Copyright © 2010 ICCES ICCES, vol.15, no.4, pp.111-121

Surface reconstrucion by means of AI

T. Podoba[1], L. Tomsu[1], K. Vlcek[1], M. Heczko

Summary

Surface reconstruction based on chaotic systems or exactly given point clouds is very difficult area. Current algorithms such as Marching Cube or Voronoi Filtering do not use methods based on artificial intelligence. In this paper, we investigate solution of polygonal surface construction based on AI. The main purpose is to generate complex polygonal mesh structures based on strange attractors with fractal structure. Attractors have to be created as 4D objects using quaternion algebra or using methods of AI. Polygonal mesh can have different numbers of polygons because of iterative application of this system. Our main goal is to develop new faster algorithm to generate 3D structures and apply its optimized computational complexity for surface reconstruction and GPU benchmarking.

Keywords: surface, fractal, strange attractor, quaternion, marching cube algorithm, voronoi filtering, artificial intelligence, evolutionary algorithm.

Introduction

Computational ability of current personal computers is generally very high due to powerful GPUs according to the amount of IPS. However, surface reconstruction of complex surfaces still presents significant computational problem. There were many applications of evolutionary algorithms that served as a tool to accelerate complicated computations during last decade.

The construction of 3D surfaces from chaotic attractors or even from real surfaces described by large amount of points is a demanding task. Current deterministic algorithms like Marching Cubes or Voronoi Filtering do not use knowledge gained from artificial intelligence theory.

Our target is to integrate methods from the field of artificial intelligence into search for suitable connections between points that create surface of a 3D object. Our algorithm will be tested on static objects created from "point clouds" and consequently on strange attractors, due to their high complexity and demandingness.

As a result, different numbers of iterations and various attractor models may provide effective computational power results including special effects such us bump mapping, displacement, anti-aliasing, anisotropic filtering and ray-tracing. Moreover, the main problem is to evolve specific algorithm that generates object based on strange attractors and modifies it as a closed 3D mesh ready for real-time visualization.

[1]Tomas Bata University in Zlín, Faculty of Applied Informatics, Czech Republic

Strange Attractors

Strange attractors are complicated sets with fractal structure which chaotic dynamical systems evolve to after a long enough time. These attractors can be generated in several ways – the most commonly used are quadratic map (1) and trigonometric map (2) where parameters *a, b, c, d, e, f, g, h, i, j, k, l* define each strange attractor.

$$\begin{aligned} x_{n+1} &= a + b\cdot x_n + c\cdot x_n^2 + d\cdot x_n y_n + e\cdot y_n + f\cdot y_n^2 \\ y_{n+1} &= g + h\cdot x_n + i\cdot x_n^2 + j\cdot x_n y_n + k\cdot y_n + l\cdot y_n^2 \end{aligned} \tag{1}$$

$$\begin{aligned} x_{n+1} &= a\cdot\sin(b\cdot y_n) + c\cdot\cos(d\cdot x_n) \\ y_{n+1} &= e\cdot\sin(f\cdot y_n) + g\cdot\cos(h\cdot x_n) \end{aligned} \tag{2}$$

The strange attractor can be revealed after several iterations of a map (1) or (2). When the strange attractor is represented geometrically, it is obvious that fixed points are locally unstable, but the system is globally stable.

The attractor is chaotic when Lyapunov exponent for that map is positive. Two dimensional chaotic maps have not only a single Lyapunov exponent, but they have a positive one, corresponding to the direction of expansion, and a negative one corresponding to the direction of contraction. The signature of chaos is that at least one of these exponents is positive and the magnitude of the negative exponent has to be greater than the positive one.

For a map $x_{n+1} = f(x)$ a small deviation δx_0 of coordinate x_0 leads to a small change in x_1.

$$\delta x_1 = \delta x_0 \cdot f'(x_0) \tag{3}$$

For n iterations:

$$\delta x_n = \delta x_0 \cdot \prod_{i=0}^{n-1} f'(x_i) \tag{4}$$

Then the Lyapunov exponent is determined as

$$\Lambda = \lim_{n=>\infty}\left(\frac{1}{n}\cdot\log\left|\frac{\delta x_n}{\delta x_0}\right|\right) = \lim_{n=>\infty}\left(\frac{1}{n}\cdot\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right|\right) \tag{5}$$

Deviation $|\delta x_n|$ grows with increasing n for a chaotic orbit and this leads to a positive Lyapunov exponent$\Lambda > 0$.

Strange (chaotic) attractors are associated with motion which is unpredictable. If we attempt to predict motion of a chaotic system then even the small deviation in the initial conditions will be amplified exponentially over the time and will rapidly destroy the accuracy of our prediction. Eventually, all we will be able to say is that the motion lies somewhere on the chaotic attractor in phase-space, but exactly where it lies the attractor at given time will be unknown to us. These properties of chaotic system, extreme sensitivity to initial conditions and unpredictability, can be very helpful for the encryption purposes.

Strange attractors themselves are markedly patterned, often having elegant, fixed geometric structures, despite the fact that the trajectories moving within them appear unpredictable. The strange attractor's geometric shape is the order underlying the apparent chaos.

Quaternions

Quanternion algebra has been developed in 1843 by a mathematician William Rowan Hamilton who was searching for extension of complex numbers from 2D to 3D space. This extension is not possible to create. Only 4D structures are the closest equivalents of complex numbers. The main difference between classic complex numbers and the quaternions is that every quaternion can be described by a linear combination of four orthogonal units: 1, i, j and k. Unit 1 is called scalar unit, units i, j and k are vector units. Quaternion is defined according to (6).

$$q = x + yi + zj + wk \tag{6}$$

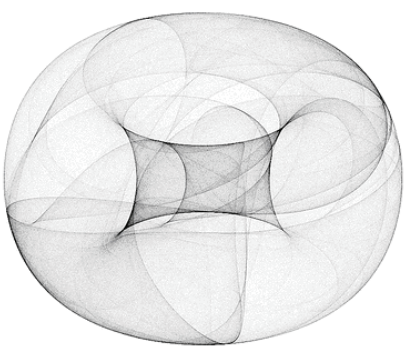

Figure 1: Example of Clifford attractor

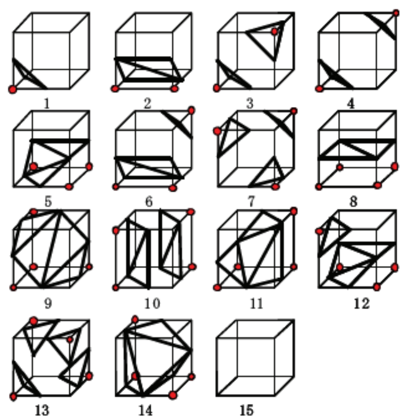

Figure 2: Basic cube configurations

Relations for mutual multiplications of the quaternion units are applied according to Tab. 1.

Tab.1 shows that units of quaternion are not following a commutative law. Thus, quaternions do not create algebraic structure. If we want to multiply two quaternions.

$$q_1 = x_1 + y_1 i + z_1 j + w_1 k \tag{7}$$

Table 1: Multiplications of quaternion units

Multiplicative operation	Result	Multiplicative operation	Result	Multiplicative operation	Result
1×1	1	j×1	j	k×j	-i
1×i	i	k×1	k	k×i	j
1×j	j	i×j	k	i×k	-j
1×k	k	j×i	-k	i×i=j×j=k×k	k
i×1	i	j×k	i	i×j×k	-1

$$q_2 = x_2 + y_2 i + z_2 j + w_2 k \tag{8}$$

we have to use the following scheme:

$$\begin{aligned} q_1 q_2 = {} & 1(x_1x_2 - y_1y_2 - z_1z_2 - w_1w_2) + i(y_1x_2 + x_1y_2 + w_1z_2 - z_1w_2) + \\ & j(z_1x_2 - w_1y_2 + x_1z_2 + y_1w_2) + k(w_1x_2 + z_1y_2 - y_1z_2 + x_1w_2) \end{aligned} \tag{9}$$

Quaternions are very often used in computer graphics for the representations of objects, their rotations and orientations because of efficient computation in complex algorithms.

Marching Cube Algorithm

Marching cube algorithm is an algorithm for creating a polygonal surface representation of a 3D scalar field. This algorithm describes voxel-object by surface of connected polygons. This method is most often used for the representation of medical data, because of simple differentiation of its parts due to thresholding.

Let's have a set of voxels. This set is browsed and processed. Values of vertices of each cube (these means intensity of 8 voxels which the cube is created from) are compared with a threshold. If the value of vertex is lower than threshold, vertex is evaluated as "internal", otherwise as "external". Evaluation is expressed as an 8-bit value which is then used for the representation of corresponding configuration of the polygon of the resultant surface.

Fig. 2 shows basic configurations of the cube, the rest of configurations are done by rotating of the basic configurations. The 8-bit value can be used as an index of 2D array with 256 rows where each row contains a proper sequence of vertices. Vertices in that table must be correctly organized in order to prevent light-collisions when drawing our object.

Shortly, the pattern of each cube is compared with 256 patterns which were previously generated by existence of measuring points and polygon is then rendered corresponding to the pattern. This procedure must be done for every cube and the 3D model is obtained this way.

Application of Artificial Intelligence and Network Theory

Complex network theory provides a large set of tools for analyzing, describing and identifying different network types and topologies. Complex networks are often a product of a complex system; thus the network theory can serve not only as a tool for understanding these networks, but consequently also as an excellent instrument for complex system analysis.

Complex system is a system composed of interconnected simple parts, that together exhibit a high degree of complexity from which emerges a higher order behavior. Complex systems cannot be described by a single rule and their characteristics are not reducible to one level of description. They exhibit properties that emerge from the interaction of their parts and which cannot be predicted from the properties of the parts.

Differential Evolution (DE) algorithm can be considered a complex system. It has many simple parts (individuals), which are interconnected through various relationships (crossover, mutation).

Network theory

Network is defined as a system of nodes (vertex) interconnected by links (edges). There many examples of networks in real life: food webs, electrical power grids, cellular and metabolic networks, the World-Wide Web, the Internet backbone, the neural network of the nematode worm *Caenorhabditis elegans*, telephone call graphs, co-authorship and citation networks of scientists, etc.

All kinds of networks can be considered a subset of a set defined by two extreme cases: n-dimensional lattice where every node connects with a well-defined set of closest neighbors and a random graph, where every node has the same probability of being connected to any other node.

Random graphs

Since all the nodes in a random graph are statistically equivalent, each of them has the same distribution, and the probability that a node chosen uniformly at random has degree k has the same form as $P(k_i = k)$. For large number of nodes, the degree distribution is well approximated by a Poisson distribution (1):

$$p(k) = \frac{e^{-\lambda} e^{\lambda k}}{k!} \tag{10}$$

Quantities used to quantitatively describe networks include:

The degree (or connectivity) ki of a node i is the number of edges incident with the node, and is defined in terms of the adjacency matrix A as (2):

$$\sum_{i,j \in N} a_{ij} \tag{11}$$

The minimum number of links that must be traversed to travel from node i to node j is called the shortest path length or distance between i and j. A graph is connected if any node can be reached from any other node; otherwise the graph is disconnected. The average path length is the average of the minimum number of steps necessary to connect any two nodes in a connected network (3).

$$L = \frac{1}{N(N-1)} \sum_{i,j \in N, i \neq j} d_{ij} \tag{12}$$

The local clustering is (roughly) the number of actual links in a local sub-network divided by the number of possible links. It quantifies the fact that if Person A is good friend with both B and C, then there is a good chance B and C are also friends.

Small-world networks

A common characteristic of networks in complex systems is the small-world property, which is defined by the co-existence of two relatively incompatible conditions, (i) the number of nodes on a path between any pair of nodes in the network is surprisingly small – usually referred to as the six-degrees of separation phenomenon— and (ii) the large local redundancy of the network— i.e., the large overlap of the circles of neighbors of two network neighbors. The latter property is typical of ordered lattices, while the former is usual for random graphs.

Recently, Watts and Strogatz proposed a minimal model for the emergence of the small-world phenomenon in simple networks. In their model, small-world networks emerge as the result of randomly rewiring a fraction p of the links in a d-dimensional lattice. The parameter $p$enables one to continuously interpolate between the two limiting cases of a regular lattice (p= 0) and a random graph (p= 1).

Scale-free networks

An important characteristic of a graph that is not taken into consideration in the small-world model of Watts and Strogatz is the degree distribution, i.e., the distribution of number of connections of the nodes in the network. The Erdös-Rényi class of random graphs has a Poisson degree distribution (1), while lattice-like networks have even more strongly peaked distributions. Similarly, the small-world networks generated by the Watts and Strogatz model also have peaked, single-scale, degree distributions, i.e., one can clearly identify a typical degree of the nodes comprising the network.

However, Barabási and coworkers found that a number of real-world networks have a scale-free degree distribution with tails that decay as a power law. Barabási and Albert suggested that scale-free network emerge in the context of growing network in which new nodes connect preferentially to the most connected nodes already in the network:

$$p_i\,(n+1) = \frac{k_i\,(n)}{\sum_{i=-n_0+1}^{n} k_i\,(n)} \tag{13}$$

where n is the time and number of nodes added to the network, n_0 is the number of initial nodes in the network at time zero, k_i is the degree of node i and $p_i(n+1)$is the probability of a new node, added at time $n+1$ linking to node i.

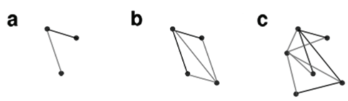

Figure 3: Three stages in the time evolution of a minimal model for generating scale-free networks

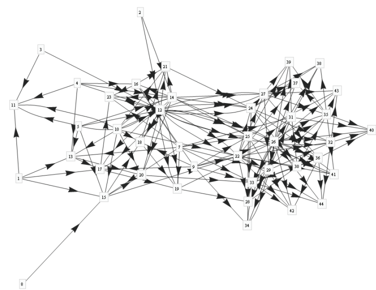

Figure 4: Network created by "DE/best/2" strategy, test function De Jong's 1, NP = 10, G = 10, F = 0.7, CR = 0.6, D = 5

Fig.3 illustrates time evolution of scale-free network:

(a) We start with a network comprising two nodes linked by a bi-directional connection. Then, we add a new node which can link to either of the existing nodes. Because both existing nodes have degree one, there is an equal probability of linked to each of them.

(b) At the following time step, we add a new node to the network. However, now the probability of linking to each of the existing nodes is no longer identical because one of the nodes has higher degree than the others.

(c) As times goes by, a heterogeneous degree distribution emerges because nodes with higher degree have a higher probability of being linked to new nodes.

Classes of small-world networks

Question: How to connect small-world networks with the new finding of scale-free structures. Specifically, one may ask Under what conditions will growing network be scale-free?

Preferential attachment is essential for creation of scale free networks. However, not every network, which is created by preferential attachment, is scale-free. What are the reasons for such results? There are several factors, which prohibit creation of scale-free networks in real world:

Aging – Represents situation, when even the most connected node stops receiving new nodes because of its age. In such case, the network will not have complete scale-free topology.

Cost of adding a link and limited capacity – There are usually limits concerning number of links to a single node – either the capacity is limited, or adding new link is too "expensive".

Limits on information and access – There may be constraints, which prohibit connections between some nodes, based on access or information.

These three factors may cause, that network won't have scale-free topology, even though preferential attachment was present during its creation.

Differential Evolution

Differential evolution (DE) is an evolutionary algorithm. It works over D-dimensional search space of considered problem. A population has NP number of individuals. These individuals are D-dimensional vectors of parameters representing possible solutions to the given problem. Initially, these parameters are randomly and uniformly set between pre-set boundaries (Hi,Lo). DE then evolves a population of NP D-dimensional individual vectors from one generation to the next. These vectors are candidate solutions. At each generation G, DE applies mutation and crossover to all individuals to produce trial vectors.

There are various DE strategies differentiated by mutation operation.

$$X_{i,G} = \{x_{1i,G}, x_{2i,G}, \ldots .x_{Di,G}\} \quad V_{i,G} = \{v_{1i,G}, v_{2i,G}, \ldots ..v_{Di,G}\}$$

X, V = target, mutant vectors at generation G.

The mutant vector is created at the generation G using one of many possible DE strategies:
"DE/rand/1":

$$V_{iG} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$$

"DE/best/1":

$$V_{iG} = X_{best,G} + F(X_{r2,G} - X_{r3,G})$$

"DE/current to best/1":

$$V_{iG} = X_{i,G} + F(X_{best,G} - X_{r1,G}) + F(X_{r1,G} - X_{r2,G})$$

"DE/best/2":

$$V_{iG} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G})$$

"DE/rand/2":

$$V_{iG} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G})$$

"DE/rand to best/2":

$$V_{iG} = X_{r1,G} + F(X_{best,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G})$$

r_j random mutually different integer values generated in the range [1,NP], which should also be different from the current trial vector's index i.

Complex networks created by DE

Our first target was to study complex networks generated by DE algorithm. These networks are created according to population history. Each new accepted individual is connected to its parents, i.e. to individuals that were part of crossover operation, which led to creation of this new successful individual (Fig 4).

The networks are based on crossover operation. In this case, "DE/best/2" strategy was studied:

$$V_{iG} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \tag{14}$$

If the trial vector is accepted as a new individual, it is added to the network as a new node, which is connected to individuals that were participating in its creation. In this case, X_{new} (the new accepted individual) is connected to$X_{best,G}$, $X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$, $X_{r4,G}$. The preferential attachment is obvious as $X_{best,G}$ takes part in all crossovers. This fact implies that the currently best individual node is connected to all new nodes added to the network. Consequently, each new node has five inward links – it has five "parents."

This study presented the complex networks generated by DE. Our next target is to match DE (population, individuals) onto surface reconstruction problem.

Surface Reconstruction by Means of AI

This preliminary research serves as a ground for further experiments involving evolutionary algorithms. Our main target is to develop algorithm based on fundamentals gained from the evolutionary algorithms theory that will effectively solve surface reconstruction problem. Strange attractors will provide highly complicated surfaces, which will be used as a testing environment. Currently we are considering modifying DE algorithm. The population should consist of points that create the reconstructed surface. However, special ways to reduce the computational complexity of such approach must be considered due to large number of these points. This approach will probably lead into developing an entirely new evolutionary algorithm, which will be fully optimized for these particular applications.

Conclusion

In this paper, possible methods how to solve the difficult problem of generating a closed 3D mesh based on strange attractors using quaternions, marching cube algorithm and especially artificial intelligence methods such as differential evolution were briefly described. An increasing amount of parameters such as number of iterations, generating attractor values or the usage of special effects will cause exponential growth of the complexity of created 3D object. Evolutionary algorithms were successfully used for such complex tasks during last decade. Based on these premises, we are searching for an algorithm based on artificial intelligence methods that can effectively solve this problem.

Note that our algorithm is in development currently and final results are going to be published in future.

Future Work

DE algorithm will be modified (if possible) for use on a surface reconstruction problem. This algorithm will be tested on a set of problems consisting of "point clouds" and consequently of strange attractors.

Acknowledgement

The authors would like to acknowledge Tomas Botta University in Zlin, Faculty of Applied Informatics in Czech Republic. This paper was published under IGA project number: IGA/35/FAI/10/D

References

1. **Alan Norton**, *Generation and Display of Geometric Fractals in 3-D*, Computers & Graphics, Vol. 16(nr 3), pp. 61–67, 1982.

2. **Albert, R. and Barabasi, A.-L.**, *Statistical mechanics of complex networks*, Rev. Mod. Phys. 74, 47–97 (2002).

3. **Albert, R., Jeong, H., and Barabasi, A.-L.**, *Diameter of the world-wide web*, Nature 401, 130–131 (1999).

4. **Barabasi, A.-L. and Albert, R.**, *Emergence of scaling in random networks*, Science 286, 509–512 (1999).

5. **Bourke, P.**, *Polygonising A Scalar Field*, May 1994, Available at WWW: http://local.wasp.uwa.edu.au/∼pbourke/geometry/polygonise/
6. **de Oliveira P.M.C.**, *Why do Evolutionary Systems Stick to the Edge of Chaos* (Theory in Biosciences, Volume 120, Number 1, 1 March 2001, pp.1-19(19))
7. **Dorogovtsev, S. N. and Mendes, J. F. F.**, *Evolution of networks with aging of sites*, Phys. Rev. E 62, 1842–1845 (2000).
8. **Giesl, J., Vlcek, K.** (2009): *Image Encryption Based on Strange Attractor*, ICGST-GVIP Journal. 2009, vol. 9, is. 2, pp. 19-26. ISSN 1687-398.
9. **L.A.N. Amarala and J.M. Ottino**, *Complex networks* (Cambridge THE EUROPEAN PHYSICAL JOURNAL B), Eur. Phys. J. B 38, 147–162 (2004)
10. **Newman M.E.J.**, *Power laws, Pareto distributions and Zipf's law* (Contemporary Physics, Volume 46, Number 5, September 2006, pp.323-351(29))
11. **N. Hansen**, *"Compilation of results on the 2005 CEC benchmark function set"*, http://www.ntu.edu.sg/home/epnsugan/index files/CEC-05/compareresults.pdf May 4, 2006.
12. **Sprott, J.C.** (1993): *Strange attractors: Creating Patterns in Chaos*. M&T Books.
13. **Storn, Rainer and Price, Kenneth** (1995). *Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical report TR-95-012, ICSI, March 1995.
14. **Storn, Rainer and Price, Kenneth** (1997). *Differential evolution - A simple evolution strategy for fast optimization*. Dr.Dobb's Journal, April 97, pp.18-24 and p.78.
15. **Yokoyama, H., Chikatsu, H.**, *3D Representation of Historical Structure for Digital Archives by Laser Scanner*, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV, Part 5/W12, pp. 347-350.
16. **Yagel R., Cohen, D., Kaufman, A.**, *Discrete ray tracing, Computer Graphics and Applications*, IEEE, Volume: 12 Issue: 5, pp. 19–28, 1992.
17. **Zelinka Ivan**, *Umlá inteligence, kap.6. "Diferenciální evoluce"*, Academia, 2004.
18. **Zelinka, I., Vasek, V., Lampinen, J.**: *SOMA a diferenciální evoluce - nové algoritmy globální optimalizace*, Fine Mechanics and Optics, 4/2002, p.112-117, ISSN 0447-6441.