Tensor Product Surfaces as Rewriting Process

Ivana Kolingerová *
University of West Bohemia
Czech Republic

Petr März †
University of West Bohemia
Czech Republic

Bedřich Beneš‡
Purdue University
U.S.A.

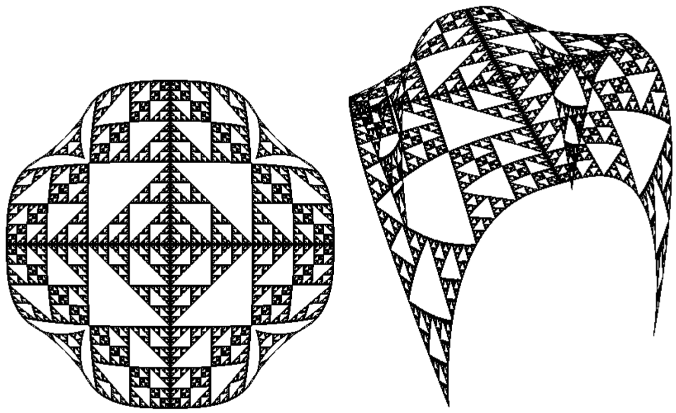

Figure 1: A top and perspective views of a fractal bicubic surface generated by an L-system

Abstract

Subdivision surfaces are becoming an important tool in Computer Graphics. They can be found in modern software as well as implemented in GPUs. However, their description is complex, and simple relations, as the inherently parallel manner of rewriting, are obfuscated by the indexing scheme.

We propose using L-systems for tensor product surfaces subdivision description. The parallel rewriting helps us to merge the parallel manner of a surface subdivision. We demonstrate their functionality on Bézier bicubic and rational Bézier bicubic surfaces. We show that the power of L-systems can be easily applied to this scheme which is intuitive and easy to control. Parametric L-systems allow us to manage the level of subdivision and Open L-systems help to generate adaptive surfaces where the level of subdivision is controlled by an external condition. The ε-rule helps to generate surfaces with holes and in this way we can emboss classical fractals to 3D surfaces.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation – Display algorithms— [I.3.7]: Computer Graphics—Three Dimensional Graphics and Realism – Virtual Reality

Keywords: L-systems, Procedural Modeling, Geometric Modeling, Meshes, Surface Refinement, Fractals

*e-mail: kolinger@kiv.zcu.cz
†e-mail:app@email.cz
‡e-mail:bbenes@purdue.edu

1 Introduction and Related Work

Subdivision surfaces are becoming an important modeling option in Computer Graphics. The classical description by indices, however, obfuscates the parallel process of subdivision. We show that Lindenmayer systems, or shortly L-systems, that are suitable for linear structures description, can can be extended to describe tensor product surfaces.

L-systems have been originally developed for description of process of cell division [Lindenmayer 1968]. The linear L-systems were later extended by Prusinkiewicz [Prusinkiewicz 1986] to bracketed L-systems and used for topological and geometrical description of plants. L-systems were recently extensively developed and they were shown as an extremely powerful instrument for description of growing structures [Prusinkiewicz et al. 1993a], plants interacting with their environment [Měch and Prusinkiewicz 1996; Prusinkiewicz et al. 1993b], and even for description of plant competition for space in entire ecosystems [Lane and Prusinkiewicz 2002].

L-systems were also used for variety of non-biological approaches, such as river shape and branching generation [Prusinkiewicz and Hammel 1993] and for subdivision curves generation and description [Prusinkiewicz et al. 2003].

Geometrical application has related L-systems with an area of subdivision curves and surface that has been of a growing interest in recent years. Subdivision curves and surfaces can be found canned in graphical hardware and they are also successfully used in commercial packages such as Maya. Survey of subdivision surfaces can be found in [Schroeder 1988] and in [Zorin et al. 1996].

Subdivision surfaces, however, suffer from a complicated mathematical description. The logically and visually apparent process of rewriting, which is elegantly described by L-systems, fails to be expressed easily by the classical indexing scheme. This has led the focus of researches to a search for an elegant scheme for subdivision surfaces that can be described in the language of rewriting systems.

Shape grammars apply a rewriting scheme and spatial relation for description of rewriting of a shape by another one. Shape grammars are a formal tool that has found its applications in architectural design [Szalapaj 2000] .

Graph grammars provide mathematical concept for graph metamorphosis and transformations. To our knowledge graph grammars were not extensively used in Computer Graphics although they have very strong potential. The two graph and mesh transformation approaches that were recently published are described in the following paragraphs.

One of the first approaches for application of mesh subdivision is described in [Velho 2003]. So called *Stellar subdivision grammars* use a description of the neighborhood of a vertex by the mathematical concept of the Stellar subdivision theory. Its direct application is shown to be viable for description of the major subdivision schemes used in computer graphics. There are two principal limitations of this concept. The neighborhood of a vertex is limited only to the first immediate neighbors and the Stellar subdivision grammars work only for subdivision surfaces. Topologically complicated structures, such as branches that are natural to L-systems, cannot be described.

So called *vertex-vertex* or shortly `vv` *programming language* operates on a polygonal meshes and was shown to be efficient for subdivision surfaces description in [Smith et al. 2003]. The `vv` programming language is implemented as an extension of C++ and describes an immediate vertex neighborhood by a concept of graph rotation system. All vertices are stored in one dimensional array in a predefined order. A set of examples in [Smith et al. 2003] shows that the `vv` programming language can describe classical subdivision schemes. There are two principal disadvantages of the `vv` programming language. First is its limitation to the one immediate neighbor. The second is its complex semantics and the need for additional data structures that obfuscate the parallel nature of the rewriting process.

The `vv` programming language was recently used to describe growing structures in [Smith and Prusinkiewicz 2004]. A model of Korn-Spalding cell division and the apical meristem were simulated. A simple physically and biologically based models influence the subdivision scheme that result in a complex three dimensional branching structures.

The principal advantage of L-systems, description of the left and the right context, determines them for a linear structures description. L-systems usually fail when trying to describe more complicated structures, such as surfaces, but they were successfully used for description of contours of leaves that are topologically identical to a disc [Prusinkiewicz and Lindenmayer 1990].

The main contribution of our paper is the description of tensor product surfaces by means of L-systems. L-systems were originally designed for linear dynamic processes modelling. We show that they can be also used for an efficient description of complex surfaces. We demonstrate this feature on Bézier surfaces but our approach is extendable to any tensor product surface.

We show that L-system representations provide new modelling possibilities which could be difficult to achieve with the classical matrix surface representation. To give a specific example it is not easy to generate a fractal tensor product surface in the classical matrix representation. Our results demonstrate that L-systems can be easily extended to describe tensor product surfaces.

When represented as L-systems, tensor product surfaces may have some unusual properties, which would be difficult or impossible to achieve in the classical index-based schemes: branching, enabling to create non-manifold objects and surfaces with holes, bidirectional communication with environment, context-related features in surfaces, stochastically influenced surfaces. This enables the surfaces to create models with non-deterministic features and a possibility to incorporate a plan for growth or development into geometric models. Some of these features have been achieved in our implementation and they are described in this paper.

The contents of our paper is as follows. Next section describes the previous and the related work from the formal description of rewriting processes and application of L-systems to subdivision curves [Prusinkiewicz et al. 2003]. The same section describes the definition of rewriting by means of classical context-free non-parametric L-systems and continues by description of Open L-systems with the ε-rule. Section 3 presents examples and results. The last Section 4 concludes the paper.

2 L-systems and Tensor Product Surfaces

We will not review the theory of L-systems and we refer reader to [Prusinkiewicz and Lindenmayer 1990] for their fundamentals. The application of L-systems to subdivision curves is described in [Prusinkiewicz et al. 2003]. Open L-systems is another fundamental extension that we use in our paper and its description can be found in [Měch and Prusinkiewicz 1996].

2.1 A Short Digression to Subdivision Curves

One of the oldest and classical applications of L-systems to subdivision curves is its description of the Koch's snowflake [Mandelbrot 1982]. The non-context and deterministic rewriting leads to the classical deterministic fractal.

Recently, L-systems were used for description of more complicated subdivision curves [Prusinkiewicz et al. 2003]. The idea is to use a context-sensitive parametric L-system to orchestrate the rewriting process. Suppose a control vertex $v = (x, y, z)$. The example rule

$$P(v_l)\,\langle P(v)\rangle\,P(v_r) \rightarrow P(\frac{1}{4}v_l + \frac{3}{4}v)P(\frac{3}{4}v + \frac{1}{4}v_r) \qquad (1)$$

uses the left and the right context vertices v_l and v_r to determine a new position of the vertex and its neighbors. The use of L-systems for subdivision curves opens door for new possibilities as recursive or subdivision way of computing is often used in geometric modelling.

2.2 Tensor Product Surfaces

We will concentrate on matrix representation as it enables us to generalize the L-system description to this kind of surfaces. We will show this extension on Bézier bicubic surfaces due to their simplicity. This is not a limitation because higher degree surfaces are expressed by higher number of control vertices and different equations, but are principally the same.

The Bézier bicubic tensor product surface is given by the sixteen control vertices $v_{0,0},\ldots,v_{3,3}$ that are stored in a matrix P:

$$P = \begin{bmatrix} v_{0,0} & v_{0,1} & v_{0,2} & v_{0,3} \\ v_{1,0} & v_{1,1} & v_{1,2} & v_{1,3} \\ v_{2,0} & v_{2,1} & v_{2,2} & v_{2,3} \\ v_{3,0} & v_{3,1} & v_{3,2} & v_{3,3} \end{bmatrix} \tag{2}$$

From the matrix P another matrix can be computed using the de Casteljau scheme. We denote the original matrix by a subscript k and the new matrix by $k+1$. The new matrix is described from the original one by *rewriting* so

$$P_k \rightarrow P_{k+1} \rightarrow \ldots$$

The limit of the rewriting process is the surface itself.

Let us denote the matrix dimension by $dim(P) \subset Nat^2$. The axiom of the rewriting, the matrix $dim(P_0) = 4\times 4$, the matrix $dim(P_1) = 8\times 8$, and matrix $dim(P_k) = 2^{k+2}\times 2^{k+2}$.

One rewriting step can be expressed as a matrix composition and is described by the following equation:

$$P_{k+1} = \left[\begin{array}{c|c} LP_kL^T & LP_kR^T \\ \hline RP_kL^T & RP_kR^T \end{array}\right] = \left[\begin{array}{c} L \\ \hline R \end{array}\right] P_k \left[\begin{array}{c} L^T \\ \hline R^T \end{array}\right]^T. \tag{3}$$

Symbols L and R denote the matrices of Bézier coefficients:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/8 & 3/8 & 3/8 & 1/8 \end{bmatrix} \tag{4}$$

$$R = \begin{bmatrix} 1/8 & 3/8 & 3/8 & 1/8 \\ 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

P_{k+1} is the matrix of newly computed control points from the k-th level of the upper left, lower left, upper right and lower right part of the surface, respectively. The complicated term

$$\left[\begin{array}{c|c} LP_kL^T & LP_kR^T \\ \hline RP_kL^T & RP_kR^T \end{array}\right]$$

is a kind of matrix composition. The new matrix has double the resolution of the original P matrix and is obtained by placing the four matrix multiplications LPL^T, LPR^T, RPL^T, and RPR^T on the corresponding positions.

As can be seen in the equation (3) the control points form a linear combination of the control points from the previous level. This can be captured by a rewriting process that is the essence of L-systems.

One of the standard ways to extend L-systems is symbol overloading. In this manner the module *vertex* is introduced in the `L+C` programming language [Prusinkiewicz et al. 2003]). We overload matrices to work with them in the standard mathematical manner. In this way the equation (3) can be expressed as:

$$p_1 : P : cond. \rightarrow \left[\begin{array}{c|c} LPL^T & LPR^T \\ \hline RPL^T & RPR^T \end{array}\right] \tag{6}$$

This rule expresses exactly the essence of rewriting. The module P is to be rewritten by a multiplication of P and L, R matrices. The matrices are composed and result in the matrix with higher dimension. The term $cond. \subset Bool$ is the condition that must be fulfilled to apply the rewriting.

2.3 Parametric L-systems and Controlled Rewriting

Some applications require controlling the level of recursion. This can be achieved blindly by applying the rule n times or can be achieved more elegantly by parametric L-systems [Prusinkiewicz and Lindenmayer 1990].

The module P is extended by the parameter *level* to $P(level)$ that indicates the level of recursion. The axiom of the extended L-system (6) will have form $P(0)$, the maximum level of recursion is denoted by *max* and the level of recursion controlling rule is:

$$\begin{gathered} p_1 : P(level) : level < max \rightarrow \\ \left[\begin{array}{c|c} LP(level+1)L^T & LP(level+1)R^T \\ \hline RP(level+1)L^T & RP(level+1)R^T \end{array}\right] \end{gathered}$$

2.4 Open L-systems and Adaptive Refinement

The principal advantage of subdivision surfaces is the ability to apply them only to those areas that require it. The adaptive refinement is usually controlled by some external condition that is implemented as a call of a method of a class.

The condition that decides whether the recursive subdivision is going to progress or will be stopped depends usually on the implementation and some of the typical criteria include:

- The projected area of the bounding box of the control points is below certain limit.
- The set of the control points is far from the camera.
- The surface is entirely flat or its curvature is small.

These clues are used as the stop condition of rewriting. For example, if the projected area of the bounding box is below certain predefined level; if the matrix is too far from the camera; or if the surface is flat; further subdivision is not supposed to bring any visual information and the process of subdivision is terminated.

The adaptive subdivision can be incorporated into L-system–based rewriting by means of the query symbol that were introduced with so called Open L-systems (see [Měch and Prusinkiewicz 1996] and [Prusinkiewicz et al. 1993b]).

Query symbol is denoted by the question mark that precede the module of rewriting system, e.g., $?E(p_1,\ldots,p_n)$ and its parameters p_i, $i = 1,\ldots,n$ are set according to the external influence. [Měch and Prusinkiewicz 1996] show how growing plants can be limited by space using query symbols.

We can extend the approach of subdivision surface to the adaptive subdivision based on the above mentioned distance criteria to L-systems. Suppose the query module $?E(d)$ that receives as its parameter the distance of the mass of density of the matrix of control points from the camera. The query module will be coupled with the matrix P and will be always in the pair $?E(d)P$. The rule for the adaptive refinement have the form:

$$\begin{gathered} p_1 : ?E(d)P(level) : d \leq \varepsilon \rightarrow \\ \left[\begin{array}{c|c} LP(level+1)L^T & LP(level+1)R^T \\ \hline RP(level+1)L^T & RP(level+1)R^T \end{array}\right] \end{gathered} \tag{7}$$

The parameter d is set to the actual distance and is parsed to the query module $?E(d)$ before rewriting. Once the value is set, the condition is evaluated. If the mass of density of the set of control points represented in the matrix P is closer than the minimum distance ε, the rule is applied.

An example of the application of the rule (6) is displayed in Figure 2. A matrix of sixteen control points is successively refined by the application of the L-system. The condition that was used in this case was the flatness of the control points defining the surface and it is clearly visible that the rewriting rule is not applied to the flat areas. It is worth mentioning that the condition of the flatness of the surface can be, in fact, calculated by the L-system itself and not read as the external value. This remark does not change the generality of our approach.

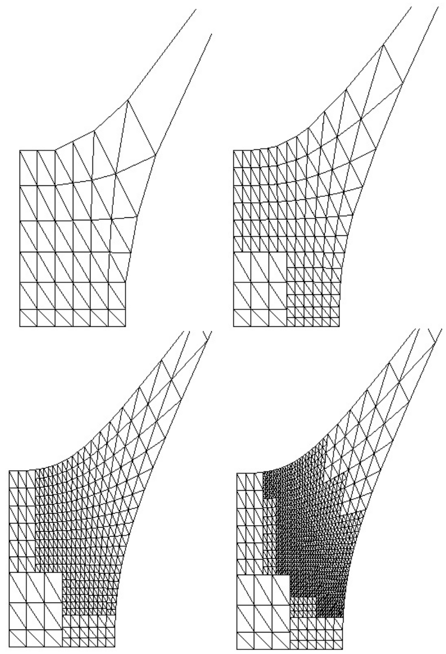

Figure 2: Adaptive refinement of the Bézier patch by means of Open L-system

2.5 Rational Surfaces

Bézier bicubic surfaces does not enable to create a quadrics such as a sphere or two parallel surfaces in Euclidean coordinate system. This disadvantage is not shared by the rational Bézier surfaces that are defined in the homogenous coordinate system. We can apply the above described rewriting by means of L-systems to rational bicubic Bézier surfaces immediately.

The input control vertices $v_i = (x, y, z)$, given in the rational Bézier surfaces by Euclidean coordinates and rational weights w_i, have to be converted into homogeneous coordinates $v_i'(x \cdot w_i, y \cdot w_i, z \cdot w_i, w_i)$. This transformation enables us to use equation (3) for rational Bézier surfaces with only one difference: Equation (3) is used for four homogenous coordinates instead of the three Euclidean coordinates. For visualization purposes, the points should be converted back to the Euclidean coordinates, by dividing each coordinate by the homogenous coordinate w if $w \neq 0$.

An example of a cylinder generated by the rewriting system is displayed in Figure 3. The axiom of rewriting was: *ABCD*, where vertices from each matrix formed one side of the opened cube. The surface is result of the application of the rule (3).

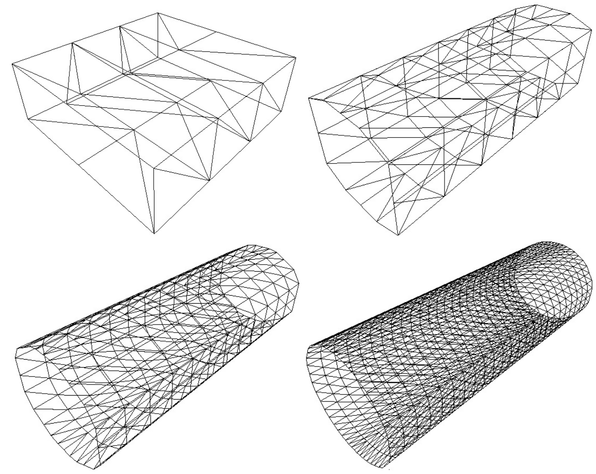

Figure 3: Rational Bézier bicubic surface showing a cylinder as the result of rewriting by L-system

2.6 The ε-rule

Parallel string rewriting systems or grammars use the concept of the empty symbol that is used to decrease the length of the generated sequence of modules or as the stop condition of rewriting. The ε-rule has form:

$$p_3 : P : cond. \rightarrow \varepsilon \tag{8}$$

and the meaning is that the module from the left side is not replaced by any module and is left empty.

This is an interesting feature that can be achieved by applying L-systems to surface rewriting. The resulting surface can be fractional and, in the limit of rewriting, it can be a fractal surface. The ε-rule brings us a possibility to generate surfaces with holes, with user-defined size and frequency.

For example, applying the ε-rule as late as in the last level of recursion produces a surface with a regular pattern of holes. This is described by the following L-system:

$$\begin{aligned}
&p_1 : P(level) : level < max \rightarrow \\
&\qquad \left[\begin{array}{c|c} LP(level+1)L^T & LP(level+1)R^T \\ \hline RP(level+1)L^T & RP(level+1)R^T \end{array}\right] \\
&p_2 : P(level) : level == max \rightarrow \\
&\qquad \left[\begin{array}{c|c} LP(level+1)L^T & \varepsilon \\ \hline \varepsilon & RP(level+1)R^T \end{array}\right]
\end{aligned}$$

A fractal surface will be obtained for example by applying the following rules:

$$\begin{aligned}
p_1 : P &\rightarrow \left[\begin{array}{c|c} LAL^T & LBR^T \\ \hline RCL^T & RDR^T \end{array}\right] \\
p_2 : A &\rightarrow \left[\begin{array}{c|c} LAL^T & \varepsilon \\ \hline RAL^T & RAR^T \end{array}\right] \\
p_3 : B &\rightarrow \left[\begin{array}{c|c} \varepsilon & \varepsilon \\ \hline \varepsilon & \varepsilon \end{array}\right] \\
p_4 : C &\rightarrow \left[\begin{array}{c|c} LCL^T & \varepsilon \\ \hline RCL^T & RCR^T \end{array}\right] \\
p_5 : D &\rightarrow \left[\begin{array}{c|c} LDL^T & \varepsilon \\ \hline RDL^T & RDR^T \end{array}\right]
\end{aligned}$$

The rule p_1 subdivides the surface four times and creates new modules - patches A, B, C, and D. They are rewritten by the rules p_2, p_3, p_4, and p_5. These rules, at the same time, write the corresponding pattern into the surface by cutting it by ε-rules. If we apply this rule to one patch of four on each level of subdivision, we obtain surface that is topologically equivalent to the Sierpinski triangle.

Figure 4 shows three and six rewritings of the original patch.

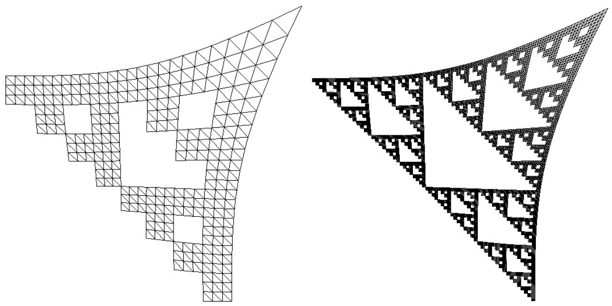

Figure 4: Sierpinsky triangle embossed into the Bézier bicubic surface by the ε-rules. Figure shows three and six iterations

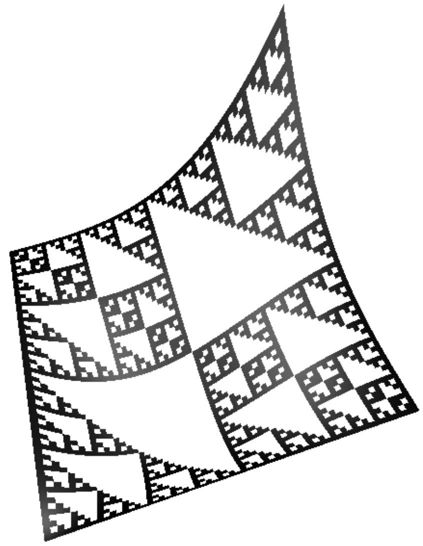

Figure 5: The double Sierpinski triangle

3 Results

We have implemented the Open parametric L-system for (rational) Bézier surfaces in Delphi under Windows. We have made also a parallel implementation in Java in two versions, distributed, based on PVM, and parallel, based on threads. Our results are just the first step to the L-system representation of the tensor product surface. Our experiments verify versatility of the L-system approach and show possibilities for the future research.

We have already shown that a substantial feature of subdivision-computed surfaces, adaptivity, can be easily achieved by L-system.

Another possible example combines the previously demonstrated properties. Let us say that we would like to get the double Sierpinski triangle, preserving the original four-sided size of the patch. To model it, we first duplicate the input control point matrix P and simulate it as two Sierpinski triangles. The first is generated exactly as described in Section 2.3, whereas the other is just its swapped version. Result is shown in Figure 5 and the rules producing the Sierpinsky carpet are as follows:

$$p_1 : P \rightarrow \left[\begin{array}{c|c} LAL^T & LBR^T \\ \hline RCL^T & RDR^T \end{array}\right]$$

$$p_2 : A \rightarrow \left[\begin{array}{c|c} LAL^T & LAR^T \\ \hline \varepsilon & RAR^T \end{array}\right]$$

$$p_3 : B \rightarrow \left[\begin{array}{c|c} LBL^T & LBR^T \\ \hline \varepsilon & RBR^T \end{array}\right]$$

$$p_4 : C \rightarrow \left[\begin{array}{c|c} \varepsilon & \varepsilon \\ \hline \varepsilon & \varepsilon \end{array}\right]$$

$$p_5 : D \rightarrow \left[\begin{array}{c|c} LDL^T & LDR^T \\ \hline \varepsilon & RDR^T \end{array}\right]$$

Bracketed L-systems show a new opportunity for the generation of non Genus zero surfaces. The branching scheme in the form $P \rightarrow P[+P][-P]P$ is applied to the surface patch as demonstrated in the example in Figure 6. Special care must be taken for the surface continuity. This need is the result of the multi-level nature of the rewriting process. The patches are treated as the whole, whereas surface continuity is defined by the tangent vectors to the surfaces that depend on the control vertices. We define the surface continuity as the global property and during the rewriting the control vertices are copied correspondingly. In the case of the Bézier surfaces either one or two columns of the matrix are copied into the new one to assure C^0 or C^1 continuity respectively. Future work should address this problem.

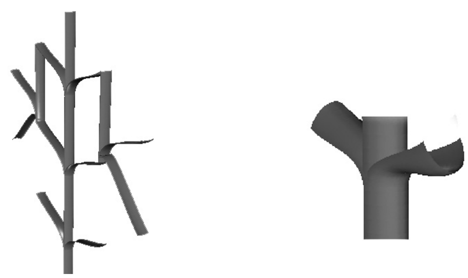

Figure 6: Branching structures as the result of tensor product surface rewriting

4 Conclusions and Future Work

We have demonstrated that L-systems that are considered as highly suitable for linear subdivision structures such as plant models, cell subdivision, or subdivision curves, can be easily adopted to tensor product surfaces. We have demonstrated this property by subdividing Bézier bicubic patch and rational Bézier patch and we have shown that this approach is able to generate classical structures such as cylinders as well as fractals.

Classical conditions, such as terminating the process of subdivision after certain amount of steps, can be achieved by parametric L-systems. The parameter controls the level of recursion.

By extending this approach to Open L-systems we are able to add adaptivity to the process of subdivision. Patches are subdivided only if an external condition is fulfilled. In this way a patch with an adaptive level of refinement can be generated.

We have shown that our approach is able to generate variety of classical examples.

Finding an intuitive and versatile description of subdivision surfaces is an important task. An inherently parallel process can be hardly described by indexing schemes and the equations describing it are becoming hard to understand even in the case of simple relations. We believe that the future work should go in the way of merging existing approaches, such as L-systems and subdivision surfaces, rather than creating new tools, theories, and models.

The potential of L-systems enables our approach to be extended to the description of "living" structures, such as biological tissues that could react to external conditions, such as intensity of light, gravity, varying CO_2 etc. Another open work is application to structures that are topologically different to a disc or disc with hole(s), i.e., generating branching 3D structures.

5 Acknowledgments

This work was supported by the project No. LC 06008 and by the Envision Center for Data Perceptualization of Purdue University.

References

LANE, B., AND PRUSINKIEWICZ, P. 2002. Generating Spatial Distribution for Multilevel Models of Plant Communities. In *Proceedings of Graphics Interface'02*, vol. I, 69–80.

LINDENMAYER, A. 1968. Mathematical Models for Cellular Interaction in Development. *Journal of Theoretical Biology Parts I and II*, 18, 280–315.

MANDELBROT, B. 1982. *The Fractal Geometry of Nature*. W.H.Freeman, San Francisco.

MĚCH, R., AND PRUSINKIEWICZ, P. 1996. Visual Models of Plants Interacting With Their Environment. In *Proceedings of SIGGRAPH '96*, ACM Press, vol. 30(4) of *Annual Conference Series 1996*, 397–410.

PRUSINKIEWICZ, P., AND HAMMEL, M. 1993. A Fractal Model of Mountains with Rives. In *Proceedings of Graphics Interface'93*, vol. 30(4), 174–180.

PRUSINKIEWICZ, P., AND LINDENMAYER, A. 1990. *The Algorithmic Beauty of Plants*. Springer–Verlag, New York. With J.S.Hanan, F.D. Fracchia, D.R.Fowler, M.J.de Boer, and L.Mercer.

PRUSINKIEWICZ, P., HAMMEL, S., AND MJOLSNESS, E. 1993. Animation of Plant Development. In *Proceedings of SIGGRAPH '93*, ACM Press, vol. I of *Annual Conference Series*, 351–360.

PRUSINKIEWICZ, P., JAMES, M., AND MĚCH, R. 1993. Synthetic Topiary. In *Proceedings of SIGGRAPH '94*, ACM Press, vol. I of *Annual Conference Series*, 351–358.

PRUSINKIEWICZ, P., SAMAVATI, F. F., SMITH, C., AND KARWOWSKI, R. 2003. L-system Description of Subdivision Curves. *International Journal of Shape Modeling 9*, 1, 41–59.

PRUSINKIEWICZ, P. 1986. Graphical Applications of L-systems. In *Proceedings of Graphics Interface'86*, vol. 22(4), 247–253.

SCHROEDER, P. 1988. Subdivision for Modelling and Animation. In *"SIGGRAPH Course No.36"*, ACM Siggraph.

SMITH, C., AND PRUSINKIEWICZ, P. 2004. Simulation Modeling of Growing Tissues. In *Proceedings of the 4th International Workshop on Functional-Structural Plant Models,*, 365–370.

SMITH, C., PRUSINKIEWICZ, P., AND SAMAVATI, F. F. 2003. Local Specification of Surface Subdivision algorithms. In *AGTIVE*, 313–327.

SZALAPAJ, P. 2000. *CAD Principles of Design, An Analytical Approach to the Computational Representation of Architectural Form*. Architectural Press.

VELHO, L. 2003. Stellar Subdivision Grammars. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 188–199.

ZORIN, D., SCHROEDER, P., AND SWELDENS, W. 1996. Interpolating Subdivision for Meshes with Arbitrary Topology. In *Proceedings of SIGGRAPH 1996*, ACM SIGGRAPH, 1996, 189–192.