Triangulation of Generative Form for Parametric Design and Rapid Prototyping

Thomas Fischer1,2, Mark Burry1, John Frazer2

1Spatial Information Architecture Laboratory, RMIT, Melbourne, Australia

2School of Design, The Hong Kong Polytechnic University, Hong Kong, China

http://www.sial.rmit.edu.au, http://www.sd.polyu.edu.hk

Abstract. In this paper we discuss recent developments in the ongoing implementation of a toolkit for developmental generative design and form finding. We examine tissues of face-centered cubically close-packed voxel cells and topologically related structures for the possibility of 3D data conversion and of rapid prototyping applications. We also demonstrate how generative and parametric design can be integrated in order to enhance design flexibility and control.

Keywords. parametric design, digital morphogenesis, cellular expression, geometry triangulation.

Introduction

Being the result of a long-term software development project, the voxel automata system Zellkalkül allows simulations of non-uniform, three-dimensional isospatial cellular automata structures with behavior based on decentralized, massively parallel, individually high-level programmable cellular units. Its design was inspired by principles identified in developmental evolutionary biology. The system is used to explore tempo-spatial mechanisms of morphogenesis in developmental architectural and design contexts. It has also been applied in theoretical generative design research and in generative design teaching. For previous discussions of the system see: Fischer, Fischer and Ceccato (2002); Fischer (2002); Fischer and Fischer (2003) and Fischer, Burry and Frazer (2003).

Figure 1 shows three modes of tissue representation currently supported by the system, all based on face-centered cubic close packing arrangement: Rhombic dodecahedra, spheres and "octet trusses" or "space frames". The latter

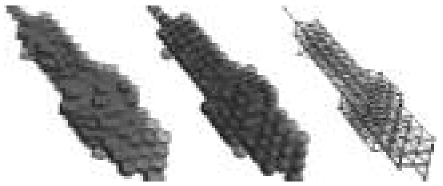

Figure 1. Tissue representation based on rhombic dodecahedra, spheres and "isotropic vector matrix".

ones were extensively discussed and applied by Buckminster Fuller (1975) who used the term "isotropic vector matrix" for this type of structure. Close-packed spheres have previously been examined as models for coding data and programs as well as for representing composite form, for example by Lefever, Rastogi as well as by Kong (1994) and Frazer (1995). Rhombo-dodecahedral close-packing was identified as a common form of natural cell composition for example by Kieser (1815) and by Thompson (1992) and has been applied to physical form composition in the Digital Clay project at Xerox Parc (www2.parc.com/spl/projects/modrobots/lattice/digitalclay: May 2003). The idea of approaching architectural form finding and construction by means of developmental cellular growth models has been discussed by Zuse (1993), Otto (1971), Frazer and Connor (1979), Smith (1992) and Frazer (1995).

Despite its isospatial symmetry, as Frazer (1992) has pointed out, it has excellent coding capabilities. From the viewpoint of finding and modelling architectural form, the basic spatial/cellular structure of face-centered cubic close packing shows particular shortcomings with respect to its formal expressive capabilities. The cellular voxel units generate structures that are geometrically largely restrained to the automata system's grid structure and the "jagged" cellular geometries. The voxel shape especially dominates those cellular designs that are based on lower voxel resolutions and hence is comprised of smaller numbers of relatively large cells. The work described in this paper attempts to describe the development of system extensions to overcome this limitation.

Voxel-Based Free-Form Modelling Strategies

Today, artificial automatic form-finding scenarios using developmental cellular approaches with adaptable cell geometries are very difficult to achieve physically. In this area there are great challenges ahead both in terms of the required self-organizational capabilities as well as in terms of the mechanical and structural engineering and robotics involved - see also Fischer, Burry and Frazer (2003). Here, we examine how self-organization could interrelate with three-dimensional geometric operations in order to generate cellular structures virtually, without confronting the physics of dynamically adjusting geometry compounds at this point. Our objective is to utilize virtual space as a generative simulation environment for dynamically developing structures but at the same time to allow for physical output by means of CAM facilities once a form is considered interesting or useful during the design process.

A variety of strategic approaches are possible to support the generation of "free form" using associative geometry based on developmental isospatial cell composition. We have identified and partially previously applied the following methods:

1. Using large numbers of automata at a high resolution to approximate smooth 3D shapes. One disadvantage of this approach is that smooth surfaces are not achieved, only approximated at the cost of exponentially increasing memory consumption during shape generation. Formal expression is moreover achieved primarily by means of an additive composition and not by parametric control of geometric relations. The usefulness of both parametric design principles in combination has been discussed by Kolarevic (2000).

2. "Skinning" of cell assemblies using surface fitting algorithms.

3. "Skinning" using cell center points or cell attributes to control separate skin geometry, e.g. mapping cell location or other cell-related data onto free curve control-points.

4. Mapping of data generated in Zellkalkül's automata structure onto secondary output geometries. This and the above two strategies however contradict the software's intention to provide a single unified geometric and data structure.

5. Parametric position control of cell polygon vertices. This approach will be discussed in the remaining part of this paper.

6. Parametric cell sizes control. This approach involved distorting the lattice of cellular center points and appears to a highly interesting future extension. We have however not yet identified suitable operations and constraints for the required geometric operations.

7. Changing the distances between cell center

points, i.e. substituting the isotropic vector matrix by an “anisotropic vector matrix”. This approach is a combination of the above two approaches in which tissues are represented in form of vector trusses instead of as “solid” cells.

Krawczyk (2002) points out that in digital architecture form is often derived from initial sets of parameters that can be changed to achieve alternative results. He then proceeds to demonstrate how adaptive geometry can be used to interpret the form of cellular-automata based form in different ways. As we have previously stated, parametric design aims to limit the problems that come along with totally free design situations – but it does so potentially at the cost of the freedom of those situations; see Fischer, Burry and Woodbury (2000), p. 146. The extent to which parametric design supports and restrains design situations largely depends on the design system’s atomic resolution – the granularity level dynamic parameters are applied at and the systems ability to “epigenetically” generate useful design solutions. In this sense, it might be useful in many cases to break down single, primary parametric patterns into smaller units based on associative geometry as demonstrated by Krawczyk and to incorporate essential expressive information into the developmental design code that governs individual cellular behavior during form generation. The overall design pattern will hence not be described in the form of a single geometric topology but rather in the form of a problem-specific developmental program, while a generic parametric design topology exists on the cellular level.

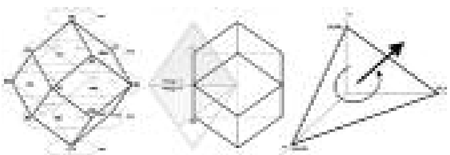

Figure 2.Rhombo-dodecaherdal voxel geometry, triangulation and vertex ranges, STL format illustration.

The result of this strategy is a parametrically enhanced generative form-finding process that shares essential features with growth processes in Nature. We speculate that interrelating the (predominantly top-down-structured) parametric design paradigm with the (predominantly bottom-up-structured) generative design paradigm yields very promising potentials for new generative design approaches.

Parametric Logic, User Interface and Constraint Space Definition

As a first step, the rhombic cell faces were triangulated in order to allow vertices to move individually without affecting other vertex positions of the same face while continuing to allow tissues to remain close-packed without void or overlapping spaces. This triangulation is also useful in facilitating the creation of triangle-based output file formats such as STL, which is the key in producing stereo-lithographic rapid prototype models of generated form. For these two purposes, we have split all the rhombic faces of the dodecahedral units into two triangles as illustrated by the dashed lines in the middle of figure 2. As a next step, it was necessary to identify the movement ranges of all vertices in order to achieve useful geometric constraints for parametric manipulation. By “movement ranges” we refer to the spaces or domains within which each vertex is allowed to move while avoiding vertex eversions (which would describe unwanted inverse spaces). This eversion problem would for instance occur if vertices {1} and {9} in the middle of figure 2 were to change their positions in such a way that {9} will be located above {1}; the cellular space between both vertices would evert and result in an undefined space.

To allow vertices to move freely without mutual eversion, the movement ranges must fill the entire tissue space without overlapping and hence allow close packing themselves. The left illustration shown in figure 2 shows a rhombic dodecahedron with faces numbered using Miller indices (mineralogical face identification system), vertex numbers as used in Zellkalkül and two different vertex types labeled 'a' and 'b'. Vertices between six adjacent cells (labeled 'a') have octahedral ranges while vertices between four adjacent cells (labeled 'b') have tetrahedral ranges as shown in the two left illustrations in figure 3. The side length (or in Fuller's terminology: the geodesic vector length) of both geometries is 1 (identical to cell diameter). The second right of figure 3 shows both types of these platonic shapes (second from left, octahedra are only shown half) and (outer right) their ability to close-pack into cuboctahedral assemblies. The twelve cells neighboring the central cell are shown as wire frames in both illustrations. By constraining vertices to always remain within their ranges, this topology prevents self-intersecting cell surfaces. Figure 4 shows sections through the assembly of overlapping coboctahedral vertex movement ranges of twelve close-packed rhombo-dodecahedral cells in two orientations.

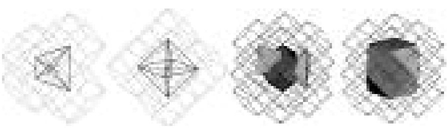

Figure 3.Tetrahedral and octahedral vector matrices and resulting cuboctahedral cluster of vertex movement ranges.

The last step of our strategy for approximating free forms required a suitable user interface and control mechanism, which allows users to move vertices in intuitive ways using scripting functions. In order to keep the user interface for vertex position control simple, intuitive and yet as flexible as possible we have designed it to not distinguish between the two different vertex types 'a' and 'b' despite the fact that they are internally processed in different ways. We have decided to

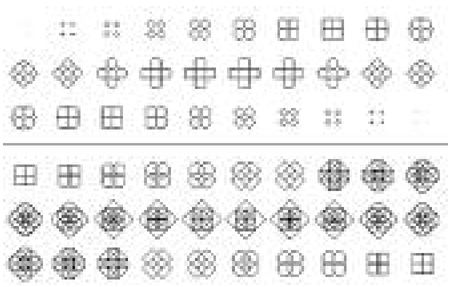

Figure 4.Continous sections across vertex movement ranges of 12 close-packed rhombic dodecahedra.

use a pressure model that allows users to control the pressure cells exert in the direction of a given vertex. The two left illustrations on figure 3 show the force vectors with which adjacent cells can manipulate both types of vertices. An advantage of using a pressure model for vertex position control is that positions, as in Nature, result relatively from intercellular "negotiation" rather than by unilaterally controlled, absolute positioning. A problem emerges when trying to move a vertex between four cells (shown on the very left of figure 3) by means of four pressure vectors in caltrop arrangement (see inside tetrahedral range) since this does not allow the vertex to reach any point within this range. We have solved this problem by modelling forces as negative pressure (or: "tension") rather than as "pressure".

The script interpreter associated with every cell represents the user interface for geometry control. For this purpose we have extended the scripting language specification with two problem-centered functions. These functions allow the identification of a vertex and a (negatively interpreted) pressure in the format setTension(v, p) to set a pressure and in the format getTension(v) to acquire the pressure of a given vertex. The pressure parameter is expressed as a byte value, which allows relative pressure control at a resolution of 255 steps per cell involved in a vertex positioning operation. The neutral default pressure of each vertex, as well as the "atmospheric" pressure (relevant at tissue edges where vertices have no neighbors), is 127. With this default value assigned to all vertices internally and to vertices of adjacent cells, a cell assumes a normal rhombic dodecahedral shape. Other values result in "morphed" variations as shown in detail in Figure 4 (screen rendering and physical model).

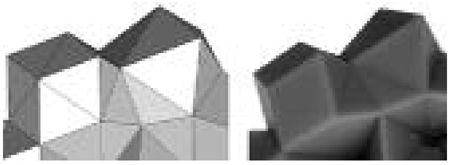

Figure 5.Detail of parametrically altered close-packed voxels.

STL Data Output

As discussed above, the rhombic faces of the dodecahedral units had to be broken down into two triangles each to allow for individual vertex movement. With the coordinates of these triangles available, it was easily possible to develop an STL file output interface. While there are solutions publicly available that allow STL data to be loaded into Java3D scene graphs (see http://www.j3d.org/utilities/loaders.html), we believe the work discussed here to be one of the very first STL output solutions for Java3D-based modelling. Of the two STL specifications existing, one is based on ASCII data while the other one is based on much more space-efficient binary data. We have found both formats to have specific advantages and disadvantages in our context. The ASCII variant has proved to be more human-readable and hence "developer friendly" during software development and debugging while the binary format produces significantly smaller files and allows for more efficient and faster software analysis. We have therefore implemented output facilities for both formats. More details on the ASCII and binary STL specifications can be found in Burns (1993), pp. 227 ff. The STL specifications require triangular face orientation (normally oriented) to be expressed redundantly by counter-clockwise vertex order as well by an outwardly-directed perpendicular normal vector as illustrated on the right of figure 2. Our STL filter hence only needs to find the vertex order appropriate for each triangle's normal orientation and a fourth point in space denoting the normal vector. Internally, Java3D represents complex models as triangle-strips, which can be analyzed and allow any shape to be converted to STL. We have already made significant progress into this direction. The more complex data generated by our system however describes close-packed spheres and space frame trusses which contain many half-enclosed spaces that easily involve support material complications in rapid prototyping. Since the availability of multicolor rapid prototyping machines, the STL specification has been enhanced to accommodate color data. Although not having this type of equipment immediately available, we have not yet focused on this option. With respect to inter-operability with other 3D modelling and rendering packages, STL output of space frame structures and spheres has already proven to be very useful and color-enhanced data output appears to be a highly interesting possibility for future work.

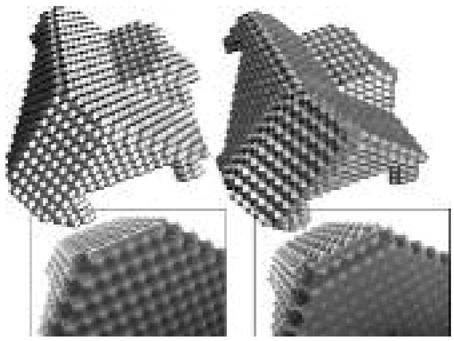

Figure 6.Enclosure structure based on a design by Frazer (1966-68) based on dodecahedral units in original (left) and parametrically altered geometries (right).

Figure 6 shows renderings and close-up photographs of rapid prototype models based on a 1968 enclosure design for the Coventry Preparatory School by Frazer (1974), (1979). Both models are generated in parallel developmental processes in which an initial 3D data array is expressed by virtual cellular proliferation. Once the final shape has developed, close-range intercellular communication is used to generate a local sense of cellular identity. This sense of identity is then used to trigger geometric alterations in parts of the design. The model on the left is based on unaltered rhombo-dodecahedral voxel units and shows a jagged model surface. The model on the right shows one surface approximating a smooth surface and one 'spiky' surface achieved by decentralized parametric cell geometry control.

Conclusion

We have given insights into our tool-making work in the area of computer-aided developmental evolutionary design. The system we have described allows for parametrically enhanced generative form finding and modelling processes that can either remain within or transcend paradigms of natural growth and development. It utilizes virtual space as an experimental form finding environment but supports STL file export for interoperability with common 3D modelling packages as well as CAM output facilities such as STL rapid prototyping machines. As the early experiments we have shown suggest, one strategy of overcoming previously discussed shortcomings of parametric design could be the introduction of fine-grain associative geometry to voxel-based generative design in analogy to growth and developmental expression of cellular form in Nature.

Acknowledgements

The authors gratefully acknowledge the support from their colleagues at the Spatial Information Architecture Laboratory at the School of Design at The Hong Kong Polytechnic University and at the Royal Melbourne Institute of Technology, in particular Andrew Maher. We also thank Torben Fischer from the Faculty of Mathematics at the University of Göttingen for his excellent contribution in algorithm and filter development as well in implementing the discussed software. Many thanks also to the workshop team of the School of Design and The Industrial Centre of The Hong Kong polytechnic University for their excellent rapid prototyping support.

References

Burns, M.: 1993, Automated Fabrication. Improving Productivity in Manufacturing. Prentice Hall, Englewood Cliffs.

Burry, M and Zolna, M.: 1997, Architectural Design Based on Parametric Variation and Associative Geometry, in Martens, B. et al. (eds.), Challenges of the Future. Proceedings of the 15th eCAADe Conference, Österreich Kunst und Kulturverlag, Vienna, Austria, pp 1–11.

Chiu M.-L., Lin, Y.-M., Lee, C.-H. and Tsai, P.-H.: 2001, Teaching Rapid Prototyping in CAD Studios for Creative Design, in Gero, J. et al. (eds.), Proceedings of CAARDIA2001. Key Centre for Design Computing and Cognition, University of Sydney, Australia, pp. 307-310.

Fischer, T. and Burry, M., Woodbury, R.: 2000, Object-Oriented Modelling using XML in Computer-Aided Architectural and Educational CAD, in Tan, Beng-Kiang et al (eds.), CAADRIA2000. Proceedings of the 5th Conference on Computer Aided Architectural Design Research in Asia, Singapore, pp. 145-155.

Fischer, T., Fischer, T. and Ceccato, C.: 2002, Distributed Agents for Morphologic and Behavioral Expression in Cellular Design Systems, in Proctor, G. (ed.), Thresholds. Proceedings of the 2002 Annual Conference of the Association for Computer Aided Architectural Design. Pomona, Los Angeles, California, pp. 111-121.

Fischer, T.: 2002, Computation-Universal Voxel Automata as Material for Generative Design Education. in Soddu, Celestino et al. (eds.), Proceedings of the 5th Conference and Exhibition on Generative Art 2002. Generative Design Lab, DiAP, Politechnico di Milano University, Italy, pp. 10.1-10.11.

Fischer T. and Fischer T.: 2003, Parametric Voxel Geometry Control for Digital Morphogenesis. Proceedings of the 19th European Workshop on Computational Geometry, Institute for Computer Science I, University of Bonn, Germany, pp. 35-40.

Fischer, T., Burry, M. and Frazer, J.: 2003, How to Plant a Subway System, in Chiu, Mao-Lin et al. (eds.), Digital Design, Research and Practice. The Proceedings of The 10th International Conference of Computer Aided Architectural Design Futures. National Cheng Kung University, Tainan, Taiwan (publication pending).

Frazer, J.: 1974, Reptiles. In: Architectural Design 4/1974, pp. 231-240.

Frazer, J. and Connor J.: 1979, A Conceptual Seeding Technique for Architectural Design, in PArC 79 Proceedings, AMK, Berlin. Uxbridge, UK: Online conferences Ltd.

Frazer, J.: 1995, An Evolutionary Architecture. Architectural Association, London.

Frazer J.: 1992, Datastructures for Rule-Based and Genetic Design, in T.L. Kunii (ed.) Visual Computing - Integrating Computer Graphics with Computer Vision, Springer-Verlag, Tokyo, pp 731-44.

Fuller, R. B.: 1975, Synergetics. Explorations in the Geometry of Thinking. Macmillan, New York.

Kieser, D. G.: 1815, Phytotomie, oder Grundzüge der Anatomie der Pflanzen. Jena, Cröcker.

Kolarevic, B.: 2000, Digital Morphogenesis and Computational Architectures, in Kos, J. R. (ed.), The Proceedings of SIGraDi2000 - Construindo (n)o espacio digital (constructing the digital Space), Facultad de Arquitectura, Universidad Nacional de Mar del Plata, Rio de Janeiro, Brazil, pp. 98–103.

Kong, G.: 1994: Cellular Automatas and "Stacking Balls". Technical dissertation forming part of the Diploma of the Architectural Association, London.

Krawczyk, R. J.: 2002: Experiments in Form Generation Using Cellular Automata, in Koszewski, K and Wrona, S. (eds.), Design e-Ducation. Connecting The Real and the Virtual. Proceedings of the 20th eCAADe Conference. Warsaw University of Technology, Warsaw, Poland, pp. 552-555.

Otto, F.: 1971, IL3 Biologie und Bauen. Mitteilungen des Instituts für Leichte Flächentragwerke, Stuttgart.

Smith A. R.: 1992, Simple Non-Trivial Self-Reproducing Machines, in Langton C. G. et al. (eds.), Artificial Life II, vol. X of SFI Studies in the Sciences of Complexity, Redwood City, CA: Addison-Wesley, pp. 709-725.

Thompson, D. W.: 1992, On Growth and Form. The Complete Revised Edition. Dover Publ., New York.

Zuse, K.: 1993, Der Computer. Mein Lebenswerk. Springer Verlag, Berlin.