International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:1, No:10, 2007

Fractal Shapes Description with Parametric L-systems and Turtle Algebra

Ikbal Zammouri, and Béchir Ayeb

***Abstract*—**In this paper, we propose a new method to describe fractal shapes using parametric l-systems. First we introduce scaling factors in the production rules of the parametric l-systems grammars. Then we decorticate these grammars with scaling factors using turtle algebra to show the mathematical relation between l-systems and iterated function systems (IFS). We demonstrate that with specific values of the scaling factors, we find the exact relationship established by Prusinkiewicz and Hammel between l-systems and IFS.

***Keywords*—**Fractal shapes, IFS, parametric l-systems, turtle algebra.

I. Introduction

A lot of work has been developed and is still being developed in fractal modeling. Some of it is related to image compression and some other to natural phenomena simulation [1], etc. in this work, we are interested in fractal modeling and its use in shapes' creation. The objective is to have the possibility to create and manipulate original, varied and plastically interesting shapes. With this objective in mind, an extension of geometric modeling to fractal forms has been developed. Now in IFS formalism itself, classic forms (curves and smooth surfaces) can be represented [13]. Using IFS generalizations, some classical approaches have been extended to fractal shapes: Constructive geometry, free form shapes and geometric operations on shapes (deformation, combination and product) [12]. Other work concerns the topological representation of fractals, where the attractors' topological equivalences are deduced from the IFS' equivalences [3].

L-systems (or grammar based systems) are used to represent complex forms, mainly trees and vegetation elements. Other extensions were introduced such as the extension of l-systems to bracketed l-systems, context sensitive l-systems, parametric l-systems, etc [5]-[6]. Parametric context-sensitive-l-systems themselves have been extended and introduced with affine geometry interpretation to describe subdivision curves [9].

Prusinkiewicz and Hammel have presented equivalence between l-systems and IFS [7]. If we consider the resulting figures, the two formalisms are nearly equivalent. In some examples, they presented an l-system model, its equivalent IFS and the resulting geometric form. The two formalisms describe structures with self-similarity between the partial and the global forms.

We are interested in fractal shapes' modeling using IFS. In this work, we present a new tool to represent fractal shapes using parametric l-systems and turtle algebra. For this purpose, we have relied on previous work from Prusinkiewicz who shows the relationship between l-systems and IFS [7]-[8].

Manuscript received June 6, 2007.

Ikbal Zammouri is with the Research Unity PRINCE in Computer Sciences Department of Faculty of Sciences at Monastir, Rue de l'environnement, Monastir 5000, Tunisia (phone: 00216 73 500 274, fax: 00216 73 500 278, email: ikbal_z@yahoo.es).

Béchir Ayeb was with Computer Sciences Department of Sherbrooke University in Canada. He is now the director of Computer Sciences Department of Faculty of Sciences at Monastir in Tunisia, Rue de l'environnement, Monastir 5000, Tunisia (email: ayeb_b@yahoo.com).

II. Reminders

A. Iterated Function Systems

IFS represent a strong tool for fractal image analysis and synthesis. Seen its mathematical aspect, it offers simple methods for the generation of fractal shapes. IFS were first studied by Hutchinson in a pure mathematical context. Then, Barnsley [4] further developed this tool in fractal geometry and computer graphics. His idea was the coding of a fractal image with a set of contractive operators. This set of contractive operators is called IFS, and the figure coded by these ifs can be generated by iterating these operators.

IFS are mathematically defined by the following definitions and theorems:

***Definition 1*:** Let E be a complete metric space and G an application of X in X. G is said contractive in E if: $\exists\ s < 1$, $\forall$ p1, p2 in E \ d(G(p1), G(p2)) $\leq$ s d(p1, p2).

Theorem 1 (Fixed Point Theorem): Let be E a complete metric space. So be T: $X \rightarrow X$ a contractive application. Then, there exists a unique point $c \in X$ /T(c) = c. The point c is called the fixed point of T.

***Definition 2*:** Let be E a complete metric space**.** We call IFS every finite set $T = (T_1, \ldots, T_N)$ of contractive operators on X.

Theorem 2: For every IFS τ, there exists a unique and not empty compact $A / A = \tau A = T_1 A \bigcup \ldots \bigcup T_N A$. A is called attractor of τ, and is noted $A(\tau)$. It is possible to associate a

figure for every attractor. The attractor *A* possesses the self similarity property.

Barnsley defines the algorithm for the visualization of IFS as follows: Let be E a complete metric space, *H*(E) compact of E and τ an IFS.

For all K_0 in *H*(E), every sequence (K_n) defined by $K_{n+1} = \tau K_n$ converges to the attractor *A(τ)*.

He called it the determinist algorithm for IFS visualization.

B. L-systems

Formally, a parametric l-system is a set formed by an alphabet, some formal parameters, an axiom and a set of production rules: G = {v, s, ω, p}.

- v: Vocabulary, it describes the different classes of the modules.
- s: The set of parameters representing the modules' proprieties.
- ω: The axiom that represents the initial state of the organism.
- p: The set of production rules that describe the development of the organism.

We use ':' and '→' to separate the three components of a production, the predecessor, the condition and the successor.

The main concept of l-systems is the rewriting, which is a technique that permits the definition of complicated objects by the substitution of the parts of an initial object using a certain number of production rules.

The initial interpretation is in fact two dimensional. The state of the turtle is defined by (x, y, α) where x, y is the position of the Turtle and α is its orientation.

Given a step d and an increasing angle δ, the turtle responds to the next symbols:

F: The turtle affects a displacement of a distance d, and the new position is: x'= x + d * cos (α), y' = y + d*sin (α) and a segment is traced between (x, y) and (x', y').

f: A displacement d without tracing a segment between the points (x, y) and (x', y').

-: A rotation to the left by the angle δ. The new state of the turtle is (x, y, α -δ).

+: A rotation to the right by the angle δ. The new state of the turtle is (x, y, α + δ).

Here is an example:

-LS0: n =1
δ = 30
ω : F
p : F-> F+F--F+F

LS0 model generates the structure in Fig. 1.

Fig.1 2D displacement of the turtle

With these basic concepts of the "Turtle interpretation of strings" we can define complex and very rich geometric forms from an artistic point of view. We can cite the Koch curves or like tree shapes as examples.

III. Turtle Algebra

Some works have been developed to help to introduce the basic concepts of computer graphics and computer aided design to students using the turtle geometry. LOGO is the software that uses this tool [10]-[11].

A. Turtle Monoïd

In this paragraph we introduce an algebraic method used for the construction of scenes and the definition of modeling languages [2].

First we give definitions of the components of this algebra:

So be E one space, *M* is a monoïd acting on E.

So be the couple (f, T), with f is a figure i.e. f in H(E) $\bigcup$ {0}, T is a transformation in *M*.

We demonstrate that with the following operations:

(f, T) (f', T') = (f $\bigcup$ Tf', TT') and [(f, T)] = (f, I) = f, with I neutral element of *M*;

The set K (E) x *M* is a monoïd:

- The operation is associative:

(f, T) [(f', T') (f'', T'')] = (f, T) [(f' $\bigcup$ T'f'', T'T'')]
= (f, T) ((f' $\bigcup$ T'f'', T'T''), I)
= (f, T) (f' $\bigcup$ T'f'', T'T'')
= (f $\bigcup$ T (f' $\bigcup$ T'f''), T'T'')
= (f $\bigcup$ Tf' $\bigcup$ TT'f'', TT'T'')
= (f $\bigcup$ Tf', TT') (f'', T'')
= [(f, T) (f', T')] (f'', T''), so the operation is associative.

- The operation possesses a neutral element: (ϕ, I), with I neutral element of *M*:

(f, T) (ϕ, I) = (f $\bigcup$ T ϕ, T I)
= (f $\bigcup$ ϕ, T I)

= (f, T)

A scene is defined by a set of figures and operators that are get busted in the couples' set:

- f → (f, I), I is the neutral element of *M*.
- T → (ϕ, T); ϕ is the empty set.

B. Turtle Moving

The space is R^2.

The monoid *M* is the group of displacements in R^2.

This group is generated by translations T(x, y) and rotations R(α).

The turtle monoïd elementary couples are:

- ([0, 1]x{0}, T(1, 0)), translation pen down along x axis.
- (ϕ, T(1, 0)) translation pen up along x axis.

The brackets correspond to the operation: [(f, T)] = (f, I).

C. L-systems without Scaling Factors

Here is an example of a grammar based plant model LS1 given by Prusinkiewicz [7]:

- LS1: δ = 30
  ω : F
  p: F → F[+F]F[-F]

This model generates the like tree structures of Fig. 2.

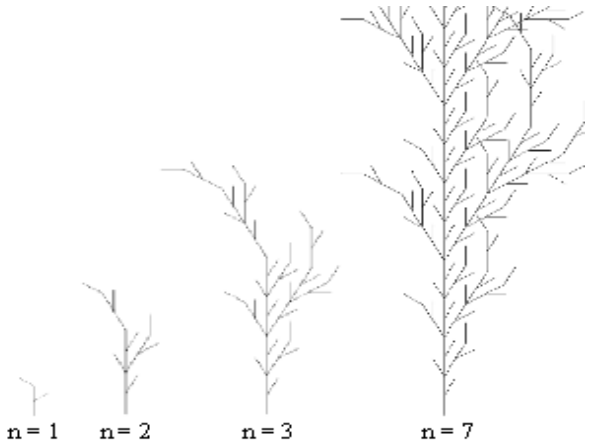

Fig. 2 Like tree structures correspondent to LS1 model

From this experimental result, we remark that tree structures correspondent to model LS1 increase when the iterations number n increases.

We can say that l-systems without scaling factors possess one characteristic of fractal geometry which is self similarity but not the second which the contraction propriety.

So we can conclude that l-systems in their initial definition don't generate fractal shapes, so they are not equivalent to IFS.

IV. Fractal Description

A. Fractal Trees

Now we introduce scaling factors r_1, r_2, s_1, s_2 < 1 to grammar based model LS1. We obtain LS2 model:

- LS2: δ = 30
  ω : F(1)
  p : F(w) → F(r_1*w)[+F(s_1*w)]F(r_2*w)[-F(s_2*w)]

w is a parameter introduced into the grammar.

Now we'll decorticate LS2 model, using the turtle monoïd algebra cited below.

We demonstrate that in the nth iteration, the symbol F is equivalent to:

$F_n(w)$ = (S(w) g_n,T(0,w)), with S(w) is the scaling operator by w factor and (g_n) sequence of figures defined by:

$$\begin{aligned} &g_{n+1} = T_1\ g_n\ T_2\ g_n\ T_3\ g_n\ T_4\ g_n \\ &\text{and} \\ &T_1 = S(r_1),\ T_2 = T(0, r_1)\,R(30)\,S(s_1), \\ &T_3 = T(0, r_1)\,S(r_2),\ T_4 = T(0,1)\,R(-30)\,S(s_2) \end{aligned} \quad (1)$$

T is the translation operator and R is the rotation operator.

Here is the demonstration:

On step n, the symbol F (in LS2 grammar) makes a displacement D_n and traces a figure f_n simultaneously.

On step n, F is equivalent to F_n = (f_n, D_n), with f_n is the associated figure and D_n is the associated displacement.

We note: $F_n(w)$ = (f_n(w), D_n(w)).

For n = 0, we have $F_0(w)$ = (f_0(w), D_0(w)), with:

$f_0(w)$ = S(w) g_0, g_0 = {0}x[0,1] and $D_0(w)$ = T(0,w), so the hypothesis is true for n = 0.

We suppose that the hypothesis is true for n:

$f_n(w)$ = S(w)g_n, $D_n(w)$ = T(0,w);

We demonstrate that it is true for n+1.

We pose: R_+ = R(δ) and R_- = R(-δ).

The expression of $F_{n+1}(w)$ respect to $F_n(w)$ is:

$F_{n+1}(w) = F_n(r_1*w)\,[(\phi, R_+)\,F_n(s_1*w)]\,F_n(r_2*w)\,[(\phi, R_-)\,F_n(s_2*w)]$;

If we develop the expressions:

$F_n(r_i*w) = (S(r_i*w)\ g_n,\ T(0,\ r_i*w))$ (the operation is true for n), using the turtle algebra, we'll have:

$(f_{n+1}(w), D_{n+1}(w)) = (S(r_1*w)\ g_n, T(0, r_1*w))$
$[(\phi, R_+)(S(s_1*w)\ g_n, T(0, s_1*w))]$
$(S(r_2*w)\ g_n, T(0, r_2*w))$
$[(\phi, R_-)\ (S(s_2*w)\ g_n, T(0, s_2*w))];$

$= (S(r_1*w)\ g_n, T(0, r_1*w))$
$[(R_+\ S(s_1*w)\ g_n, R_+\ T(0, s_1*w))]$
$(S(r_2*w)\ g_n, T(0, r_1*w))$
$[(R_-\ S(s_2*w)\ g_n, R_-\ T(0, s_2*w))];$
$= (S(r_1*w)\ g_n, T(0, r_1*w))\ (R_+\ S(s_1*w)\ g_n, I)$
$(S(r_2*w)\ g_n, T(0, r_2*w))\ (R_-\ S(s_2*w)\ g_n, I);$
$= (S(r_1*w)\ g_n \bigcup T(0, r_1*w)R_+\ S(s_1*w)\ g_n, T(0, r_1*w))$
$(S(r_2*w)\ g_n \bigcup T(0, r_2*w)\ R_-\ S(s_2*w)\ g_n,$
$T(0, r_2*w));$
$= (S(r_1*w)\ g_n \bigcup$
$T(0, r_1*w)\ R_+\ S(s_1*w)\ g_n \bigcup$
$T(0, r_1*w)\ S(r_2*w)\ g_n \bigcup$
$T(0, r_1*w)T(0, r_2*w)\ R_-\ S(s_2*w)\ g_n,$
$T(r_1*w)\ T(r_2*w));$

$= (S(r_1*w)\ g_n \bigcup T(0, r_1*w)R_+\ S(s_1*w)\ g_n \bigcup$
$T(0, r_1*w)\ S(r_2*w)\ g_n\ T(0, (r_1+r_2)*w)\ R_-\ S(s_2*w)\ g_n,$
$T(0, (r_1+r_2)*w);$

The symmetry condition of fractal geometry is verified: The total sum of rotation angles is equal to zero ([+F(s_1*w)] and [-F(s_2*w)] in model LS2).

To have the fractal contraction condition, we must conserve the translation vector, so we have: $r_1+r_2 = 1$.

We'll have $T(0, (r_1+r_2)*w) = T(0,w)$
and
$D_{n+1}(w) = T(0,w) = D_n(w);$

$f_{n+1}(w) = S(r_1*w)\ g_n \bigcup T(0, r_1*w)\ R_+\ S(s_1*w)\ g_n \bigcup$
$T(0, r_1*w)\ S(r_2*w)\ g_n \bigcup T(0,w)R_-\ S(s_2*w)\ g_n;$

$= S(w)\ S(r_1)\ g_n \bigcup S(w)T(0, r_1)\ R_+\ S(s_1)\ g_n \bigcup$
$S(w)\ T(0, r_1)\ S(r_2)\ g_n \bigcup S(w)\ T(0,1)\ R_-\ S(s_2)\ g_n;$
$= S(w)\ (S(r_1)\ g_n \bigcup T(0, r_1)\ R_+\ S(s_1)\ g_n \bigcup$
$T(0, r_1)\ S(r_2)\ g_n \bigcup T(0,1)\ R_-\ S(s_2)\ g_n);$
$= S(w)\ g_{n+1};$

So the sequence of figures generated by the parametric L-system LS2 is equivalent to:

$$g_{n+1} = T_1\ g_n\ T_2\ g_n\ T_3\ g_n\ T_4\ g_n$$
with
$$T_1 = S(r_1)$$
$$T_2 = T(0, r_1)\ R(30)\ S(s_1) \quad (2)$$
$$T_3 = T(0, r_1)\ S(r_2)$$
$$T_4 = T(0,1)\ R(-30)\ S(s_2)$$

The IFS equivalent to LS2 grammar model is given by equation (3).

$$T_1 = S(r_1)$$
$$T_2 = T(0, r_1)\ R(30)\ S(s_1)$$
$$T_3 = T(0, r_1)\ S(r_2) \quad (3)$$
$$T_4 = T(0,1)\ R(-30)\ S(s_2)$$
$$s_1, s_2 < 1$$
$$r_1 + r_2 = 1$$

If we give the values $r_1=r_2=s_1=s_2=1/2$ in (2), we'll obtain exactly the expression given in [7]. With these values we obtain the like tree fractal shape of Fig. 3.

Fig. 3 Like tree fractal shape correspondent to LS2 model for n = 7 and $r_1 = r_2 = s_1 = s_2 = 1/2$

B. Fractal Curves

Let be a general production rule p of a grammar based model (l-system) given by:

p: $F(w) \rightarrow R_0 F(r_1 * w) R_1 \ldots F(r_m * w) R_m$.

- Symmetry condition of fractal geometry:

Rotation angle equal to zero: $R_0 \ldots R_m = I$;

- Contraction condition of fractal geometry:

Invariance of translation vector:

$R_0 r_1 u + \ldots + R_{m-1} r_m u = u$, u is the unitary vector.

As an example, we can cite the Von Koch curve given by the general l-system grammar LS3:

-LS3: $\delta = 60$

ω : (-90)F(1)

p : $F(w) \rightarrow F(w*r_1)+F(w*r_2)$ --$F(w*r_2)+F(w*r_3)$

If we give the values $r_1 = r_2 = s_1 = s_2 = 1/3$, we obtain the classic Von Koch shapes of Fig. 4.

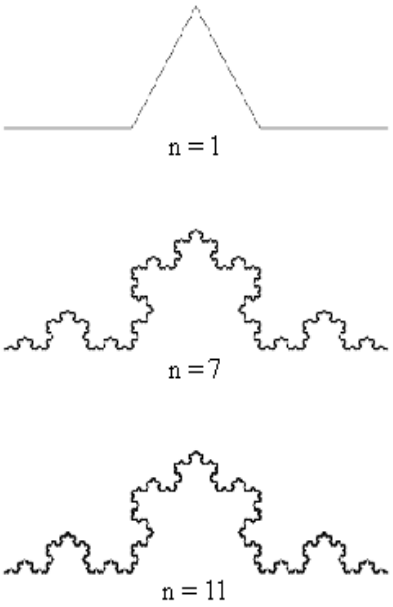

Fig. 4 Von Koch Curves correspondent to LS3 model for $r_1 = r_2 = s_1 = s_2 = 1/3$

C. Curved Trees

Starting from:

Theorem: the union of two IFS is an IFS [12]; we obtain:

Corollary: The union of two l-systems, with scaling factors verifying the conditions cited below, is an IFS.

With this corollary we can construct varied and rich interesting combined fractal forms [14].

LS4 is an example of an l-system grammar that combines two IFS described each one by a production rule to which we insert scaling factors inferior to 1.

-LS4:

$\delta = 60$

ω : F(1)

p1 : F(w) -> G1(0.4*w)G2(0.6*w)

p2: G1(w) -> F(0.5*w)[+F(0.5*w)]F(0.5*w)[-F(0.5*w)]

p3: G2(w) –> F(w*1/3)+F(w*1/3)--F(w*1/3)+F(w*1/3)

In this model, p1 is a production rule indicating that F(w) is the union of two symbols G1(0.4*w) and G2(0.6*w) (The contraction condition is verified: 0.4+0.6 = 1). G1 is an IFS described by parametric l-system grammar with scaling factors described in production p2. G1 attractor corresponds to a fractal tree shape. G2 is an IFS described by parametric l-system grammar with scaling factors described in production p3. G2 attractor corresponds to Von Koch fractal curve. LS4 grammar based model attractor corresponds to the curved fractal tree of Fig. 5.

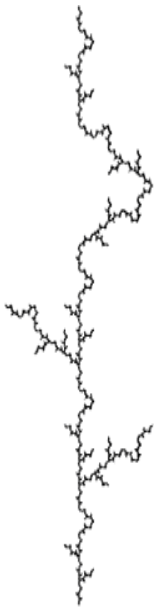

Fig. 5 Curved tree shape correspondent to LS4 model for n = 8

V. Conclusion

In this paper we presented a new method for fractal shapes' description with parametric l-systems and turtle algebra. This method permits us to generate filarial fractal shapes which structure can be smooth or rough. This can be of a great importance to represent some natural organisms (e.g. plants' organs). IFS as an iterative formalism capable to generate varied and rich forms, mainly quadric and parametric surfaces [13], can be integrated within l-systems grammars. One immediate consequence is that plants' organs (branches, leaves and flowers) can be represented iteratively within l-

systems grammars, and can be easily deformed due to the flexibility of parametric surfaces.

REFERENCES

[1] Alvy Ray smith, "Plants, fractals and formal languages", *Proceedings of the 11th Annual Conference on Graohics and Interactive Technics*, pages 1-10, 1984, ISSN0097-8930.
[2] E. Tosan, P. Maillot, J. Azencot, B. Décoret, "On process for the construction of scenes: an attempt of formalization", *Proceedings of IFIP CAPE'86*, North-Holland, P382-395 (1986).
[3] E. Tosan, "Wire frame fractal topology and IFS morphisms", *4th Conference Fractals in Engeneering*, Delft, The Netherlands, 14-16 June 1999, organization INRIA and TUDelft, pages 67-81.
[4] M. F. Barnsley, "Fractals everywhere", Morgan Kaufmann, 1993.
[5] P. Prusinkiewicz and A. Lindenmayer, "The algorithmic beauty of plants", Springer Verlag, 1989.
[6] P. Prusinkewicz and J. Hanan, "Visualization of botanical structures and processes using parametric l-systems",*Scientific Visualization and Graphic Simulation*, pages 183-201, 1990.
[7] P. Prusinkiewicz and M. Hammel, "Automata, languages and iterated function systems", *Lecture Notes for the SIGGRAPH '91 course: Fractal Modeling in 3D Computer Graphics and Imagery*, 1991.
[8] P. Prusinkiewicz, In M. Novak (ed.), "Self-similarity in plants: integrating mathematical and biological perspectives, thinking in patterns", Fractals and Related Phenomena in Nature, pages. 103-118, 2004.
[9] P. Prusinkiewicz, F. Samavati, C. Smith and R. Karwowski, "L-system description of subdivision curves", *International Journal of Shape Modeling* 9 (1), pages 41-59, 2003.
[10] R. Goldman, S. Shaefer and Tao Ju,, "Turtle geometry in computer graphics and computer aided design", *Computer Aided Design* 36(4): 1471-1482 (2004).
[11] T. Ju, S. Shaefer, R.Goldman, "Recursive turtle programs and iterated affine transformations", *Computer & Graphics* 28(6): 991-1004 (2004).
[12] T. Joëlle., « *Extension du modèle IFS pour une géométrie fractale constructive* », *Phd These*, Claude Bernard University of LYON1, 9 September 1996.
[13] Z. Chems Eddine, [44], «*Formes fractales à pôles basées sur une généralisation des IFS* », *Phd These*, Claude Bernard University of Lyon 1, June 1998.
[14] Z. Ikbal and A. Béchir, "A description of a family of iterated functions systems with parametric l-systems", *Proceedings of IEEE Sciences of Electronics, Technologies of Information and Telecommunications,* pages 1-7, Hammamet, 25-29 March 2007, Tunisia.

Ikbal Zammouri was born in Moknine, Tunisia in 13 of June 1971. He obtained his primary diploma from Fath School in Moknine in 1983 and his baccalaureat in Mathematics and Sciences from Said Abu Bakr secondary school of Moknine in June 1990. In July 1994, he obtained his computing engineering diploma from the Engineering National School of Monastir in Tunisia. In November 2002, he obtained his Sufficient Research diploma from la Facultad de Informatica de Murcia in Spain. Now he is a Phd thesis student in the National School of Computer Sciences of Manouba in Tunisia.

He was an ANALYTICAL ENGINEER in GTI Company in Tunis, from June 1995 to June 1997. He is an ASSITANT PROFESSOR in computer sciences department at the Faculty of Sciences at Monastir in Tunisia since October 2003. He published some papers in international conferences: A Realistic Plants' Modeling and Rendering with Lindenmayer Systems and Constructive Solid Geometry (Constantine, Algeria: Proccedings of ACIT'04, 2004), Modèle Itératif et Surfaces de Subdivision (Poitiers, France: Proceedings of GTMG'05, 2005)

Mr. Zammouri has recently published a paper titled: A Desciption of a Family of Iterated Function Systems with Parametric L-systems (Hammamet, Tunisia: Proceedings of IEEE Sciences of Electronic, Technologies of Information and Telecommunications'07, 2007).