Geometric Modeling without Coordinates and Indices (Extended Abstract)

Przemyslaw Prusinkiewicz
Department of Computer Science, University of Calgary
pwp@cpsc.ucalgary.ca

Since Descartes, coordinates have been the standard mathematical construct for describing and manipulating geometric objects. In addition, indices are commonly used to label elements of discrete geometric structures, such as polygon meshes. Since the coordinates and indices are numbers, they can be manipulated algebraically. This sets the framework for the handling of geometric objects in algebraic terms that is the essence of analytic geometry. In some modeling applications, however, the use of coordinates leads to complications that are not inherent in the modeling problems themselves. For example, DeRose pointed to both confusion and errors stemming from different geometric interpretations of the same transformation matrix [3, 4].

We will focus on dynamic geometric objects, which can be seen to develop over time. Examples include fractals, subdivision curves and surfaces, and models of growing biological structures. Coordinates and indices do not identify elements of these objects in a convenient, time-invariant way. For instance, the coordinates of a cell in a growing biological tissue may change over time, even though it materially is the same cell. Furthermore, if the cells are indexed, the indices may have to be updated following cell division in order to preserve consecutive numbering of the adjacent cells. Similarly, in the case of subdivision curves and surfaces, the existing points may change position over time, while the insertion of new points may require all points to be reindexed.

We can address these difficulties using formalisms that hide coordinates and indices from the modeler's perspective. Such formalism are expressed in local terms, which means that:

- arguments of geometric operations are selected based on the state of the elements of the structure and their neighbors,
- these neighbors are identified using suitably defined topological relations, and
- all operations are expressed in a coordinate-free manner.

Cellular automata [11] provide an example of a formalism that meets these criteria. In each step of a cellular automaton operation, every cell is assigned a new state according to its previous state and the states of its neighbors. These neighbors can be referred to as the left, right, up, and down neighbors, which does not require the use of coordinates or indices.

L-systems [8] with the turtle interpretation [9] provide another example of algorithm specification in local terms. Unlike cellular automata, which may represent surfaces, L-systems are limited to linear and branching structures. On the other hand, the L-system rules allow for the insertion and deletion of cells, and thus support objects with a changing topology (dynamical systems with a dynamic structure [7]). This is the foundation of the L-system applications to the generation of fractals and simulation of plant development [10].

I will show that other geometric constructs, based on the affine geometry operations [4], Euclidean geometry constructions [6], and physically-based modeling, rather than turtle geometry, can also be used to interpret L-systems. These alternative interpretations have useful practical applications. For example, L-systems with the affine geometry interpretation provide a compact and intuitive representation of subdivision algorithms for curves.

Specifically, let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ be a sequence of control points, which defines a closed subdivision curve. Denote by $P(\mathbf{v})$ the L-system module [10] representing point $\mathbf{v}$, and let the string of modules

$$P(\mathbf{v}_1)P(\mathbf{v}_2)\ldots P(\mathbf{v}_n)P(\mathbf{v}_1)$$

denote the axiom of a family of L-systems operating on circular words. Assume that the standard notation for L-system productions [10] has been extended with the affine geometry operations [4] on module parameters. The following sample productions implement Chaikin [2], B-spline [1] and Dyn-Gregory-Levin [5] subdivision schemes:

$$P(\mathbf{v}_l) < P(\mathbf{v}) > P(\mathbf{v}_r) \rightarrow \\ P(\tfrac{1}{4}\cdot\mathbf{v}_l + \tfrac{3}{4}\cdot\mathbf{v})P(\tfrac{3}{4}\cdot\mathbf{v} + \tfrac{1}{4}\cdot\mathbf{v}_r)$$

$$P(\mathbf{v}_l) < P(\mathbf{v}) > P(\mathbf{v}_r) \rightarrow \\ P(\tfrac{1}{2}\cdot\mathbf{v}_l + \tfrac{1}{2}\cdot\mathbf{v})P(\tfrac{1}{8}\cdot\mathbf{v}_l + \tfrac{3}{4}\cdot\mathbf{v} + \tfrac{1}{8}\cdot\mathbf{v}_r)$$

$$P(\mathbf{v}_{ll})P(\mathbf{v}_l) < P(\mathbf{v}) > P(\mathbf{v}_r) \rightarrow \\ P(-\tfrac{1}{16}\cdot\mathbf{v}_{ll} + \tfrac{9}{16}\cdot\mathbf{v}_l + \tfrac{9}{16}\cdot\mathbf{v} - \tfrac{1}{16}\cdot\mathbf{v}_r)P(\mathbf{v})$$

Proceedings of the Shape Modeling International 2002 (SMI'02)
0-7695-1546-0/02 $17.00 © 2002 **IEEE**

The essence of L-systems can be extended to polygon meshes. This requires a representation of mesh topologies with suitably defined operations for accessing neighboring elements. Several such representations exist and, similar to the linear and branching structures generated by L-systems, they can be associated with different geometric interpretations. The affine interpretation is well suited for the generation of subdivision surfaces, whereas the interpretations based on Euclidean geometry and physically-based modeling are useful in the simulation of growing biological structures. The resulting geometric rewriting systems preserve the index- and coordinate-free character of L-systems, thus making it possible to express further classes of modeling algorithms in a simple manner.

Acknowledgment

I would like to thank Jean-Louis Giavitto, Pascal Lienhardt and Faramarz Samavati for interesting discussion that contributed to the development of the above ideas.

References

[1] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350–355, 1978.

[2] G. Chaikin. An algorithm of high speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.

[3] T. DeRose. Three-dimensional computer graphics. A coordinate-free approach. Manuscript, University of Washington, 1992.

[4] T. DeRose. A coordinate-free approach to geomeric programming. In W. Strasser and H.-P. Seidel, editors, *Theory and practice of geometric modeling*, pages 291–305. Springer-Verlag, Berlin, 1989.

[5] N. Dyn, J. Gregory, and D. Levin. A four-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4:257–268, 1987.

[6] N. Fuller and P. Prusinkiewicz. Geometric modeling with Euclidean constructions. In N. Magnenat Thalmann and D. Thalmann, editors, *New trends in computer graphics. Proceedings of CG International '88*, pages 379–391. Springer-Verlag, Berlin, 1988.

[7] J.-L. Giavitto and O. Michel. MGS: A programming language for the transformation of topological collections. Research Report 61-2001, CNRS - Universitè d'Evry Val d'Esonne, Evry, France, 2001.

[8] A. Lindenmayer. Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.

[9] P. Prusinkiewicz. Graphical applications of L-systems. In *Proceedings of Graphics Interface '86 — Vision Interface '86*, pages 247–253, 1986.

[10] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

[11] T. Toffoli and N. Margolus. *Cellular automata machines: A new environment for modeling*. MIT Press, Cambridge, 1987.