Extending Grammatical Evolution to Evolve Digital Surfaces with Genr8

Martin Hemberg martin.hemberg@imperial.ac.uk
Una-May O'Reilly unamay@ai.mit.edu

Abstract

Genr8 is a surface design tool for architects. It uses a grammar-based generative growth model that produces surfaces with an organic quality. Grammatical Evolution is used to help the designer search the universe of possible surfaces. We describe how we have extended Grammatical Evolution, in a general manner, in order to handle the grammar used by Genr8.

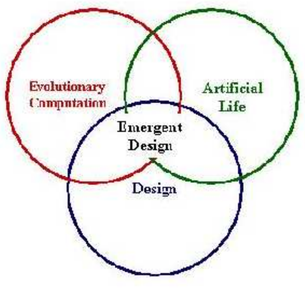

Figure 1: The field of Emergent Design is the intersection of Evolutionary Computation, Artificial Life and Design. By exploiting ideas from EC and ALife, we try to find novel approaches to design.

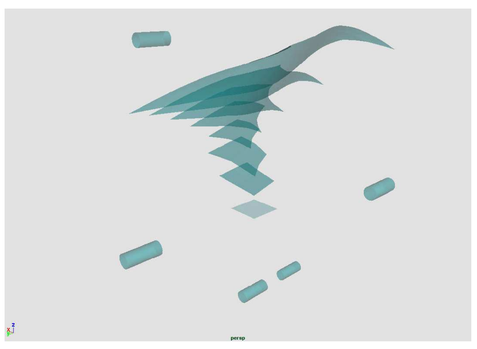

Figure 2: A time series capturing 7 growth steps of a HEMLS surface in an environment with five repellors (here drawn as cylinders). The smallest surface is the axiom. The largest surface is the final growth step. Had the repellors been absent, a flat square would have been formed.

```
N = { L-System, Axiom, RewriteRule, Predecessor, Successor,
      Modifier, AngleValue, BranchAngleValue }
T = { +, -, &, ^, \, /, ~, [, ], <, >, ->, Edge, Angle, Sync,
      EdgeX, BranchAngle }
S = { <L-System> }
P = {
<L-System> ::= <Axiom> <RewriteRule> { <RewriteRule> } Angle
               AngleValue [ Sync ] BranchAngle
               BranchAngleValue
<Axiom> ::= <Edge> [ ~ ] + <Edge> [ ~ ] + <Edge>
            { [ ~ ] + <Edge> }
<RewriteRule> ::= <Predecessor> -> <Successor>
<Successor> ::= { <Modifier> } <Edge>
<Predecessor> ::= <Edge> { <Edge> } |
                  <Edge> ''<'' <Edge> |
                  <Edge> ''>'' <Edge> |
                  <Edge> ''<'' <Edge> ''>'' <Edge>
<Modifier> ::= { <Edge> } |
               + <Modifier> - |
               - <Modifier> + |
               & <Modifier> ^ |
               ^ <Modifier> & |
               \ <Modifier> / |
               / <Modifier> \ |
               ~ <Modifier> |
               <Edge> ''['' ''['' + <EdgeX> '']'' - <EdgeX>
               '']'' <Edge>
               <Edge> ''['' ''['' + + <EdgeX> '']'' - -
               <EdgeX> '']'' <Edge>
<AngleValue> ::= 30 | 45
<BranchAngleValue> ::= 15 | 30 | 45 | 60 | 75 }
```

Figure 3: A HEMLS growth language expressed in Backus Naur Form (BNF). Any derivation of this BNF produces a HEMLS (i.e. a grammar). The terminals are interpreted to form a surface using *turtle graphics*.

Figure 4: A physical model made by Jordi Truco at the Architectural Association in London. The design process started with Genr8 and the evolved digital surface was subsequently exported so that a physical model could be manufactured.

Growing Digital Surfaces in a Simulated Physical Environment

Genr8 simulates organic growth and creates digital surfaces [1]. Its growth process is reactive: it takes place in a simulated physical environment. Surface growth is generated by a HEMLS - a Hemberg Extended Map L-System. A HEMLS is a more complex version of the widely known Lindenmayer System (L-system) [3]. An L-system is a *grammar*, consisting of a *seed* (or axiom) and a set of *production rules*, plus a *rewrite* process in which productions rules are applied to the seed and its successive states.

The most important feature incorporated into a HEMLS is how the physics of our simulated environment is factored into the rewriting of an edge. With this additional set of environmental factors, the surface growth mimics *tropism*, the response of an organism to external stimuli. Genr8 allows the user to specify the environment using attractors, repellors, gravity and boundaries as elements. Figure 2 shows how a HEMLS surface's growth steps in response to five repellors.

Why is Evolutionary Computation Useful in a Design Tool?

- It automatically generates a HEMLS with correct syntax; creating meaningful HEMLS by hand is a hard task.
- It adaptively explores the universe of surfaces following preferred surfaces yet yielding creatively different ones.
- It can be used to explicitize a surface an architect has in mind by allowing its search process to be influenced by the architect.

Why Use Grammatical Evolution?

Blind variation on a direct grammar representation would have to be either contrived to adhere to syntax constraints or it would result in offspring genotypes that have to be syntactically repaired or culled afterwards.

Grammatical Evolution (GE) [2] provides a means of mapping a fixed length integer sequence (which is its genotype and thus blindly crossed over or mutated) into a program via the use of a programming language specification given in Backus-Naur Form (BNF).

Extending Grammatical Evolution to Handle the Growth Language

The BNF for HEMLS contains symbols that have not been used in previous GE implementations. Moreover, we wanted to incorporate heuristics into the algorithm in order to facilitate the search for interesting designs.

Restrict the expansion of the program tree

If there are many non-terminals in the BNF's production rules, an expanded (or derived) HEMLS is likely to be very large. In general, such outcomes produce uninteresting surfaces. For instance, in the BNF of Figure 3, six of the nine productions of the non-terminal `<Modifier>` themselves expand to `<Modifier>`. A parameter called `max_depth` limits the maximum depth of the *expanded* syntax tree rather than limiting the mapping of the genotype.

Multiple and optional nodes

In a BNF representation of a language, the symbols $\{\ldots\}$ and $[\ldots]$ indicate that whatever symbols (terminals or non-terminals) appears between them is to be written zero or more times or zero or one times, respectively. Because of the importance of genetic inheritance (and the propogation of genetic characteristics) in a GE system, it is important that this optional quantity stay consistent through all decodings of a genotype (i.e., from one generation to the next or within multiple copies of the genotype in the population). The ideal way to achieve this consistency is to use the genotype itself to determine how many times a symbol is written when it is optional.

Genr8 as a tool

Genr8 is implemented as a plug-in to the 3D modeller Maya, which makes it easy and intuitive to use. Making the tool a plug-in is also extremly useful for the developer as the host software provides a lot of functionality "for free".

A Parameterized Fitness Function Governs the Selection

In Genr8 the fitness function is metaphorically a steering wheel that helps the user guide the evolutionary search. The fitness function is automated to allow a larger portion of the search space to be explored. Genr8 uses a parameterized fitness function relating six features of a digital surface. The user chooses a target value, t_i, and a weight, w_i, for each feature. In order to calculate fitness, each feature is assessed and compared with the user specified target values. The six features are called *smoothness*, *undulation*, *size*, *symmetry*, *soft boundaries* and *subdivisions*. Figure 7 and 8 show examples of how the fitness function can be used to produce surfaces with different charactersitcs.

Using Genr8 as a Design Tool

Genr8 has now been used for two years in a design course within the Emergent Design & Technologies graduate program at the Architectural Association (AA) in London. Figure 4, 5 and 9 are examples of designs that have been produced by students and staff with the aid of Genr8. Additional images are available on the World Wide Web at `http://www.ai.mit.edu/projects/emergentDesign/genr8`.

Summary

We have described how evolutionary computation, and in particular, extensions of grammatical evolution, empower a digital surface exploration tool named Genr8. Because of its responsive growth and evolutionary algorithm, Genr8's results are attractive, spontaneous and organic in appearance.

Figure 5: A rendered image of a design for a pneumatic strawberry/champagne bar. This is Genr8's largest design that has been physically actualized. By Achim Menges, AA, London, UK, 2003.

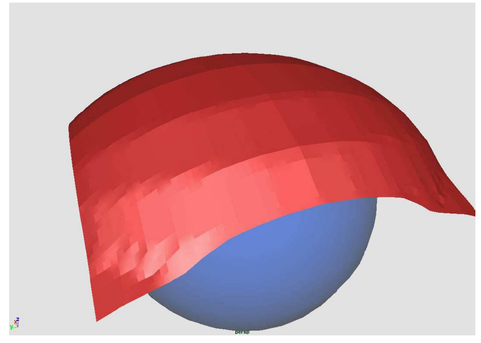

Figure 6: A Genr8 surface grown in an environment where it has been pulled down by gravity on top of a spherical boundary. The grammar is the same one as in Figure 6 and it is clear how the environment influences the final outcome of the growth process.

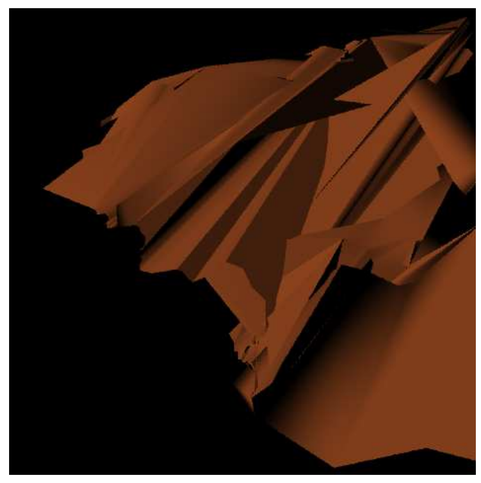

Figure 7: A rendered image of a surface evolved with Genr8. The fitness criteria has been set to select for highly articulated surfaces.

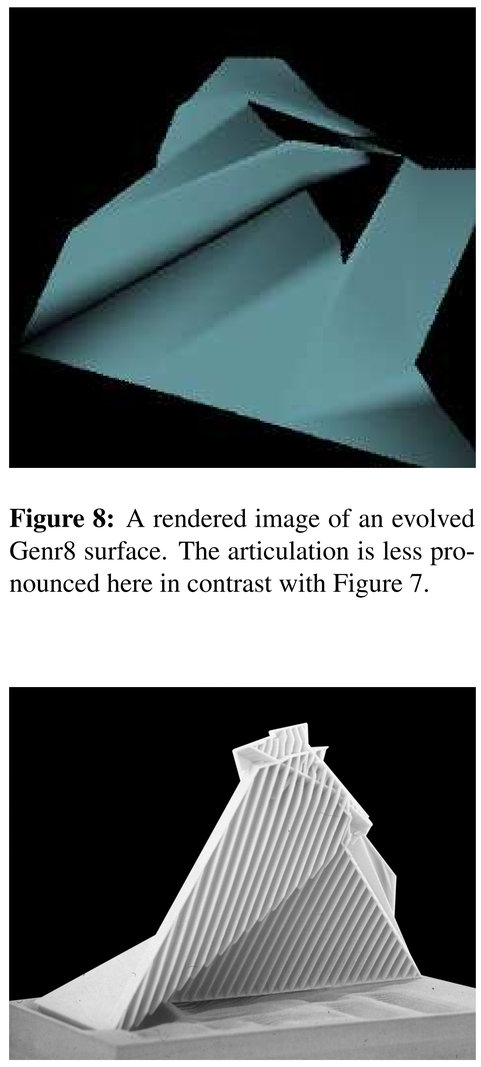

Figure 8: A rendered image of an evolved Genr8 surface. The articulation is less pronounced here in contrast with Figure 7.

Figure 9: A photograph of a physical model. The model was produced using a CNC mill and it is based on a Genr8 surface. By Linus Saavedra, AA, London, UK, 2003.

References

[1] Martin Hemberg. Genr8 - a design tool for surface generation. Master's thesis, Chalmers University of Technology, 2001.

[2] Michael O'Neill and Conor Ryan. *Grammatical Evolution - Evolving programs in an arbitrary language*. Kluwer Academic Publishers, 2003.

[3] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer, 1996.