

The Generation System of Textile Pattern Based on L-Systems

Suyi Liu^{1, a*}, Leduo Zhang^{2, b} and Haixia Zhang^{3, c}

^{1,3}College of Mechanical Engineering and automation

Wuhan Textile University, Wuhan ,430073, China

²SVA Communication Technology Co., Ltd, Xuhui Shanghai, 200233, China

^a liusuyi@sina.com

Keywords: Generation System; Textile; Pattern; L-Systems

Abstract. This paper is based on the basic principle of L-systems, through the visual means to express the basic function equation into graphical form, throughout these colorful graphics, are suitable as textile patterns. This paper describes the graphics composition principle and implementation method of L-systems, doing research on the fractal tree 1 generated by a single rule, fractal tree 2 generated by multi-rule and curve, the number of the graphics that generated by the basic principles is limited, therefore, this article starts from the basic functions and graphs, through the exchange, combination, nested of calculation symbols, and parameter change method, gets a lot of novel graphics. The content of this research not only enriches the algorithm and graphics of L-systems, but also provides a new thought to textile pattern design.

Introduction

The designs of textile pattern draft belong to the field of traditional arts and crafts [1]. Every textile pattern draft costs a lot of time and experience when it first comes out. The basic characteristic of traditional design methods is a manual operation. The features of traditional design methods are narrow-minded design thought, less design pattern, a slow creation cycle, and a difficulty in producing an initial pattern process. For the above reasons, pattern generation technology used in textile pattern draft generates a lot of beautiful patterns. When American biologist Aristid Lindenmayer did research on plant morphology evolution and structural, in 1968 he put forward a method to describe the grammar. Graftal, and later developed into an important branch of formal language, called L-Systems (LS) [2].

The emergence of L-Systems, attracted the interest of many computer scientist, botanist and mathematicians, they put forward various improved methods for L-systems, and developed some new method on generated plant graphics, such as IFS [3] (Iteration Function System) reference axis technology [4], fractal method [5], the environment particle system [6] and the dual scale automaton model [7], etc.

In 1984 and 1986, Smith A R and Prusin Kiewic P introduced L-systems into the computer graphics [8~9]. In 1985, Barnsley M F began to use iteration function system (IFS) to generate fractal graphics [10~11]. L-systems has obvious advantages in the description of plant topological structure, the IFS procedure is simple and easy to realize, the geometric meaning is clear.

Because the L-systems generates graphics mostly clear in lines, neat and beautiful, very suitable for various uses of textiles. Therefore, this paper has conducted the research on the method of using L-systems to construct textile pattern.

Composition Theories of L-system and Its Realization

L-system grammar is a unique iterative process. Its core concept is overwriting. That expresses initial form of generated object with alphabet and symbol string. Then based on a group of overwriting rule every character of initial form is replaced by new character form. On this process repeatedly replaced and overwrote, it last generates final pattern.

The basic composition theory of L-system is a theory [12] that uses a ternary form to express the object required to form. $L = \langle G, W, P \rangle$.

The G is a character set, which constituted of “F”, “[”, “]”, “+” and “-”. They explain graph command of character string. The W is initial character element, also called initial element, which decides initial state of character string, and $W \in G$. The P is generative rule set. It explains algorithm of output character of L-system. It is just the method of generating pattern according to character command of G .

Generated rule of L-system grammar has two types, which are single rule and majority rule.

Majority rule of L-system increase two or more letters in alphabet of single rule. And it also can increase generated rules.

In Fig. 1, Koch snowflake is generated by L-system grammar.

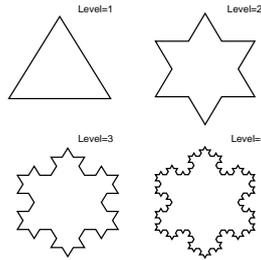


Figure 1. Koch snowflake

Generated rule of L-system grammar has two types, which are single rule and majority rule.

Single Rule

The key to this approach is the introduction of a new symbol, and remember each key point, when go to a key point, remember the current position, so that after walking along the direction, can return to the key points, go another direction.

For example, as shown in Fig.2, fractal tree is generated by single rule of L-system grammar. The triad $L = \langle G, W, P \rangle$, as a representation.

There: symbol set $G = \{F, +, -\}$; A symbol used to indicate the beginning of some set of data. $W = F ++F ++F ++F ++F ++F ++F ++F ++F ++F$; revolving angle $\delta = 45^\circ$; A generation rule is $P = \{F \rightarrow F --F +++F --F\}$.

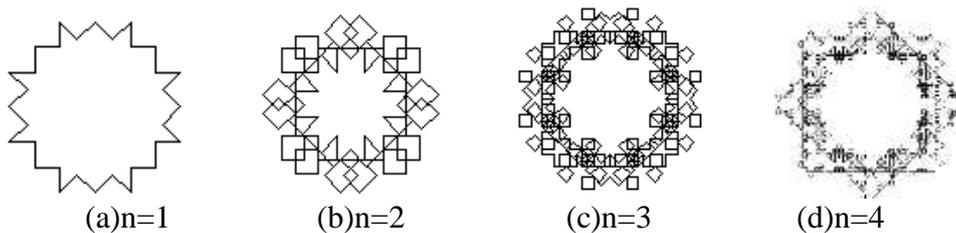


Figure 2. Fractal tree 1

Transform Methods of Patterns by Single Rule. “F”, “[”, “]”, “+” and “-” are characters in generative rule. These characters are changed by transform methods. And it is to generate various patterns.

Added sign “[” and “]” to the generative rule of fractal tree 1.

Changed generative rule $P = \{F \rightarrow F --F +++F --F\}$ into $P = \{F \rightarrow [F --F +++F --F]\}$ it gets the pattern shown as Fig.3.

Changed arithmetical sign of generative rule of fractal tree 1.

Changed “+” into “-”, and “-” into “+”, in generative rule $P = \{F \rightarrow F --F +++F --F\}$, it gets the pattern shown as Fig. 4.

Repeated generative rule of fractal tree 1

It repeats $F--F+++F--F$, which is in $P=\{F \rightarrow F--F+++F--F\}$. It gets the pattern shown as Fig.5.

Increased sign “[” and “]” for generative rule and took nested transformation

Added sign “[” and “]” to the generative rule $P=\{F \rightarrow F--F+++F--F\}$, it becomes $P=\{F \rightarrow [F--]F[+++F]-[-F]\}$. And it gets the pattern shown as Fig.6.

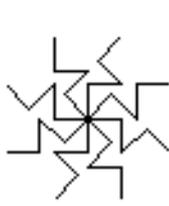


Figure 3

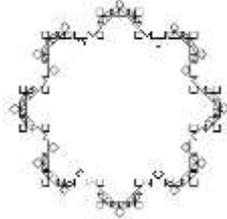


Figure 4

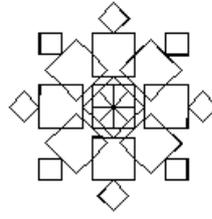


Figure 5



Figure 6

Transformed sign of initial element, or transformed generative rule and initial element in the meantime. It gets patterns shown as Fig.7.

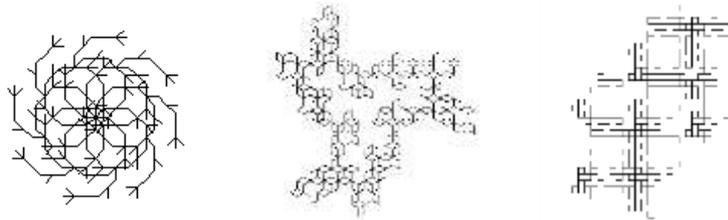


Figure 7. Transformed patterns

Majority Rule

Majority rule of L-system increase two or more letters in alphabet of single rule. And it also can increase generated rules.

This rule is mainly applied to two kinds of situations

The Character Work As A Replacement Character Not As Drawing Character.

Changed:

Axiom = "F"

$a(0) = "F": x(0) = "F-F+++F-F"$

Delta = 60

into:

Axiom = "X"

$a(0) = "X": x(0) = "-YF+XFX+FY-"$

$a(1) = "Y": x(1) = "+XF-YFY-FX+"$

Delta = 90

Among them, X and Y characters participate in the string replacement, but the program draw graphics not based on X or Y, it only according to F[34].

The fractal picture is elegancy and singularity, and has obvious geometrical meaning. just as dragon graph.

$$W = FX, P_1 = F \rightarrow (\text{empty}), P_2 = X \rightarrow -FX ++FY -, P_3 = Y \rightarrow +FX --FY +, \delta = 45^\circ,$$

numbers of iteration $n = 4$. It gets patterns shown as Fig.8.

To transform the characters of generation rules of “F”, “[”, “]”, “+”, “-”, to produce a variety of patterns.

Added sign “[” and “]” to the generative rule of the dragon graph, to add a generative rule to P_1 , and let $\delta = 30^\circ$, it gets the new generative rules: $P_1 = F \rightarrow XY$, $P_2 = X \rightarrow [-FX ++FY-]$, $P_3 = Y \rightarrow [+FX --FY+]$ and the pattern

shown as Fig.9.

Changed arithmetical sign of generative rule of the dragon graph, to add “[” and “]” in generative rule, it gets the new generative rules:

$P_1 = F \rightarrow (\text{empty}), P_2 = X \rightarrow [+FX-][-F]Y+, P_3 = Y \rightarrow -FX++[FY-] P = \{F \rightarrow F--F+++F--F\}$,
the pattern shown as Fig.10.



Figure 8. A dragon graph

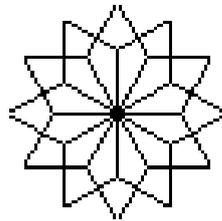


Figure 9

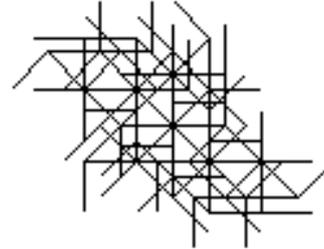


Figure 10

Repeated generative rule of the dragon graph after added sign “[” and “]” to the generative rule, it gets the new generative rules:

$P_1 = F \rightarrow XY, P_2 = X \rightarrow -F[X--F]Y++F[X--F]Y+, P_3 = Y \rightarrow -F[X++F]y--F[X++F]Y-$

and the pattern shown as Fig.11.

Increased sign “[” and “]” for generative rule and took nested transformation and $\delta = 20^\circ$. it gets the generative rules :

$P_1 = F \rightarrow XY, P_2 = X \rightarrow -F[X+XY+F]Y+, P_3 = Y \rightarrow +F[X-XY-F]Y-$ and the patterns shown as Fig.12.

Transformed sign of initial element, or transformed generative rule and initial element in the meantime. It gets patterns shown as Fig.13.

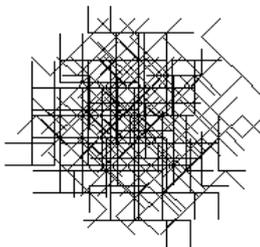


Figure 11



Figure 12



Figure13. Transformed patterns



The character work both as replaced characters and drawing character. In this case, not only to modify the array elements in the rule module program, but also to add graphics statements in the program. The details are as follows

let:

Case "F"

```
Picture1.PSet (4500 + xPos(K1_er), 6000 - yPos(K1_er)), RGB(0, 0, 0)
xPos(K1_er) = xPos(K1_er) + Cos(Delt(K1_er)) * 100
yPos(K1_er) = yPos(K1_er) - Sin(Delt(K1_er)) * 100
Picture1.Line -(4500 + xPos(K1_er), 6000 - yPos(K1_er)), RGB(0, 0, 0)
```

into:

Case "L"

```
Picture1.PSet (4500 + xPos(K1_er), 6000 - yPos(K1_er)), RGB(0, 0, 0)
xPos(K1_er) = xPos(K1_er) + Cos(Delt(K1_er)) * 100
yPos(K1_er) = yPos(K1_er) - Sin(Delt(K1_er)) * 100
Picture1.Line -(4500 + yPos(K1_er), 6000 - xPos(K1_er)), RGB(0, 0, 0)
```

Case "R"

Picture1.PSet (4500 + xPos(K1_er), 6000 - yPos(K1_er)), RGB(0, 0, 0)

xPos(K1_er) = xPos(K1_er) + Cos(Delt(K1_er)) * 100

yPos(K1_er) = yPos(K1_er) - Sin(Delt(K1_er)) * 100

Picture1.Line -(4500 + yPos(K1_er), 6000 - xPos(K1_er)), RGB(0, 0, 0)

For example, in the computer program of fractal tree 2: $W = L$,

$P_1 = L \rightarrow L[-LR - LR - LR]L[+LR + LR + LR]L[LR]$,

$P_2 = R \rightarrow RL[+R][++R][+++R][-R][-R][-R]R, \delta = 22^\circ, n = 4$.

it gets the pattern shown as Fig.14.

To transform the characters of generation rules of "F", "[", "]", "+", "-", to produce a variety of patterns.

Added sign "[" and "]" to the generative rule of fractal tree 2, it gets the new generative rules:

$P_1 = L \rightarrow [L[-LR - LR - LR]L[+LR + LR + LR]L[LR]]$,

$P_2 = R \rightarrow RL[+R][++R][+++R][-R][-R][-R]R$ and the pattern shown as Fig.15.

Changed arithmetical sign of generative rule of fractal tree 2, it gets the new generative rules:

$P_1 = L \rightarrow L[-LR + LR - LR]L[+LR - LR + LR]L[LR]$,

$P_2 = R \rightarrow RL[+R][++R][---R][-R][-R][+++R]R$, and the pattern shown as Fig.16.

Repeated generative rule of fractal tree 2, it gets the new generative rules:

$P_1 = L \rightarrow L[-LR + LR - LR]L[+LR - LR + LR]L[LR]$,

$P_2 = R \rightarrow RL[+R][++R][---R][-R][-R][+++R]RRRL[+R][++R][---R][-R][-R][+++R]R$, and the pattern shown as Fig.17.



Figure 14. Fractal tree 2

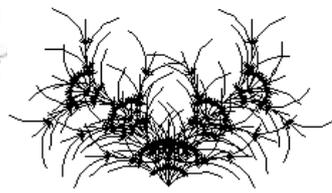


Figure 15



Figure 16



Figure 17

Increased sign "[" and "]" for generative rule and took nested transformation and $\delta = 20^\circ$, it gets the new generative rules:

$P_1 = L \rightarrow [L[-LR + LR - LR]L[+LR - LR + LR]L[LL[-LR + LR - LR]L[+LR - LR + LR]L[LR]R]$,

$P_2 = R \rightarrow RL[+R][++R][---R][-R][-R][+++R]R$, and the pattern shown as Fig.18.

Transformed sign of initial element, or transformed generative rule and initial element in the meantime, it gets patterns shown as Fig.19.

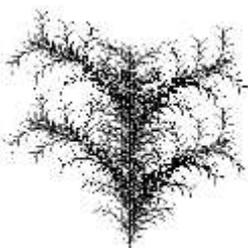


Figure 18

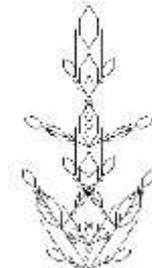
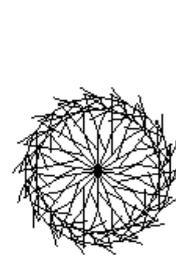


Figure 19. Transformed patterns

Conclusion

L-system is an algorithm to construct the graph that modelled grammar generation methods in linguistics, it though generating a series of string to construct a graph or curve. But due to the number of the graphics that generated by the basic equations of the L-system is limited, and patterns are too monotonous, therefore, this paper focuses on the extension problem of basic functions, through the exchange, combination, nested of arithmetic symbols and parameter change method, greatly expands the types of functions, derived more patterns, at the same time, these graphics generated by derivative functions are more gorgeous and exquisite. In this paper, Fig. 2 and Fig. 4 pattern shown with a smooth, well proportioned, complete, the contour fine features, their style is suitable for carpet, bed sheet, bedspread, tablecloths and so on. Fig. 7 shows the pattern of elegant and beautiful, can be used as a lady's skirt, men's shirts, as well as decorative textiles edges and corners etc. Fig. 11 shows the pattern has a well-balanced network structure, with regularity and rhythm, Fig. 19 has a compact and strong sense of hierarchy characteristics, suitable for the hat and scarf.

Therefore, we believe that the various graphics that generates by L-systems graphical were mainly have geometrical pattern features, strong flavor of the times, accord with the textile consumer aesthetic habits, completely suitable for textile decoration art performance.

Acknowledgements

This project was funded by the Hubei Provincial Digital Textile Equipment Key Laboratory Project (DTL2018029).

References

- [1] Janguo Sun, *The Pattern Design of Textile* (Textile & Apparel Press 2013). (In Chinese)
- [2] Lindenmayer A, *Mathematical Models for Cellular Interaction in Development*, *Journal of Theoretical Biology*, 1968, 18, pp.280~315.
- [3] Prusinkiewicz P. and Lindenmayer A, *The Algorithmic Beauty of Plants*. (New York: Springer-Verlag 1990).
- [4] Dekmo S, Hodges L and Naylor B, *Construction of fractal objects with it crated function systems*, *Computer Graphics*, 1985, 19(3) , pp.271-278.
- [5] Blaise F, Barczy J F, and Jaeger Meta1, *Simulation of the Growth of Plants Modeling of Metamorphosis and Spatial Interactions in the Architecture and Development of Plants*. (Cyberworlds, Tokyo, Springer Verlag 1998), pp.81-109.
- [6] Oppenheimer P., *ReM Time Design and Animation of Fractal Plants and Trees*, *Computer Graphics*, 1986, 22(4), pp.55~64.
- [7] Reeves W., *Approximate and Probabilistic Algorithms for Shading and Fendering Structured Particle Systems*, *Computer Graphics*, 1985, 19(3), pp.313—322.
- [8] Smith A R, *Plants Fractals and Formal Languages*, *Computer Graphics*, 1984, 18(3), pp.1-10.
- [9] Lindenmayer A, *Mathematical Models for Cellular Interactions in Development*, Part-I and *Journal of Theoretical Biology*, 1986, 18, pp.280-300.
- [10] Barnsley M F and Demko S, *Iterated Function Systems and the Global Construction of Fractal*, *The Proceedings of the Royal Society*, 1985, A399, pp.243-275.
- [11] Barnsley M F, *Fractals Everywhere*. (New York: Academic Press 1988).
- [12] Bowen Sui, *Fractal Algorithm and Program Design*. Science Press.2014. China.