyed more about the comparative rate of growing organisms than the visualization of that data (Soddell and Soddell 2000). In doing so, we discovered innate musical characteristics that we further explored to create compositions for a web-based sound installation, *autonomousAudio*, exhibited in 2000 at Artspace Gallery in Sydney (http://www.artspace.org.au/2000/autonomousaudio/).

In the following paper, we describe L-systems, how we mapped to MIDI and adapted existing L-systems to benefit from this mapping, describe how we developed systems from data of the growth of the fungus *Geotrichum*, and how we used the resulting MIDI files in the *Art House* installation presented in the Conservatory (a glasshouse with plant display) in Rosalind Park in Bendigo, during Easter 2005.

L-systems

An L-system is a string rewriting mechanism, which has a finite set of symbols (alphabet), a start symbol or string of symbols (axiom), and a finite set of rules (productions). Since Lindenmayer, a biologist who worked with fungi, first introduced them in 1968 (Lindenmayer 1968a; 1968b), they have been popular for modeling the growth and development of multicellular organisms. Smith (1984) proposed using them to create computer generated images of plants, and Prusinkiewicz and Lindenmayer (1990) created systems that produce realistic images of trees, bushes, and flowers. In a biological context, the axiom represents the start state of a growing organism. First the rules are applied in parallel to the axiom (generation zero), with the resulting string being the first generation, and then to successive strings to produce generation after generation. For example, the L-system

```
START: F-F-F-F-
p1: F -> F-F+F-F-F
```

taken from *The Algorithmic Beauty of Plants* (Prusinkiewicz and Lindenmayer 1990), has a start string F-F-F-F- and one production rule specifying that each F be replaced by F-F+F-F-F. Since there are no rules specifying otherwise, the + and – symbols are reproduced without change, so that the first generation becomes F-F+F-F-F-F-F+F-F-F-F-F+F-F-F-F-F+F-F-F-. For the second generation, replace every F in the first generation as specified by the rule p1, and so on for successive generations. In this example, the string will become bigger and bigger and the images resulting from them become more fractal in nature.

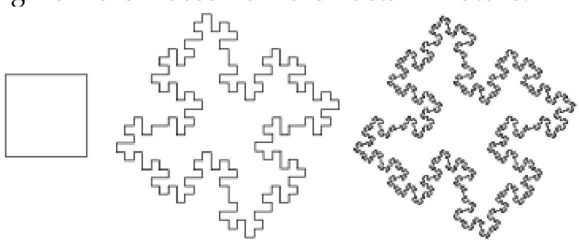

Figure 14. Quadratic Koch curves resulting from L-system above, generations 0, 2, 3.

The strings resulting from this system may be interpreted in different meaningful ways, with the turtle graphics interpretation being the most popular for generating images. In turtle graphics the symbol F is interpreted as an instruction to move forward by one step drawing a line, + means turn left, and – means turn right, and Figure 1 shows the images for generation 0, 2, and 3.

Since the L-system specifies neither the length of the line to be drawn by the turtle nor the size turning angles, these revert to default values supplied by the turtle graphics interpreter, 90^{O} being the default size of the turning angle. Different lengths between turns are managed by the number of Fs in a row – for example a step of FF will be twice as long as that specified by a single F. Other types of L-systems were introduced to address various issues. Parametric L-systems associate numeric parameters with symbols as can be seen in the in the L-system,

```
#define step 0.25
#define delta 10
START: F(step)b
p1: b ->   (0.334)[+(delta)F(step)b]
                  [-(delta)F(step)b]
      ->
(0.333)[+(delta*2)F(step)b]
                  [-
(delta*2)F(step)b]
      ->
(0.333)[+(delta*3)F(step)b]
                  [-
(delta*3)F(step)b]
```

which specifies that each F should be of length 0.25 and angles a multiple of 10^{O}, so the L-system itself controls more of the features of the organism being modeled.

Bracketed systems, introduced to represent the growth and development of branching structures (Lindenmayer 1968a; 1968b), use a left-bracket delimeter, [, to signal the beginning of a string of symbols representing a branch, and a right delimiter,], to signal a return to the branching filament, as shown above). Sub-branches and sub-sub-branches are represented by strings with nested brackets. Deterministic systems generate a single infinite derivation sequence because only one rule applies to any one symbol at a given time. However, stochastic systems, as seen above, may specify a number of different rules for replacing a given symbol and the probability distribution to be used by the interpreter when choosing which rule to apply. The system above demonstrates these features: each F is of length 0.25 and angles are a multiple of 10, every step branches to the right and left, and there is an almost one in three chance that the branching angle will be 10^{O}, 20^{O}, or 30^{O}.

For further details concerning parametric, probabilistic and branching L-systems, see Pruskinkiewicz and Lindenmayer (1990).

Generating MIDI from L-systems

Prusinkiewicz (1986) discussed score generation using L-systems, implemented the control of pitch and duration, and proposed how they may be used to control tempo and volume. Prusinkiewicz and Hanan (1989) described how an L-system used to generate a fractal curve consisting of horizontal and vertical segments can be interpreted as a sequence of notes if the y-coordinate of a line segment is interpreted as pitch and the length is interpreted as duration.

The turtle sonics MIDI interpretation, based on the concept of turtle graphics as presented by Prusinkiewicz and Lindenmayer (1990), is shown in Table 1. It was devised to suit the simple developmental pattern of fungal hyphae (Soddell and Soddell 2000), which grow across surfaces creating an essentially 2D circular branching structure, as seen in the third image of Figure 4. Like turtle graphics, this interpretation executes a depth-first traversal of the data structure representing the branching organism (each branch is traversed to its end before the next branch is visited), resulting in a monophonic sequence of notes, rather than using branching points to generate polyphonic sequences. We related changes in direction of growth of microbes (as determined by branching angles) to changes in pitch, and distances between branches to note duration. Thus, a fungus with sparse branching produced long notes, and hyphae that changed direction often produced greater pitch changes. Like Mason and Saffle (1994), we found that the pitch went out of range – MIDI, hearing and instrument range. Unlike them, we could not restrict our interpretation to images within audible range because we wished to explore the usefulness of sonifying them. Therefore, we adopted the idea of a circular keyboard, as described below, by modifying a software program lsys2midi, previously written by students under our supervision at La Trobe University, Bendigo. The modified lsys2midi interprets parametric L-system symbols as shown in Table 1. For MIDI files, available pitches range from 0 to 127, where the interval between each pitch is a semitone. Like McCormack (1996), we interpreted + and – as instructions to increment and decrement pitch by one semitone, except that, in our interpretation, when the pitch reaches a maximum the next increment forces the pitch back through 0 and when it reaches a minimum the next decrement sets it back through 127. So we consider a pitch of 0 equivalent to that of 128 and picture this in terms of a circular keyboard, which resembles the turtle interpretation of the angles where 360^{O} coincides with 0^{O}.

Users of lsys2midi may specify the lowest, highest, and the starting pitch, so that the same string can be interpreted over different ranges, which is very useful for simulating different musical instruments. For this project all our runs specified MIDI notes of 25, 95, and 60 (middle C).

Symbol	Graphics (lsys)	MIDI (lsys2midi)
F(x)	pen down, move forward x	note down, hold for time x (play)
f(x)	pen up, move forward x	note up, hold for time x (rest)
+(x)	change direction left by x^{O}	increase pitch by x
-(x)	change direction right by x^{O}	decrease pitch by x
Angles of 360^{O} and 0^{O} are equivalent. Similarly, maximum and minimum pitches are equivalent.		

Table 2. Turtle Graphics and Turtle Sonics

As shown in Table 1, we interpreted the x parameter associated with F as both the length of a line and the duration of a note, which resembles Prusinkiewicz and Hanan's suggestion that the length of a line segment may represent the duration of a note (Prusinkiewicz and Hanan 1989). However, this may cause problems especially when working with existing L-systems that have been developed with the turtle graphics interpretation in mind. The appearance of a line that has been created by the turtle graphics instruction F(1)F(1)F(1) (or FFF) and one created by F(3) , or by F(2)F(1) will be the same – they will all be drawn as a line three units long – but the MIDI files generated by the turtle sonics interpretation will sound quite different. The first two strings will sound as three notes one after another, whereas F(3) will sound as one note with three times the duration, and the last string will sound as two notes, the first twice as long as the second. So there is more to consider when designing systems for MIDI interpretation, and the next section discusses how we adapted the L-system for a plant-like structure so that it is suitable for the turtle sonics interpretation.

Adapting L-systems to generate music

Since the installation was to be presented in the Conservatory, we wished to create music inspired by nature so we chose the plant-like L-system from Prusinkiewicz and Lindenmayer (1990) as the basis for one piece.

```
#define maxgen 7
#define delta 25.7
START: X
p1: X ->F[+X][-X]FX
p2: F ->FF
```

The image generated from the turtle graphics interpretation can be seen in Figure 2, note that it is generation seven and that the branching angles are all 25.7^{O} as defined by the L-system. Although this system can be used to generate music using our MIDI interpreter, it results in a large number of short notes sounding one after the other, which does not accurately reflect the nature of the plant.

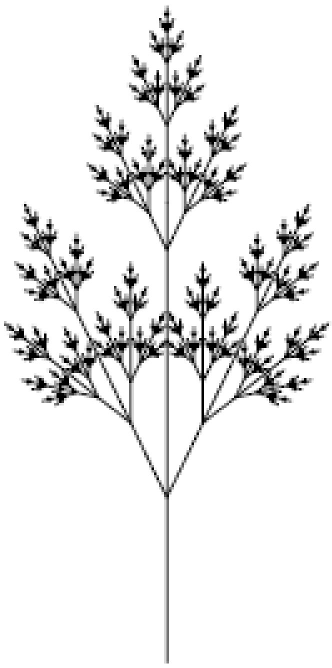

Figure 15. Plant-like structure generated, using the turtle graphics interpretation, from the previous L-system (Prusinkiewicz and Lindenmayer 1990)

Therefore, we developed the following parametric adaptation, which specifies the length (duration) of a line (note) via the parameter associated with the symbol F.

```
#define maxgen 7
#define delta 25.7
#define step 0.125
#define maxStep 3

START :X
p1: X -> F(step)[+(delta)X][-
(delta)X]
         F(step)X
p2: F(x):x<=maxStep -> F(x*2)
p3: F(x)->F(x)F(x)
```

The length of each step is defined by the L-system, rather than being left up to the interpreter, and can be changed for different runs (generating slower or faster pieces). Parametric systems are not only useful for associating different values with certain symbols but also for controlling which production rule is to be applied, depending on the value of the parameter. We decided that the notes in this composition required a maximum duration. Rule p2 controls this by doubling the value of parameter associated with F only if the value is less than or equal to a maximum duration of 3, otherwise rule p3, which is the same as rule p2 of the original L-system, is applied. The image generated by the turtle graphics interpreter for this L-system is the same as that in Figure 2.

The MIDI file generated by this system resulted in a musical piece that complemented the sculptures of birds by Yvonne George. These were exhibited in one room of the conservatory, where we also set up one installation based on the plant-like structure in Figure 2, using a simple stereo speaker system and one organ-like instrument. To complete the installation, we turned to new data we had just finished collecting for a filamentous fungus.

Music of the fungus *Geotrichum*

One major difference between our work and that of others is that the L-systems we generate to produce music are based on empirical data of the growth of a living organism, rather than on theoretical structures or fractals. Thus we have incorporated real-life data into an A-life technique.

Geotrichum, a filamentous fungus commonly found in the environment, including soil, grows by extending a filament from a single spore and by branching, eventually forming a circular colony of branching filaments. We conducted growth experiments on *Geotrichum*, saving digital images of early stages of growth, such as that in Figure 3, from which we later collected measurements using image analysis software.

We previously found that bracketed, stochastic, parametric L-systems were well-suited for entering empirical data collected from the fungus

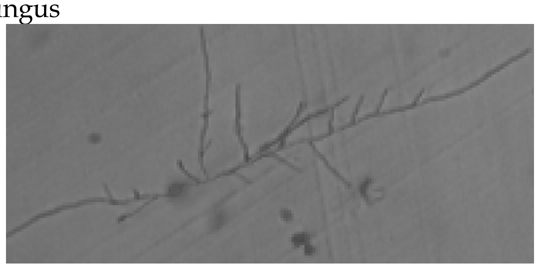

Figure 16. Microscopic image of early stage of colony growth of the fungus Geotrichum.

Mucor M41 (Soddell et al. 1994; Soddell and Soddell 2000), so we used similar systems to record the frequencies and sizes of distances between branches, distances between changes of direction of growing filaments, observed branching angles and their frequencies, observed angles of change of direction and their frequencies. The system required 15 rules, five of which are stochastic and so have a number of different alternatives (up to 50). For example, rule p15 below recorded the observed lengths of growing tips of branches and their frequencies.

```
p15:T ->(0.08333)F(4*step)
      ->(0.16667)F(5*step)
      ->(0.16667)F(6*step)
      ->(0.08333)F(8*step)
      ->(0.08333)F(9*step)
      ->(0.08333)F(11*step)
      ->(0.08333)F(12*step)
```

```
        ->(0.08333)F(16*step)
        ->(0.08333)F(22*step)
        ->(0.08333)F(23*step)
```

We had not recorded tip lengths in our previous work with *Mucor* M41 (Soddell and Soddell 2000), but found this important for musical composition as we sometimes wish to indicate the end of a branch and a return to the parent branch by means of a pause, which we first introduced in the axiom to ensure that there is a rest as the composition ends, indicated by f which is the symbol for pen up in turtle graphics.

```
START: H(0,0,-1,0,-1,0,0)
       f(pause*step*4)
```

There are seven parameters associated with the initial symbol H to keep track of the important events in the growth of the fungal hyphae, such as change in direction of a branch, branching, reaching the end of a branch. Their use is demonstrated in the rule below, which is used to record the frequency and length of interbranch distances (represented by the symbol b). The parameter g allows us to keep track of the number of generations (how old the fungus is).

```
p5:H(x,y,b,z,c,s,g):b=-1
     ->(0.07143)H(x,0,5,z,c,s,g+1)
     ->(0.07143)H(x,0,7,z,c,s,g+1)
     ->(0.07143)H(x,0,13,z,c,s,g+1)
     ->(0.07143)H(x,0,15,z,c,s,g+1)
     ->(0.07143)H(x,0,16,z,c,s,g+1)
     ->(0.07143)H(x,0,17,z,c,s,g+1)
     ->(0.14286)H(x,0,21,z,c,s,g+1)
     ->(0.14286)H(x,0,24,z,c,s,g+1)
     ->(0.07143)H(x,0,25,z,c,s,g+1)
     ->(0.07143)H(x,0,32,z,c,s,g+1)
     ->(0.07143)H(x,0,44,z,c,s,g+1)
     ->(0.07143)H(x,0,62,z,c,s,g+1)
```

The following rule makes use of the value of g because, now that the fungus is completing its growth, it is time to pay attention to symbols that are important for both the turtle graphics and the turtle sonics interpretations.

```
p4:H(x,y,b,z,c,s,g):g=(maxgen-2)&x>0
     ->G(x)
```

Finally, we substitute the F for G.

```
p2:G(x)->F(step*x)
```

The full system, with example output from runs can be found at http://ironbark.bendigo.latrobe.edu.au/~fran/microbes/geo.html. Since the L-system rules are stochastic, every run represents a unique organism as can be seen in Figure 4, which shows three images of differing branching frequency although they are all the same generation (150).

Figure 17. Three runs of L-system for Geotrichum

The stochastic nature of the growth of the fungus, as recorded in the L-system, is reflected in the music produced and this is particularly apparent in the organisms with greater number of branches. That is, as more variations are introduced the music becomes more complex allowing for more interesting abstract soundscapes. This is particularly apparent when compared to the more regular branching structures of plants and trees.

We used the MIDI file generated by one of these images of a colony of *Geotrichum* (the 3rd image) as the basis of a 4-channel sound installation during the "Art House" sculpture exhibition held in Bendigo during the Easter period 2005. Four speakers were placed in the 4 corners of one room of the conservatory in Rosalind Park in Bendigo (the conservatory was set up into 3 sections or "rooms" to accommodate work by 3 sculptors). This room was filled with plants and a series of abstract sculptures by local artist Trefor Prest. For the installation we created a prerecorded composition based on six instruments playing the same MIDI file simultaneously. Using this MIDI file, we made six recordings (tracks) each using a different instrument created in the software synthesisers Absynth, FM7 and Pro-53 (Native Instruments). These instruments were selected for their different timbral and envelope characteristics. Four of these tracks were allocated to their own channel (one per channel). The remaining two tracks were placed at various (moving) positions within the room. The volumes of all tracks were then adjusted relative to each other, so that not all were playing at the same time. Thus the MIDI file generated the sequence of the notes played, but we made aesthetic decisions about how, when and where the different instruments were used.

In another room, containing bird-like sculptures by Yvonne George, we presented organ-like music generated by the MIDI file associated with the plant-like structure in Figure 2. The L-system of this plant-like structure generated light and airy sounds, reflecting the plant growth environment, while the fungal sounds were much darker and more abstract. Stereo versions of the 2 installations can be found at http://cajid.com/jacques/lsys/plant.mp3 and http://cajid.com/jacques/lsys/geotrichum.mp3 .

Conclusion

Algorithmic composition methods are widely used and L-systems have become popular as a means of generating music from fractals. We have used measurements of growth of a living fungus as the basis for production of MIDI files and have attempted to relate the MIDI mapping to take account of the circular growth patterns of fungi. Using this mapping, we found that the L-system approach led to musical compositions that were uniquely suited for an installation in a plant environment.

We would like to further explore the branching nature of fungal organisms in different ways. For example, Prusinkiewicz (1986) proposed that branching L-systems may be used to create polyphonic scores by assigning simultaneous musical interpretations to branches. Until now, because of the many branches and sub-branches in most of our organisms (we would rapidly run out of polyphony on contemporary synthesisers), we have not pursued this idea but now wish to apply it to those that are in very early stages of growth. In keeping with this approach, we also wish to interpret the strings generated by our L-systems in a way that is more in keeping with the temporal growth of the organisms. Strings generated by L-systems are linear arrays of symbols with branches nested in brackets and our interpreter scans the symbols from left to right so that each branch with its sub-branches is fully sonified before the next branch is encounters. To solve this problem we are looking at ways of rearranging the strings output by the L-systems so that the symbols are in order of growth before they are scanned by the interpreter.

References

Lindenmayer, A. 1968a. “Mathematical models for cellular interactions in development I. Filaments with one-sided inputs.” *J. Theoret. Biol.*, 18: 280-299

Lindenmayer, A. 1968b, “Mathematical models for cellular interactions in development II. Simple branching filaments.” *J. Theoret. Biol.*, 18: 300-315

Mason, S. and Saffle, M. 1994. “L-systems, melodies and musical structure.” *Leonardo Music Journal*, 4: 31-38

McCormack, J. 1996. “Grammar-based music composition.” *Complexity International*, 3. http://www.csu.edu.au/ci/vol3/mccorm/mccorm.html

Nelson, G. 1996. “Real time transformation of musical material with fractal algorithms.” *Computers Math. Applic.* 32, 1: 109-116

Prusinkiewicz, P. 1986. “Score Generations with l-systems”. In: International Computer Music Conference: Proceedings of the International Computer Music Conferece 1986. 445-557

Prusinkiewicz, P. and Hanan, J. 1989. *Lindenmayer systems, fractals, and plants.* Lecture Notes in Biomathematics. Berlin: Springer-Verlag

Prusinkiewicz, P. and Lindenmayer, A. 1990. *The Algorithmic Beauty of Plants.* New York: Springer-Verlag

Smith, A. 1984. “Plant, fractals and formal languages. *Comp. Graph.*, 18 (July): 1-10

Soddell, F., Seviour, R., Soddell, J. 1994. “Using Lindenmayer systems to investigate how filamentous fungi may produce round colonies.” In: Complex Systems: Mechanisms of Adaptation. eds. Stonier, R.J., XingHuo Yu. Amsterdam: IOS Press 61-68

Soddell, F. and Soddell, J. 2000. “Microbes and Music.” In: PRICAI 2000 Topics in Artificial Intelligence: 6th Pacific Rim International Conference on Artificial Intelligence Melbourne, Australia, August 28-September 1, 2000 Proceedings. (Lecture Notes in Artificial Intelligence 1886, LNCS). eds. Mizoguchi, R., Slaney, J. Berlin, Heidelberg: Springer-Verlag 767-777

Supper, M. 2001. “A few remarks on algorithmic Composition. *Computer Music Journal*, 25, No.1: 48-53