18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Towards the simulation of three dimensional image forms using a graphical genetic algorithm

Fairall, R. G.[1], Herbert R.[1], Summons, P.[1]

[1] *Discipline of Information Technology, School of DCIT, Faculty of Science and Information Technology,*
THE UNIVERSITY OF NEWCASTLE
Email: ray.fairall@studentmail.newcastle.edu.au

Abstract: This paper examines some of the problems confronted in evolving a population of three dimensional computer generated shapes according to a set of desired constraints. The eventual aim is to automatically search a design or problem space for a viable solution or range of solutions to 3-D shapes. Such shapes are often used in generating computer images. Such images are needed in engineering and science and are useful in visualizing output from computer models. There may be applications within the domains of aesthetic design.

Previously, most of the successful attempts to evolve three-dimensional computer generated form have used systems of genetic encoding with a high degree of complexity in the link between the Genotype (the genetic endowment of the individual solution) and the Phenotype (the constellation of all available individual solutions), and have mapped the complete volume of the shape. An approach described is to encode the shape surface.

This has the potential to considerably simplify the mapping of the Phenotype to Genotype.
The new approach is inspired by the principle behind the operation of a “shape grabber”. In this a mathematical array consisting of a “point cloud” of Cartesian coordinates is created corresponding to those mapped on the surface of a real object. The former can then be used by a CAD program to plot a polygonal mesh and produce a 3-D representation of the original object.

The new approach uses a direction vector, the magnitude of which becomes a floating point” gene” within the Genetic algorithm. The endpoint of the latter defines a point within the point cloud.
An enhancement of this model could use an evolved Lindenmayer System to define a point in the cloud.

This study compares the Volumetric and Surface Models of shape encoding and surface evolution and points to new exciting possibilities.

1.0 Introduction:

Previous attempts at Evolving useful 3-D form by means of a computer program using a Genetic Algorithm (GA) have tended to involve a complex linking between the digital "Genetic code" and its realization. While much of the research in the field of Evolutionary Graphics was done only a decade ago; it predates the realization of the role of "Gene Expression" in the Biological Development of a complex organism. (Bentley, Kumar, 2003)
The artificiality of the GA metaphor and it's limitations to simpler graphical applications doesn't mean that its usefulness is diminished. The mapping of the Human Genome and that of other mammals has made it obvious that in real organisms, genes represent potentials only, and that other mechanisms such as proteins are responsible for growth and development as well. (Bentley, Kumar, 2003) It should become clear that much utility can be gained within the domain by concentrating on approaches that limit the Biological Evolutionary metaphor to that of Mendelian inheritance. Hence a focus on plant evolution and development rather than that of more complex organisms is more likely to be useful.

2.0 Previous Studies:

2.1 Computer Graphics and Evolutionary Design.

Over the past decade researchers such as Peter Bentley, Sanjeev Kumar, and Toshiharu Taura and many others have explored the possibilities of computational models of the growth of 3-D form using the Evolutionary Metaphor. The domain has been termed Aesthetic Evolutionary Design (AED), and Evolutionary Form Design, and only a very small number of generic systems have been attempted. They have tended to breed entities within a narrow problem space. For example certain styles of buildings, images, tables, and cars body shapes. (Lewis and Parent, 2000)

2.2 The Genetic Algorithm

Genetic Algorithms (GA's) are computer search algorithms that use the metaphors of natural selection and genetics to solve optimization tasks.
Genetic Algorithms in their modern form were developed by John Holland and his associates at the University of Michigan, beginning in the early sixties, and reaching an effective form in the seventies with the publication of his primary monograph titled "Adaptation in Natural and Artificial Systems" (Holland, 1975).
Since then a number of evolutionary type algorithms have been developed that produce solutions to problems. These include "evolutionary strategies", "evolutionary programming", "genetic programming" and "memetic algorithms". All work in a similar fashion to Genetic Algorithms. (Bentley, 2001)
Goldberg and others have reflected upon the similarity between the random mutation of the form of existing artifacts by inventors, combined with the natural selection by society of superior modifications, to the effective workings of a simple genetic algorithm (Goldberg, 1989).
Summary of the GA model:

1. Nominate a number of generations
2. Create an initial population
3. Set up a "biased" virtual roulette wheel.
4. Randomly generate a generation of new individuals using the "roulette wheel".
5. Randomly select a "mate" for each new individual using the "roulette wheel".
6. Using the resultant "pair" perform crossover to create a new "child" individual.
7. Randomly mutate.
8 Evaluate, then either finishes if the number of Generation has expired, or loop to 4 above.

(Goldberg, 1989).

The Genetic Algorithm seems to be the ideal tool for searching through the "Design Space" type problems provided an **effective selection criterion** can be developed
The latter, also known as the "Objective Function" or "Fitness Function", is the part of the Genetic Algorithm that performs the vital mechanism of Selection. The algorithms within this mechanism can be the most difficult to develop.
Normally if a selection criterion can be expressed mathematically in a non-trivial form, then it can be used in an "Objective Function". A major problem area in the domain in question is that of quantifying Aesthetic judgment so a suitable Objective Function can be constructed. To date there are a number of possible choices for types of Objective Functions:

- The "Subjective Function" aka the "Interactive Human Evaluation Function" or the "Artistic Human Intervention" where possible solutions are rated by "talented" observers.

- "Beauty Algorithm" Symmetry and adherence to "Golden ratio".
  See the work of Glunes and Piccardi

- Biological Development metaphors see the later work of Bentley and Kumar.

- Insect construction metaphor "Swarm Intelligence"

2.3 The "Subjective Function" or the "Interactive Human Evaluation Function"
Numerous researchers have used these techniques dating from the earliest days of research within the domain. To date it has proven very difficult to otherwise produce a fitness function to allow the computer to automatically judge the quality of procedurally generated designs. (Lewis and Parent, 2000).There are applications where interactive human evaluation of the assignment of fitness is very effective. Even partial use of the technique may prove worthwhile.

2.4 Successful Evolutionary Shape Generating Systems
The task has yielded a range of attempted solutions. These include a one to one mapping of the "genetic string" to the shape features. This means that each bit in the Genotype has a corresponding feature in the final shape. Other solutions impose a complex biologically inspired mapping of Genotype to the Phenotype. (Bentley, 1996)

The approaches used include:

2.41 Shape Generation using CAD Solid Modeler Primitives and Boolean Functions
.Researchers such as Graham, Wood, and Case (1999) have used a GA to manipulate the 3-D "Primitives" of the Constructive Solid Geometry (CSG) features of a CAD package to evolve objects. A major problem using CSG primitives and Boolean operators is that when shapes are combined, the result is a combination of the parent geometry, and the common spaces of the two parts can be completely destroyed. (Taura et al, 1998).

2.42 Using Parametric / Polynomial Functions and Solids of Revolution within a 3-D CAD Package.
Parametric Design within a CAD package lends itself well to the inclusion of a Genetic Algorithm to optimise various features of a design, providing an effective Objective Function can be devised.
Lewis and Parent (2000), have described an effective system for evolving human figure geometry to be used as unique "Avatar" characters within 3-D computer games. Their system begins with a default prototype body made up of thirty five "metaballs". The work shows great promise for the evolution of 3-D organic shapes particularly when the scope is relatively limited for example to the human shape.

2.43 Shape Generation by means of Component based Unitary Basic Building Blocks (The LEGO approach)
Bentley, 1996, 1999 and 2002 have all described aspects of his "GADES" project. Essentially these have involved the evolution of 3-D objects from basic geometric building blocks. Some, such as the coffee table designs shown have been straightforward in their construction since they were built up from discrete elements such as legs, supports and table tops. Other examples have required dramatic dynamic changes to the shape of the basic block as the evolution of the total shape evolved.

2.44 The Shape Feature Generation Process Model (SFGP Model)
Taura, Nagasaka and Yamagishi (1998) have proposed a Evolutionary 3-D Graphical system that they call the Shape Feature Generation Process model. This system has a number of advantages over those of previous methods.
Usually, the structure of data and the feature represented correspond directly/ However, in our model, shapes are represented by a process that consists of sets of rules which generate the shapes as they are executed, and the design feature of the shapes is indirectly hold in the sets of rules.
Therefore, when two shapes represented by this model are combined by integrating two sets of rules,

the features of the shapes are preserved, and often exaggerated, in the combined shapes.
(Taura, Nagasaka, and Yamagishi, 1998 P29)

With the SFGP model, the process starts (in this case) with a primary sphere. Within the GA, rules and conditions are applied as appropriate depending on the required shape of the object. The key to the latter is the cell division model used in the process. The rules act by deciding whether a dot (a cell analog) on the surface of the sphere will undergo "cell division" or not. The initial state for the model has a few "cells" placed in various places on the surface of the sphere. As the GA runs those cells corresponding to "highly rewarded" or desired states undergo cell division and spread over the sphere. After the number of generations is exhausted, there exists a distribution of "cells" (or dots) over the surface of the sphere. The final shape of the object is decoded by examining the density of "cells" around each cell. The average density at each dot becomes the length of a normal, drawn from the centre of the sphere through the dot or cell. The end of the normal then becomes a point on the new surface of the evolved object. The resulting "point cloud" can then become the basis for a polygon mesh to define the surface shape of the object. (See **Figure One**)

2.45 Associated Developments.
A number of computerised tools for the professions interested in manipulating 3-D computer images have been developed during the past decade, a useful example is the "Shape Grabber" or three-dimensional copier. These can produce a 3-D computer image file from a precision laser beam scan of a real object via a "point cloud" of surface coordinates. The rapid creation of a "virtual vector" model of a real object is a great boon for the Rapid Prototyping of industrial objects and for the reverse-engineering of products.

3.0 The Encoding of the Design Shape Problem into a form suitable for Evolution
A re-classification of Models of Shape Evolution.
With the hindsight of a decade's further development of the models of Shape Evolution it has become obvious that it may be more useful to classify the methods used into those that encode the total volume of the model into their Genotype, and those that encode their surface shape into their Genotype. For real world applications in Art and Industry in 2009, it seems that the latter is likely to be far more useful.

<u>3.1 A Volumetric Model of Shape evolution.</u>
Shapes" a Graphical Evolutionary Algorithm using the volumetric approach
In order to effectively search the "Design Space" or "Problem Space" within which the problem exists, it is necessary to encode the volume of an object into a form suitable for manipulation by a computer program. I.E. it is necessary to be able to digitize volume. It will also be necessary to decode a digitally encoded volume.
The "proof of concept" software defines a shape or volume by constructing it from tiny cube shaped volumes. Currently the above software can handle a "chromosome" of up to one million bits in length. This exists as an "individual" within the Genetic Algorithm. It can be realized as a set of 3-D axis with the maximum potential for one hundred small cubes down each X, Y, Z axis. The Genetic Algorithm worked best with a population of one hundred individuals, over ten thousand generations. The Objective function rewarded those sites that had a maximum number of neighbors. That is it encouraged individuals that had a maximum number of cubes and a minimum number of spaces as neighbors. This Objective Function was for test purposes only and may be replaced by another at a later stage.
The program fills the space in normal ascending Cartesian (X, Y, Z) order. It "populates" the space with cubes or "spaces", beginning at the origin (0, 0, 0), from the data held within the chromosome. A cube will be placed at a particular location if the corresponding bit is a binary "one". If it's a "zero", no cube is placed and the location is left empty.
The total effect from many small closely packed 3-D cubes appears as a very fine resolution shape similar to the way a high resolution image is made up of tiny pixels. See **Figure Two**
While the result shown in the figure is very angular in shape, this is a consequence of the method of populating the axes. Possibly a more organic shape could be evolved by rearranging the populating of the (x, y, z) axes into the sequence demonstrated in the process of biological cell division. This is shown in **Figure five** and results in growth in all six directions rather than just in the positive quadrant of 3-D space.

<u>3.2 A Surface Shape Model of Shape evolution.</u>

The first real attempt at encoding shape rather than volume was Taura's Cell Division Model. While it's scope was limited it was an impressive attempt to encode the surface of an object rather than it's volume. The surface is of much more interest in the domains where the Evolution of shape is likely to have practical applications.

<u>3.21The "Sea Urchin" Model of Shape Encoding.</u>

This model takes it's name from the similarity of it's shape at stages of it's development to the Australian Long Spined Sea Urchin, a creature common to the east coast of Australia (**Figure Three**). The approach borrows a concept from a particular type of "Shape Grabber" device. This scans an object and creates a file consisting of x, y, and z Cartesian coordinates corresponding to arbitrary points on the objects surface. The total array of these points (called the "point cloud") can be used to re-create the surface shape of the scanned object on a computer by the imposition of a Polygonal Mesh over the points.

By selecting an arbitrary origin, at Cartesian Coordinates corresponding to

(x,y,z) = (0,0,0), the magnitude and direction of a line from the origin to a point in the "cloud" can be easily calculated. This corresponds to the concept of a Direction Vector. So an alternative method of describing the shape of a surface would be by listing the magnitudes of such vectors within an array sequenced by incrementing their appropriate latitudes and longitudes on the surface of an imaginary unitary sphere. The endpoints of these direction vectors can then be used to create a Polygonal Mesh defining the surface shape. The Phenotype of the Graphical Evolutionary Algorithm would consist of individual "Real Number" Genes within a "Chromosome" structured such that it's position within determined it's position in 3-D space. While this system uses a non biological one-to-one mapping between the Phenotype and the Genotype, a change in any single gene immediately gives a corresponding change in the surface shape of the object.

Using an increment of ten degrees between individual vectors (or chromosomes) means that a potential three hundred and sixty degree shape could be defined by:

36 x 36 = 1296 chromosomes or surface points.

Using one degree increments, the shape could be defined by:

360 x 360 = 129600 chromosomes or surface points.

Clearly the method will work best using the courser increments.

The "proof of concept" software produced the result shown in **Figure Four**. The Vector magnitudes were generated by a software random number routine. The endpoints of each became part of the "point cloud" and were subsequently over-laid by a Polygonal Mesh. Clearly the method will result in a reasonable approximation to a 3-D shape given sufficient resolution. Work is continuing implementing the system into a working Genetic Algorithm using floating point numbers within the chromosomes.

4.0 Conclusion and further work

The usefulness of the Volumetric approach to evolving 3-D form is probably limited for most practical applications. The major advantage of the Surface encoding method is that its product is eminently usable by existing "Rapid Prototyping" software used in the Professions interested in manipulating-3-D form.

However further development of the Shapes software system is intended, particularly to recast the evolved shape into a more realistic "organic" form. It is planned to do this by mimicking the "Gene layering" process of cell "Pattern Formation".

The "Sea Urchin" model of Shape Evolution will also be fully developed. It is obvious that its resolution will have to be limited to fairly course limits otherwise the number of single genes will reach unacceptable sizes. A promising enhancement to the Surface encoding approach is to incorporate an Lindenmayer Plant Growth system to replace the Direction Vectors. These would be evolved branching structures. As each part reached its final place in 3-D space its endpoint would become a part of the final "point cloud". Finally a Polygonal Mesh would be laid over the latter. The effect would be similar to that of an ornamental hedge. This has the potential to evolve quite complex organic looking shapes, with a minimal number of mathematical chromosomes.

References.

Bentley, Peter J. and Kumar, Sanjeev (2003) *On Growth, Form and Computers*
Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris
ELSEVIER ACADEMIC PRESS, P8.

Bentley, Peter J. and Kumar, Sanjeev (2003) *On Growth, Form and Computers*
Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris
ELSEVIER ACADEMIC PRESS, P8-9.

Lewis, M, Parent, R (2000): *An Implicit Surface Prototype for Evolving Human Figure Geometry*.
Technical Report OSU-ACCAD-11/00-TR2, P2-3.

Holland, J, (1975) *Adaptation in Natural and Artificial Systems*,
The MIT Press; Reprint edition 1992 (originally published in 1975), P9

Bentley, Peter J. (2001) Digital Biology
New York/London/Toronto/Sydney/Singapore
SIMON & SCHUSTER ACADEMIC PRESS. P53

Goldberg, David E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning.*
Reading, Massachusetts
Addison-Wesley Publishing Company, INC. P10-12.

Goldberg, David E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning.*
P10-12.

Lewis, M, Parent, R (2000): *An Implicit Surface Prototype for Evolving Human Figure Geometry*. P2-3.

Bentley, Peter J. (1996) Generic Evolutionary Design of Solid Objects using a Genetic Algorithm (PhD thesis), University of Huddersfield.

Graham, I J, Wood, R L, Case, K (1999): Evolutionary *Form Design: The Application of Genetic Algorithmic Techniques to Computer Aided Product Design*
Department of Manufacturing Engineering, Loughborough University, LE113TU
www-staff.lboro.ac.uk/~enijg1/efd/ncmr.html

Taura, T, Nagasaka, I, and Yamagishi, A (1998); Application of Evolutionary programming to shape design.
Computer Aided Design, Vol. 30, No. 1, Elsevier Science Ltd. PP29-35.

Lewis, M, Parent, R (2000): *An Implicit Surface Prototype for Evolving Human Figure Geometry*. P2-3.

Bentley, Peter J. and Corne, David W. (2002) *Creative Evolutionary Systems*
London/San Diego
ACADEMIC PRESS PP41-42.

Taura, T, Nagasaka, I, and Yamagishi, A (1998); Application of Evolutionary programming to shape design. PP29-35.

Taura, T, Nagasaka, I, and Yamagishi, A (1998); Application of Evolutionary programming to shape design. PP29.

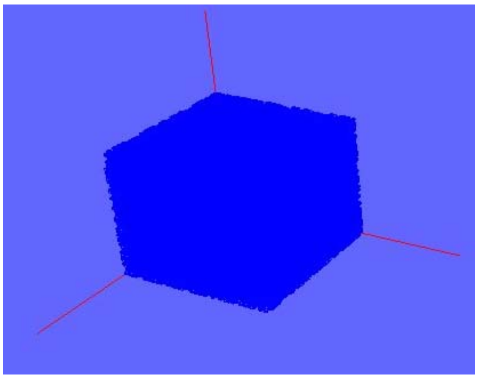

FIGURE TWO

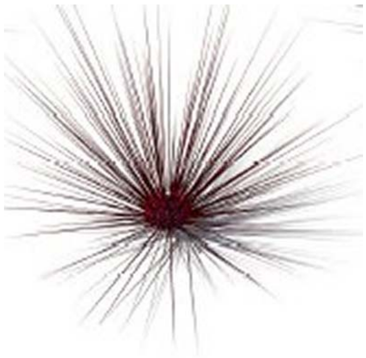

FIGURE THREE

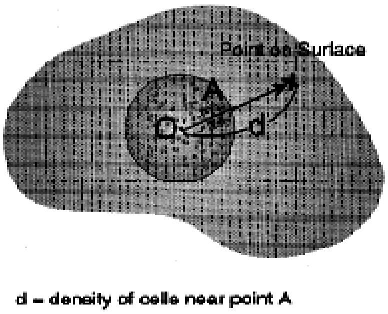

FIGURE ONE

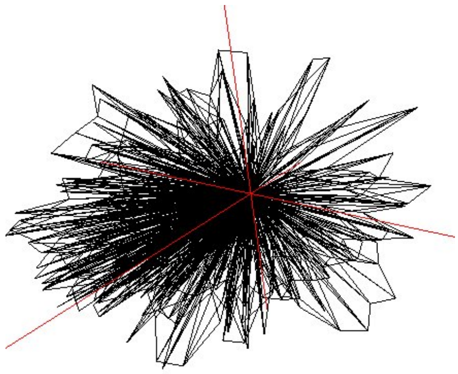

FIGURE FOUR

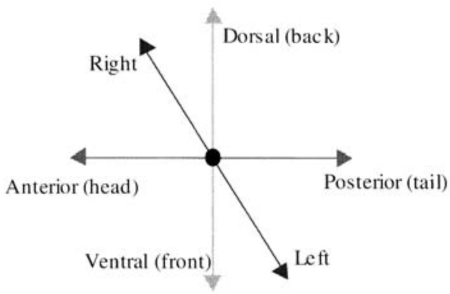

FIGURE FIVE