A New Conception in Constructive Branching Structures and Leaves using L-system

Yasser M. Abd El-Latif
Mathematics Department (Computer Science)
Faculty of Science, Ain Shams University
Abbassia 11566, Cairo, Egypt
Y.AbdelLatif@sci.asu.edu.eg

Received 9 August 2010; Revised 2 September 2010; Accepted 20 September 2010

One of the important open problems in modeling plants is the extension of subdivision algorithms to branching structures. Most of the applications use the concept of L-system to produce branching structures as a sequence of lines and apply the subdivision scheme to appear as curves. In this paper, we explain how L-systems can be modified to produce branching structures. This is also very useful for generating the geometry of various shapes. The proposed technique, called an adaptive L-System, describes branching forms and leaves by making local curve without applying the subdivision steps. Advantages of the suggested algorithm over previous techniques are given. Validation of the algorithm are discussed, analyzed and illustrated by some experimental results.

Keywords: Computational Geometry and Object Modeling, L-system, and Subdivision Scheme

1. INTRODUCTION

One of the basic methods of generating fractal objects is Lindenmayer system, called L-system [Furmanek 2004]. L-systems have been introduced by Artistic Lindenmayer in 1968 [Lindenmayer 1968]. Lindenmayer noticed that complex biological plants and structures have recursive patterns and could be compactly represented through simple grammars (strings of text), called L-Systems [Prusinkiewicz and Lindenmayer 1990]. Lindenmayer displays the raw power of these simple L-Systems by producing beautifully magnificent plants and structures [Farkas et al.]. The framework of L-system consists of an initial structure and rewriting rules (or generating rules). The essence of development is parallel replacement using the rewriting rules. Starting from the initial structure, the L-system replaces each part of the current structure by applying the rule sequentially [Ijiri et al. 2006]. Along with the development of the theory, the L-system has been enriched with geometric aspects and has become a universal tool not only for modeling plants [Hanan 1992], but also for creating fractals whose shape is a function of time.

Another idea, besides the rule of rewriting, of L-system is the method of notation for the fractal's structure which is based on graphical interpretation of a string of characters. This method is known in literature under the name of the turtle graphics, which is created by Seymour Papert. The turtle graphics gives graphical interpretation of a string of characters in the form of commands given to a specially trained turtle. The method's tool is the turtle's tail which draws straight lines on the plane according to the received commands [Furmanek 2004].

L-systems capture the local character of subdivision rules and the dynamic character of the subdivision process [Prusinkiewicz et al. 2003] and this compatibility is closely related to the biological motivation of L-systems. They were originally proposed to describe the growth of linear structures made of locally communicating discrete elements. Subdivision can obviously be seen as an instance of such growth.

Several problems relating L-systems to subdivision are still open. For example, L-systems with affine geometry interpretation can also be used to generate fractals [Prusinkiewicz et al. 2003]. This gives the relation between subdivision curves and fractals and has been pointed out by Warren and Weimer [Warren and Weimer 2002]. The possibility of integrating fractals and subdivision curves using the same L-system formalism are interesting from the theoretical perspective and may have useful applications.

In the geometrical output of the L-system, branching structures are represented by lines. One of the most recent methods to convert these lines to curves is by using the subdivision schemes. The definition and generation of smooth curves and surfaces specified by a small set of control points is a fundamental problem of geometric modeling [Karem et al. 2006; El-Zahar et al. 2004]. One class of solutions is based on the concept of subdivision. Subdivision schemes for curves are defined by a set of rules that take a set of control points as input and produce a new refined set of control points as output. Subdivision is a recursive procedure that yields a limit curve. The goal of the paper is to introduce a new technique using L-system that produces branching structures and leaves with curves instead of lines and so, no need to the subdivision steps (Figure 1).

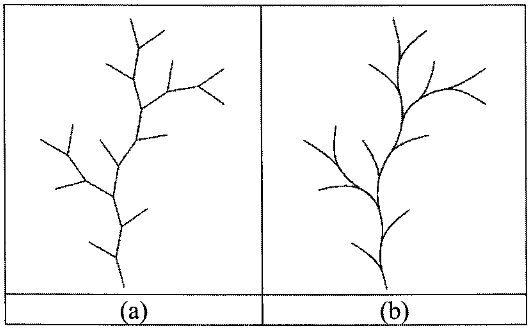


Figure 1. A branching structure (a) and the result of its smoothing (b).

One of the challenges in the modeling of natural phenomena is the synthesis of realistic images of plants. Recently, several methods, including L-systems (Lindenmayer 1968), were successfully applied to model branching plant structures (Prusinkiewicz 1988, 1990).

This paper proposes a new L-system technique called an adaptive L-system with affine geometry interpretation as a key solution for such a challenge. The suggested technique describes branching forms by making local curve without applying the subdivision steps as illustrated in Figure 1. The advantages of the proposed technique are that it produces good-quality for branching form approximations depending on the parameter of curvature during the construction process; simple to implement; reliable in which it can be applicable to any objects; optimizing the subdivision curve; and its computations are fast enough for an interactive environment.

The paper is organized as follows. Section 2 reviews the concept of subdivision curve and Chaikins algorithm. In Section 3, we give a brief summary of the L-system in two dimensions with some examples and explain how L-systems can be modified to produce branching structures and leaves. Some experimental results with discussions are given in Section 4. The time complexity of the subdivision algorithm and comparison with the complexity of our proposed technique are analyzed in section 5. Finally, Section 6 includes the conclusion of the paper and some directions for future work.

2. SUBDIVISION CURVE

A simple and useful set of geometric algorithms is subdivision curves [Chaikin 1974; Stollnitz et al. 1996; Warren and Weimer 2002]. A subdivision curve is a curve, composed of vertices connected with line segments, on which a refinement process can be iteratively applied. The refinement process adds vertices and alters the positions of existing vertices by affine transformations such that, at the limit, the curve is continuous and smooth.

The rules used to subdivide the curve are often depicted as a mask. This mask is a graphical representation of a transformation to the curve that depicts some segment of the curve and the weights applied to the vertices in an affine combination of the vertex positions to produce the new geometry. The mask is just a graphical representation, not a programming construct.

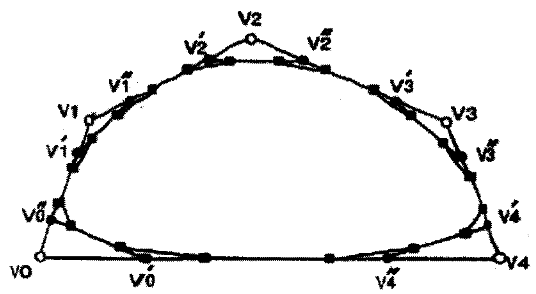


Figure 2. Generate B-spline curve from a polygon using Chaikin's Algorithm.

A curve subdivision is traditionally implemented by a matrix transformation. As given in [Chaikin 1974], L-systems can be used to define a curve subdivision algorithm. An L-system production is a concise form to express a subdivision rule using only local relations similar to the mask representation.

2.1 Chaikin's algorithm

The fundamental idea of subdivision curves return to Chaikins algorithm [Chaikin 1974], which generates a quadratic B-spline curve from a polygon by successively cutting its corners. Each subdivision generates two new points on each polygon leg at (1/4, 3/4).

For the polygon shown in Figure 2 with n vertices $(V_i)_{0<i<n+1}$ on polygon leg (V_i, V_{i+1}), two new points given by $V_i''=3/4V_i+1/4V_{i+1}$, $V'_{i+1}=1/4V_i+3/4V_{i+1}$ are created. The new generated points (V_i'', V'_{i+1}) corresponding to V_i $(0<i<n+1)$ are linked to form the new polygon. By repeating the subdivision process, quadratic B-spline curve can be found.

3. ADAPTIVE L-SYSTEM STRATEGIES

This section begins by presenting a method for constructing any curve defined by an arbitrary three points. The first and the third points determine the starting and ending of the curve. The curvature of this curve determines by the position of the middle point.

3.1 Curve Description

Constructing curve begins with a means of generating a parabola connecting three arbitrary points. The line between two points (X_1, Y_1) and (X_2, Y_2) can be drawn using the equations:

$$X=X_2t+X_1(1-t) \text{ and}$$
$$Y=Y_2t+Y_1(1-t)$$

This can be more compactly expressed as the single equation:

$$(X(t), Y(t))=(X_2, Y_2)t+(X_1, Y_1)(1-t) \tag{1}$$

For t between 0 and 1, the two equations produce points on the straight line connecting the two points. This method is known as parametric representation. The X and Y values of the line are defined as functions of a third value, the parameter t.

Parametric representation uses three points $P_1=(X_1, Y_1)$, $P_2=(X_2, Y_2)$, and $P_3=(X_3, Y_3)$ to define a parabola. The parametric equations for a parabola are as follows:

$$\begin{aligned}(X(t), Y(t))=&(X_1, Y_1)(t_2-t)(t_3-t)/((t_2-t_1)(t_3-t_1))+\\ &(X_2, Y_2)(t_1-t)(t_2-t)/((t_1-t_2)(t_3-t_2))+\\ &(X_3, Y_3)(t_1-t)(t_3-t)/((t_1-t_3)(t_2-t_3))\end{aligned} \tag{2}$$

For numbers t_1, t_2, and t_3 $(t_1<t_2<t_3)$ and the parameter t starts at t_1 and goes

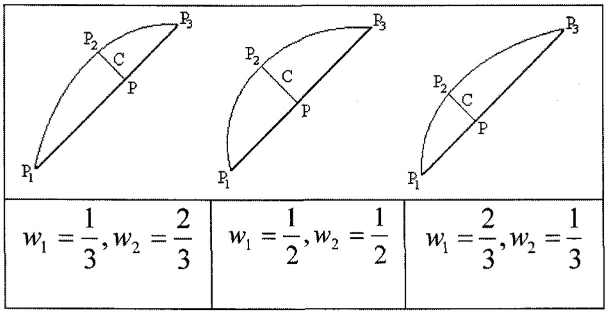


Figure 3. The curve that passes from P_1 to P_3, and depends on the position of P_2.

through t_2 to t_3. t_1, t_2, can t_3 have any convenient value as long as the relative order $t_1 < t_2 < t_3$ is maintained. Hence forth, points of this kind are called reference values. Note that for $t=t_1$, $(X(t), Y(t))=(X_1, Y_1)$ and similarly for $t=t_2$ and $t=t_3$. As t goes from t_1 to t_3, the points generated fall on the unique parabola connecting P_1, P_2, and P_3. In fact, the parabola is formed by generating a weighted average of the three points. Equation (1) is the parametric equation for a straight line with reference values $t_1=0$ and $t_2=1$.

It is obvious that the curvature of the curve that passes from P_1 to P_3 depends on the position of the middle point P_2. A method to generate algorithm for manipulating the position of P_2 is described in the following way:

(i) Computing the position of P on the edge P_1P_2.

(ii) Computing the perpendicular distance which determines the curvature of the curve, joins P_1 to P_3 passing through P_2, and the direction of the curve will be determined by the sign of a parameter multiplying in C.

To establish step (i), suppose we have the edge which joins points P_1 and P_3. The position of a new point $P=(X, Y)$, determined by a weight w_1 and w_2 of the two points P_1 and P_3 respectively, would then be:

$$P=w_1P_1+w_2P_3$$

$$\text{such that } w_1, w_2 \in \Re, \; w_1+w_2=1$$

To establish step (ii), we use the new point $P=(X, Y)$ as a result from step (i) in the following formula:

$$P_2(X_2, Y_2)=\begin{cases} (X, DC) & if \quad m=0 \\ \left(X-D\frac{C^2}{m+1}, \; Y-D\frac{mC^2}{m+1}\right) & otherwise \end{cases}$$

where m is the slope of the line P_1P_3, D indicates the direction of the curvature if +1 (or −1) is the curvature on the left side (or right side) of the line P_1P_3, and C represents the perpendicular distance between P and P_2, and its value corresponding to the curvature of the curve joining between P_1P_3. Figure 3 illustrates this process with different values of w_1 and w_2 corresponding to different shapes of the curves.

The combination of these two mechanisms ((i), (ii)) is sufficient to describe curvature for curves. For example, the specification of curves using the curvature parameter C is described below. Figure 4 shows a parabola connecting three arbitrary points with different values of curvatures C and the direction of the curvature depending on the sign of D.

3.2 Plant Parameter Description

An actual plant usually consists of many parts such as branches, trunks, leaves, and flowers. To simplify the modeling task without losing much of reality, branches, trunks, and leaves are considered in the next. We start by observing the **self-similarity** characteristic of the plant, and then take a slice of real branch structure as described in Figure 5.

Now, a brief introduction to L-systems and graph grammars are given [1]. The basic idea of the L-system is a rule of rewriting, also called a rule of replacing. A short list of commands, called axioms, by means of which the simplest fractals using the turtle graphics method can be created are as follows:

F: Move forward a step of constant length > 0, draw a line segment from the previous to the new position.
G: Move forward a step of constant length > 0, draw a curve from the previous to the new position with curvature C and left turn direction ($D = +1$).
H: Move forward a step of constant length > 0, draw a curve from the previous to the new position with curvature C and right turn direction ($D = -1$).
V: Construct a leave by move forward a step of constant length > 0, draw two curves from the previous to the new position with curvature C with right and left turn direction.
f: Move forward a step of constant length > 0 without drawing a line.
+: Turn left by constant angel δ.

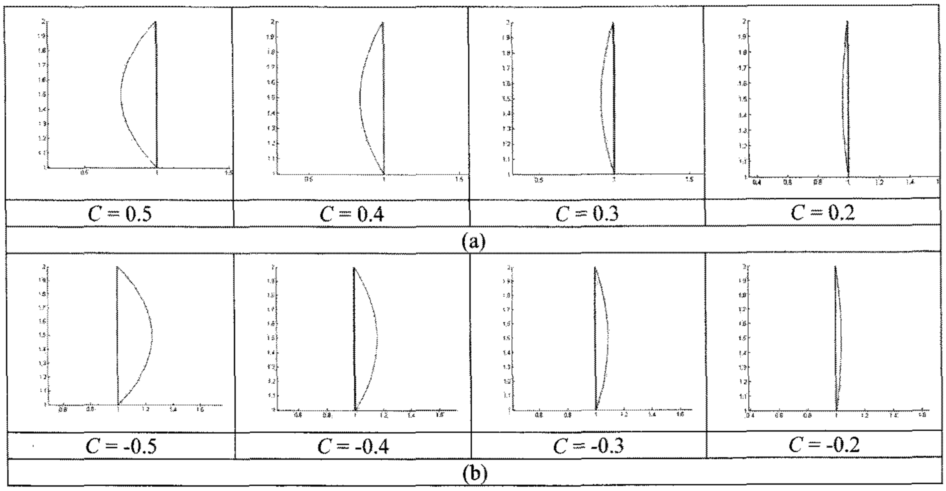


Figure 4. a parabola connecting P_1 to P_3 through P_2 (w_1=1/2, w_2=1/2) with different values of curvatures C, D=+1 (a), and D=−1 (b).

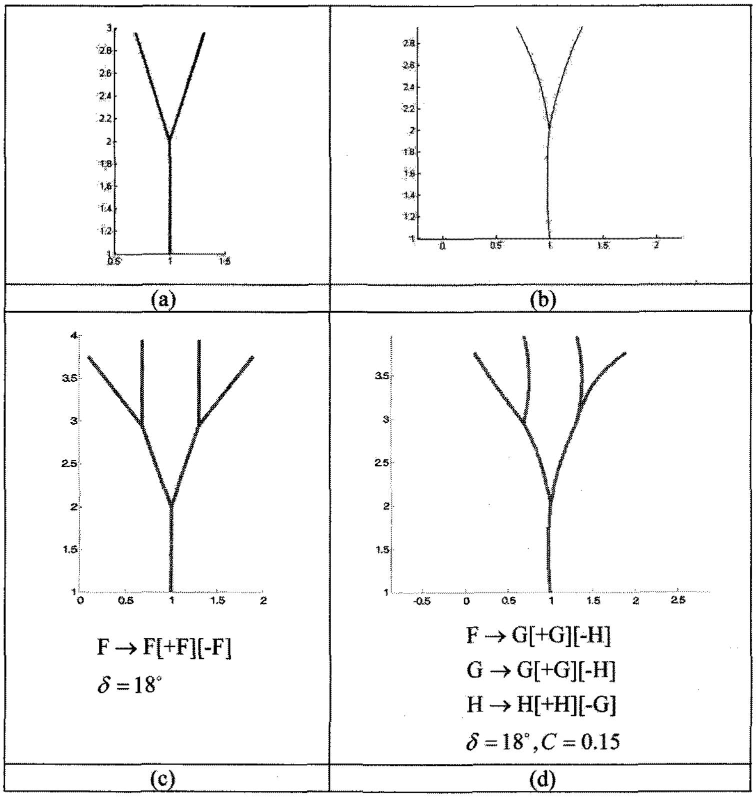


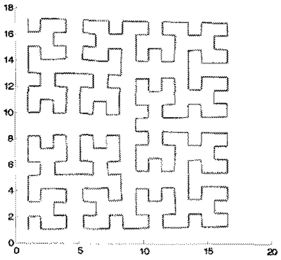
Figure 5. A tree using L-System with lines (a), (c) and with branching structures (b), (d).

−: Turn right by constant angel δ.
[: cause that the turtle's current state is remembered.
]: make the turtle come back to the state before the symbol [.
[...]: Between these symbols occur commands defining the construction of the branches.

The time complexity of L-system algorithm depends on the number of iterations of rewriting rule. For any given example, let k be the length of the longest rule in this example. Then we need at most $O(k^n)$ to calculate the final string of the axiom where n is the number of iterations, and $O(k^n)$ to draw the fractal plant according to the resulting final string of the axiom.

4. EXPERIMENTS AND COMPARISONS

Based on the suggested algorithm, several experimental results are performed over arbitrary rules of L-system with specific values for the curvature C, the direction of the curve D, and the angle δ. Models produced by the adaptive L-System are constructed and compared with models generated by using usual L-System and needed to apply subdivision schemes to produce smooth curve.



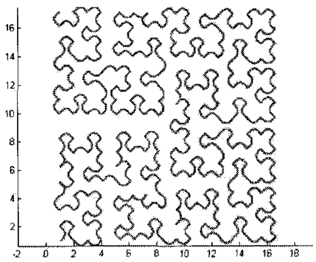
axiom = L

L → +RF - LFL - FR +

R → -LF + RFR + FL -

$\delta = 90^\circ$

(a) A Hilbert line



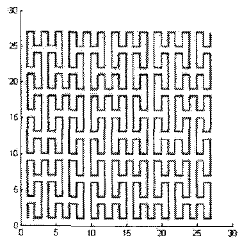
axiom = L

L → +RG - LGL - HR +

R → -LH + RHR + GL -

$\delta = 90^\circ, C = 0.6$

(b) A Hilbert curve



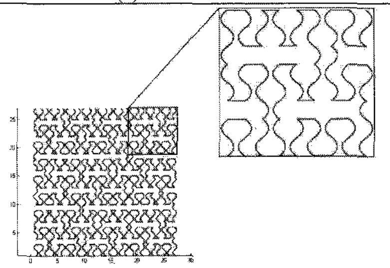
axiom = S

S → SFZFS + F + ZFSFZ - F - SFZFS

Z → ZFSFZ - F - SFZFS + F + ZFSFZ

$\delta = 90^\circ$

(a) A Peano S-Shaped line



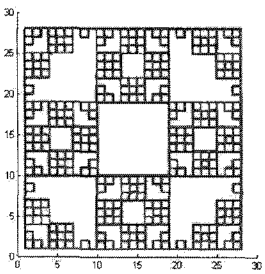
axiom = S

S → SHZGS + F + ZHSGZ - F - SGZHS

Z → ZGSHZ - F - SGZHS + F + ZHSGZ

$\delta = 90^\circ, C = 0.6$

(b) A Peano S-Shaped curve

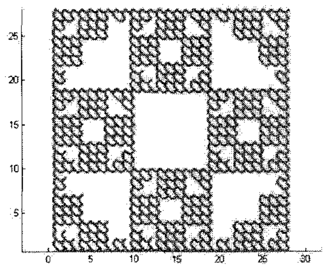


axiom = F + F + F + F

F → FF + F + F + F + FF

$\delta = 90^\circ$

(a) A Sierpinski Square line



axiom = G + G + G + G

G → GG + G + G + G + GG

$\delta = 90^\circ, C = 0.6$

(b) A Sierpinski Square curve

Figure 6. Different shapes with normal L-system (a), and with adaptive L-system (b).

Figures 5 and 6 (a)-(b) demonstrate a comparison between the given adaptive L-System and the usual one. Continuing in the direction of a general description for

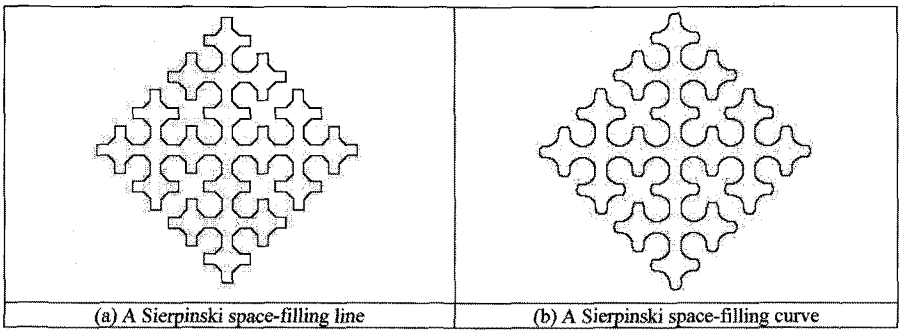


Figure 7. A Sierpinski space-filling curve (a) and its smooth version obtained using subdivision (b).

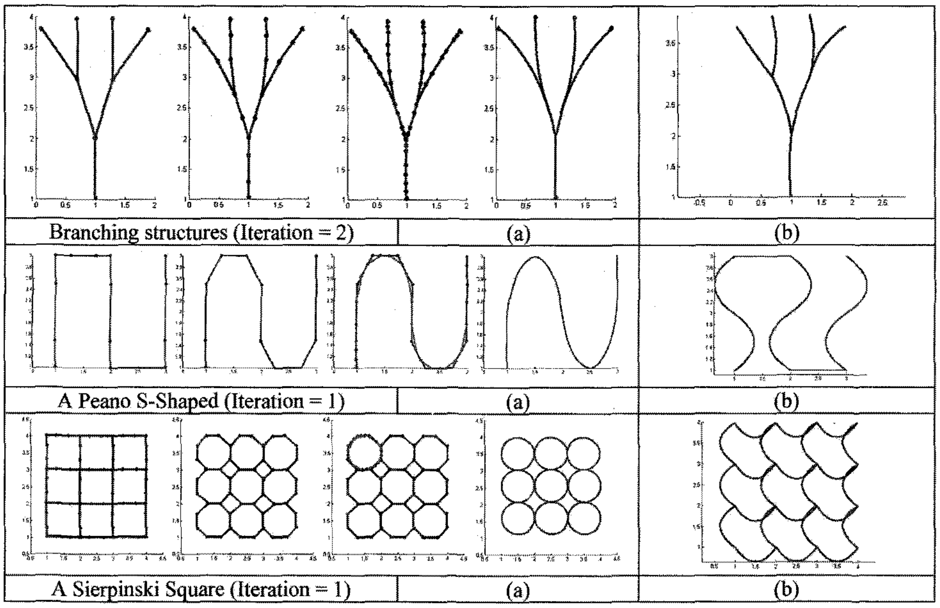


Figure 8. (a) Subdivision curves are used in an L-system, and (b) the adaptive L-system.

branch structures, we proposed to use L-systems as a way to specify lines as curves. Thus, articulation of the curvature represents an exercise in parameter *C*, and the exercise of this parameter usually produces a change in shape. For instance, Figure 7 shows a finite approximation of the Sierpinski space-filling curve [Prusinkiewicz et al. 2003], and the result of its smoothing using Chaikin subdivision. The resulting curve is a kolam pattern, a representative of patterns that were developed as folk art in India.

Figure 8(a) shows how subdivision curves are used in an L-system by using subdivision scheme. After applying a specific number of subdivision curve iterations, the line of L-system appears as curves. From the experimental results, one can observe that our proposed adaptive L-system scheme is smoothing and most shrinking

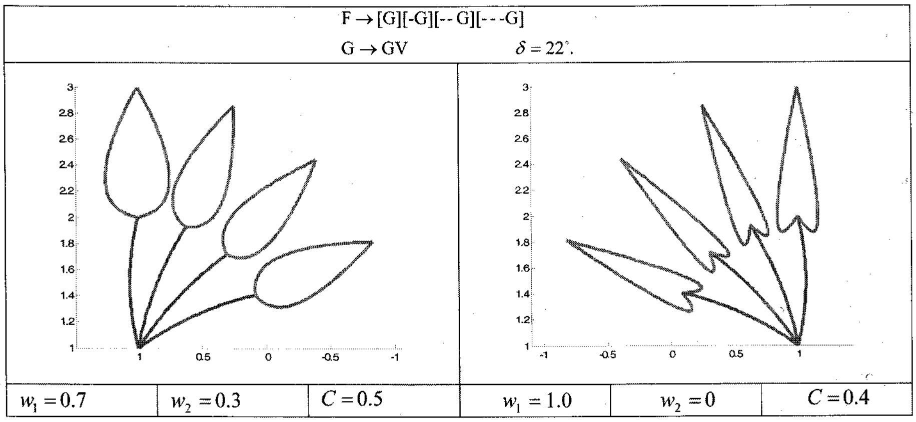


Figure 9. Construction leaves with different values of w_1, w_2, and C.

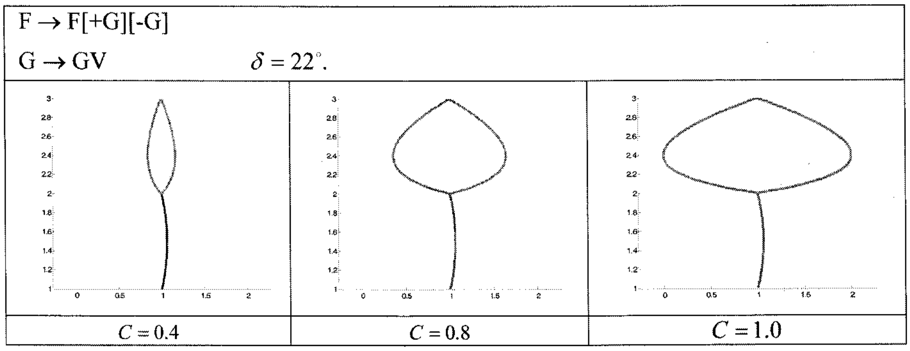


Figure 10. Construction of leaves with different values of curvature C, and w_1=0.6, w_2=0.4.

with no need to apply any subdivision schemes. This is why the proposed technique called "an adaptive" L-system. In addition the proposed technique controls the curvature that is above or down as shown in Figure 8(b).

In addition, the system uses additional rules for generation of curved branches, which can have a positive or negative curvature. This means that the generated curve structure may not be continuous at the point of branches. We should precisely explain the limitation of the system to produce continuous curve with a quality results. The

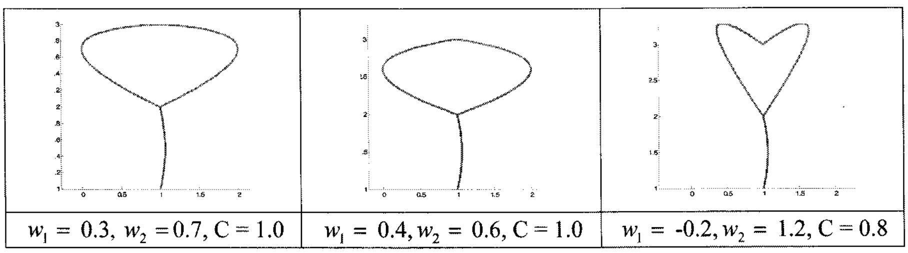


Figure 11. Construction of leaves with different values of w_1, w_2, and C.

limitation concentrates on the value of curvature parameter $C=|PP_2|\leq|P_1P_3|$ that is made sense to the size of the leaves. Connectivity is usually checked by the algorithm of L-system. The continuity of the line segments of the usual L-system can yield the continuity of the curves of the adaptive L-system. The curves in examples are perfectly continuous and the algorithm explicitly maintains connectivity during computation. Figures 9, 10, and 11 show how the adaptive L-system can produce the leaves as a unit with different shape according to the weights w_1, w_2, and the curvature C. The rules of L-system that is used to generate these leaves are attached with each figure.

To generate different shapes in different parts, a very useful property of the proposed technique is the ability to have the parameter function which has a high differences in value of the parameter C in different zones. This permits the creation of fine detail in one zone, without the need to deal with too many points all over the shape. The suggested method is to design the adaptive L-system which assigns a weight for each level.

5. ANALYSIS

This section analyzes time complexity of subdivision algorithm and compares it with the complexity of our proposed technique. The subdivision scheme defined two types of new edges created during the construction of the new polygon, namely, an edge-edge and a vertex-edge [Chaikin 1974]. In order to construct the connectivity of the new polygon, the number of edges and vertices have to be counted after every subdivision step.

Before step m, let E_m be the number of edges, and V_m be the number of vertices. After step m, let E_{m+1} and V_{m+1} be the new number of edges and vertices respectively. These numbers can be calculated according to the following formula:

$$V_{m+1}=2\,V_m, \qquad E_{m+1}=E_m+V_m$$

If the subdivision scheme is applied on the vertices obtained from the last stage of the rewriting process, then we need extra $O(E_m+V_m)$ time. Hence the total run time to generate object using L-syatem and apply subdivision scheme is $O(k^n)+O(k^n)+O(E_m+V_m)$. While our proposed technique needs only $O(k^n)$ time complexity to generate a smoothing, most shrinking curve of L-system is with no need to apply any subdivision schemes. Hence, the adaptive L-system has much lower complexity than using the subdivision surfaces with L-system, which endows our method with higher performance.

6. CONCLUSIONS AND FUTURE WORK

L-system is an efficient tool for fast generation of high quality objects and has been in the highlights of the computer graphics community for the last few years. Nowadays, L-system is a tool commonly used for modeling and simulating the growth of fractal plants. The problem with the L-system rules are that most of plants object represented by lines and after applying rule few steps, a number of generated lines will become huge and the number of edges grows exponentially, consequently difficult to manipulate. Due to memory and time restrictions, the number of applied recursive rules that can be performed, is relatively small. This paper proposes a new L-system technique called an adaptive L-system as a key solution for such a challenge. The objective of the given technique is to describe branching forms by making local curve with no need to apply the subdivision steps.

The use of curved lines increases the perception of realism in many models of organic forms [Bloomenthal 1995]. The formalism of L-systems is useful in describing branching forms, and therefore may provide a convenient general basis for subdividing branching curves as well. The new approach can be applied to other space filling curves. We give additional examples involving the Sierpinski and Peano curves as illustrated in Figure 6. The accessibility to a simple and general approach to create space filling curve algorithms may encourage researchers to experiment with a greater variety of curves. This is useful as different curves have different characteristics, and may be useful in particular applications. We are currently working on a rendering system for geometric modeling in 3 dimensions using L-system rules [Abd El-Latif et al. 2008].

REFERENCES

ABD EL-LATIF, YASSER M., ABOUSALEH, FATMA S., AND DAOUD, S. S. 2008. A New Vision of Fractal Geometry with Triangulation Algorithm, The International Conference on Computational Geometry - ICCG 2008 – Italy, proceedings of world academy of science, engineering and technology 34 October, 516–522.

BLOOMENTHAL, J. 1995. Skeletal design of natural forms. PhD thesis, University of Calgary, January

CHAIKIN, G. 1974. An algorithm for high speed curve generation, *Computer Graphics and Image Processing 3*, 346–349.

EL-ZAHAR, M. H., GHALEB, F. M., KAREM, H., AND ABD EL-LATIF, YASSER M. 2004. A novel optimized subdivision algorithm for modeling and animation, International Conference on Modeling and Simulation (MS 2004), USA, CA March, 436–636.

FARKAS, B. D., ROMANYSHYN, N., AND CLARY, P., L-Systems Construction Kit, Online: TUhttp://l3d.cs.colorado.edu/~ctg/classes/ttt2005/projectreports/Lsystemreport2.pdfUT

FURMANEK, P. 2004. Polyhedral covers based on l-system fractal construction, *The Journal of Polish Society for Geometry and Engineering Graphics 14*, 40-47.

HANAN, J. S. 1992. Parametric L-systems and their application to the modelling and visualization of plants, PhD thesis, University of Regina, June.

IJIRI, T., OWADA, S., AND IGARASHI, T. 2006. The Sketch L-System: Global Control of Tree Modeling Using Free-form Strokes, Smart Graphics, 138–146.

KAREM, H., GHALEB, F. M., AND ABD EL-LATIF, YASSER M. 2006. Towards Fast and Smooth Subdivision Surfaces Reconstruction, *International Journal of Computers and Applications*, Canada, 28 April No. 2, 170-176.

LINDENMAYER, A. 1968. Mathematical models for cellular interaction in development, parts I and II, *Journal of Theoretical Biology 18*, 280–315.

PRUSINKIEWICZ, P. AND LINDENMAYER, A. 1990. The algorithmic beauty of plants, Springer-Verlag, New York.

PRUSINKIEWICZ, P., LINDENMAYER, A., AND HANAN, J. 1988. Developmental models of herbaceous plants for computer imagery purposes, Proceedings of SIGGRAPH '88 Atlanta, Georgia, August 1-5 (1988), in Computer Graphics 22, 4, ACM SIGGRAPH, New York, 141–150.

PRUSINKIEWICZ, P., SAMAVATI, F., SMITH, C., AND KARWOWSKI, R. 2003. L-System Description of Subdivision Curves, *Internation Journal on Shape Modeling 9*(1): June, 41–59.

STOLLNITZ, E., DEROSE, T., AND SALESIN, D. 1996. Wavelets for computer graphics, Morgan Kaufman, San Francisco.

WARREN, J. AND WEIMER, H. 2002. Subdivision methods for geometric design, Morgan Kaufman, San Francisco, CA.

WARREN, J. AND WEIMER, H. 2002. Subdivision Methods for Geometric Design: A Constructive Approach. Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kaufmann, San Francisco.

Yasser M. Abd El-Latif He is an Egyptian, birth in Kuwait on 15/04/1971, and graduated from Faculty of Science "pure mathematics and computer science" on 1992. He obtained his MSC in computational geometry in 2000 and PhD in subdivision surfaces in 2005 from Faculty of Science, Ain Shams University, Cairo, Egypt.
He is a lecturer of computer science, Faculty of Science, ASU, Egypt. He has been publishing articles in computer graphics and computational geometry.
Dr Abd-El-Latif has a membership in Egyptian Mathematical Society (EMS) since 1995, and he is a reviewer in American Mathematical Society (AMS) since 2008 (MR: 2 455 503).