Visualization of a two-dimensional tree modeling using fractal based on L-system

E W Hidayat1, I K G D Putra2, I A D Giriantari2 and M Sudarma3

1Siliwangi University, Siliwangi Street, Tasikmalaya 46115, Indonesia
2Udayana University, Panglima Besar Sudirman Street, Denpasar 80234, Indonesia

Email: ekawahyu@unsil.ac.id

Abstract. The purpose of this paper is to present a visualization of two-dimensional trees using fractals and L-Systems. Grammar in L-System consisting of letters, axioms, and production rules can be used to construct trees in various forms according to the production rules used. With the formation of grammar from the tree pattern in a fractional manner, the grammar can be reused to construct other tree patterns at a higher level. The iterations made on certain production rules visually affect the tree model. Experiments were carried out on the L-System generator to see the results of tree visualization by testing various grammars. Testing was done by trial and error in production rules to get a realistic visualization of a two-dimensional tree model. From the experimental results produced a variety of grammar, where modifications to the production rules produced different tree models. Visually, from the iteration results, the higher the iteration value, the two-dimensional tree was not formed properly. When the iteration value or \( n > 5 \) the two-dimensional tree model results from an iteration it lost detail from the branch.

1. Introduction

The form of self-similarity naturally exists in nature, according to Mandelbrot called as Fractal. The fractal object has infinite details and has a similar structure of self at different magnification levels. This property means that the more fractal objects enlarged the more detailed objects will be obtained. Details of the fractal object are not visible directly but will appear gradually in the event of magnification. There are three levels of self-similarity in fractals [1], namely Exact self-similarity, Approximate self-similarity and Statistical self-similarity. Approximate self-similarity is a natural form that matches the paradigm of self-similarity in nature, such as trees, clamshells, clouds, lightning, river tissue, and blood vessels. Fractals have been applied to various aspects, for example, applied to various forms of mathematical computations that are to produce tree branch patterns [2], leaf pattern [3], and fibrous roots pattern [4]. Tree Modeling is the part of the fractal with Approximate self-similarity. Trees have very complex structures but are well defined. One of the major factors that organize tree structure is the concept of self-similarity as characterized by the nature of fractals. Fractal geometry can be generated using a number of methods namely Lindenmayer System (L-System) and Iterated Function System (IFS).

Imitation of natural forms as a basis seed has been done to imitate the fractal geometry-based butterfly pattern. Objects of fibrous roots can also be produced using L-System and combined with existing batik patterns [4]. Other studies offer methods and algorithms for constructing butterfly patterns by adopting two-dimensional IFS models [5]. While other studies were carried out to construct a line
graph on two-dimensional fractals with the L-System approach. This study shows that the fractal dimensions of the graphs obtained in all the cases analyzed are the same as the original graph, both for the original graph and for the line graph by identifying the node class that reflects the graph symmetry. Fractals are applied to the construction of tree branch patterns [2]. Other researchers have developed L-System to become an Adaptive L-System so that it can be used to produce geometric branching structures in various forms and to illustrate the shape of leaves and tree branches with finer nodes [3]. Other studies propose an approach in studying the modeling of Zinnia plant growth using the L-System method using Mathematica software [6]. The study conducted by Lim (2017) compared L-System applications for plant modeling, music rendering, generating musical scores [7]. In the study several applications such as TreeSketch, L-Studio, and L-Py.

The research roadmap in this study is as a prefix in visualizing tree models from two-dimensional shapes to three-dimensional shapes which can later be used for three-dimensional generated trees for application development in the fields of Games, Augmented Reality and Virtual Reality. Research challenge in this research is how to visualize trees using L-System which is limited by Angle, Axiom, and Production Rules. This paper will explore the use of fractal geometry as a basis for simulating tree shapes using the L-System approach and visualize computational two-dimensional tree construction from the L-System grammar made up of four iterations.

2. Fractal based on l-system
Fractals are roughly fragmented geometric shapes, can be divided into several parts, and each part is a clone of size equal to or greater than the original shape of the whole. Fractals are defined as branches of geometry that show that a set of repeated geometric patterns with scaled scales can be used to describe geometric patterns contained in nature or to describe the complexities of mathematical models [1, 5]. Fractals show that the forms in nature are not dimensionless integers but dimensionless fractions, such as the branches of a tree will be seen as miniature of the tree as a whole, not exactly similar but similar [8]. A fractal can be generated by repeating a pattern, usually in a recursive or iterative process. Mandelbrot uses the word fractal to describe objects that do not have clear dimensions but have self-similarity.

The use of fractals in modeling a two-dimensional tree using L-System means that the tree has a self-similarity characteristic, in which each part of the tree is an imitation of size equal to or greater than the original shape as a whole which is repeated many times on a minimized scale. The repetition can be formed through grammar in the L-System. The following figure shows that in each branch of a tree in the constellation using L-System it is a repetition of the seed shape.

![Figure 1. Fractal tree using L-System with $\delta = 25^\circ$, Axiom P: F, $\alpha$: F [+F] [-F]. The tree branch image in the circle line shows the result of the loop from the first iteration. The branches of the tree are self-similar to the seeds. So it can be said that the tree is recursive and has fractal properties.](image)

3. L-system
L-System or Lindenmayer System is a parallel rewrite system and is included in a formal grammatical type. This system was introduced by Aristid Lindenmayer (1925-1989) a Hungarian theoretical biologist and botanist from Utrecht University in 1968 to model the pattern and process of the growth of various types of algae. L-System has been successfully used to produce models of many natural forms, including higher plants.

L-System is a context-free grammar in which each production rule applies only to one symbol on a set. Other symbols are not affected by the production rule. The basic idea of L-System is to form an object starting from the initial structure by swapping or replacing some parts of the grammar notation
(rewriting rules) on a rule by sequentially looping mechanisms that refer to a self-similarity. The loop is performed according to the number of desired iterations [9].

The L-System was initially in form to provide a formal description of the development of simple multicellular organisms and illustrates the relationship between plant cells. Then, the system is expanded to describe higher plant species and complex branch structures. Recursive behavior in L-System leads to self-similarity so that natural forms such as fractals can be easily drawn using L-System. The forms of plants and other forms of organisms are quite easy to define because by increasing the rate of repetition, this form will grow and become more complicated [9].

3.1. L-System grammar
The grammar of L-system is very similar to the grammar in the Chomsky hierarchy. The L-System grammar is defined according to Prusinkiewicz (1990) and Mishra (2007) as a pair of 3 tuples \( G = (V, \omega, P) \) where \( V \) (alphabet) or letter is a set up to \( V \) and formal symbols, \( \omega \) (start, axiom, initiator) or axiom is a string \( \omega \) of the \( V \) symbol that defines the initial state of the system [9, 10]. The string set of \( V \) is denoted as \( V^{*} \), and \( P \) is a set of production rules a symbol mapping \( a \in V \) to string \( w \in V^{*} \) is written as \( p: a \rightarrow w \). A variable can be replaced with a combination of constants and other variables. Production consists of two strings namely predecessor and successor. For all symbols \( a \) if \( a \in V \) has no production rules, it is assumed that the symbol is mapped to itself so that \( a \) becomes the constant or terminal of L-System.

3.2. Rewriting production rule
The L-System fractal consists of a symbolic alphabet that can be used to create strings, a collection of production rules that extend each symbol to some larger symbol strings, the initial "axiom" string to begin construction, and a mechanism for translating the string generated into geometric structures. The principle of rewriting rules in L-System grammar is exemplified by L-System notation used by Lindenmayer to model algae growth.

For example, given two symbols A and B. Each symbol declared as a production rule. A produced AB and B produced A, then every iteration of each symbol is rewritten according to its production rules. The product of L-System is defined as a sequence \( \{g_n\} \) with \( n = 1, 2, 3, 4, \ldots, k \) where the first iteration \( g_0 \) is axiom \( w \). The process can be seen in table 1 showing the L-System writing process for Algae growth with 7 iteration from \( g_0 \) to \( g_6 \).

| Grammar       | Result    |
|---------------|-----------|
| Variabel : A B| g0: A     |
| Axiom : A     | g1: AB    |
| Production Rule : A → AB| g2: ABA |
| B → A         | g3: ABAAB |
|               | g4: ABAABABA |
|               | g5: ABAABABAABAAB |
|               | g6: ABAABABAABAABAAB |

3.3. Visualization of L-system
In L-System there are symbols that can be interpreted graphically. If it is assumed to be a unit of length \( h \) and angular rotation \( \delta \), then the commands of the symbols in L-System used for visualization of trees in this study are as follows:

- F : drawing forward one unit along a \( h \). (draw line forward).
- + : spin clockwise with angle \( \delta \) (rotate right).
- - : spin counter clockwise by angle \( \delta \) (rotate left).
• [: save current position and move according to the next command (begin new branch).
• ]: back to the original position stored by the symbol "[" (end branch).

3.4. L-system algorithm
Solving the problem of how to interpret the grammar above can be described in the form of systematically arranged steps and then to interpret grammar into a graphic can be aided by the general algorithm of fractal object interpretation on L-System with the following steps:
• Enter value of generation (k), branch inclination angle (δ) and line segment length (h).
• Determine the starting angle (a₀) and enter the a₀ value to get the starting point (F₀) ; then enter F₀ on the production (P) so as to produce P₀.
• Perform the iteration process i for i = 1, 2, 3, ..., k so get the next angle value (aᵢ) and forwarded to get the next point value (Fᵢ) ; then enter Fᵢ on production rule Pᵢ₋₁ to get Pᵢ.
• Plotline segments generated from axioms and production rules.

4. Results and discussion
The turtle graph grammar reconstruction into a tree graph was implemented using a generator built using JavaScript and visualization of the reconstruction results viewed using the internet browser. The following are four experimental results from several production rules used can be seen in the figure 2 to figure 6.

Figure 2. Fractal tree using L-system with δ = 25⁰, Axiom P: F, ω: FF [+F+FF] [-F-F] [+FF+F].

Figure 3. Fractal tree using L-system with δ = 25⁰, Axiom P: F, ω: FF [-F+F+F]+[+F-F-F].
Figure 4. Fractal tree using L-system with $\delta = 25^0$, Axiom P: F, $\omega$: FF+[F-F+F]-[F+F-F].

Figure 5. Fractal tree using L-system with $\delta = 25^0$, Axiom P: F, $\omega$: FF [+F][-FF] [-F+F].

Figure 6. Fractal tree using L-system with $\delta = 25^0$, Axiom P: F, $\omega$: FF+FF [F-F] [-F+F].

Figure 2 through figure 6 show the experimental results for generated model trees. It can be seen that production rules in the construction will produce different models. Visually, figure 2 to figure 5 approaches the visualization of trees, but figure 2 and figure 3 are visually more realistic. While figure 6 successfully constructed grammar but fails to visualize the tree model.

5. Conclusion and future research
In this paper, a variety of L-System grammar had been tested to reconstruct fractal trees with the same branch slope for all experiments, namely $\delta = 25^0$. The axiom used was F. The deterministic grammar was used as an approach. The experimental obtained several grammars forming two-dimensional trees visually to see which iteration that were good for modeling trees. Some trials and errors must be done in the production rules. From the iteration results, it can be seen that the higher the iteration value, the two-dimensional tree was not well-formed visually. When the iteration value or $n > 5$ iteration result trees lose detail from the branch. The results of the trial show that different production rules will produce different fractal trees. For further research, modeling trees in three dimensions using L-System must be.
tested first using the L-System grammar for a two-dimensional tree can form a better visual tree. Because L-System grammar consists of two types, namely Deterministic and Non-Deterministic, it is necessary to conduct further studies to see the results of the reconstruction in a non-deterministic manner.

References
[1] Mandelbrot B B 1982 The Fractal Geometry of Nature (New York: W H Freeman and Company)
[2] Warchalowski W and Krawczyk M J 2017 Line Graphs for Fractals (Commun Nonlinear Sci Numer Simulat: Elsevier) pp 506–512
[3] El-Latif Y M A 2010 A New Conception in Constructive Branching Structures and Leaves Using L-System JCSE 4 240–52
[4] Kusuma P D 2017 Fibrous Root Model in Batik Pattern Generation JATIT 95(14) 3260–9
[5] Barnsley M F 1988 Fractals Everywhere (Sandiego: Academic Press)
[6] Suhartono 2013 Pemodelan Pertumbuhan Tanaman Zinnia Menggunakan Lindenmayer System Dengan Mathematica Cauchy: Jurnal Matematika Murni dan Aplikasi 3(1) 33–7
[7] Lim C K Tan K L and Suppramaniam V 2017 Comparison of L-System Applications Toward Plant Modelling Music Rendering and Score Generation Using Visual Language Programming Int. Conf. Appl. Sci. Tech. 020086:1–6
[8] Buchanan T 2017 Power of Pattern Fractal (Huntington: Teacher Created Materials)
[9] Prusinkiewicz P and Lindenmayer A 1990 The Algorithmic Beauty of Plants (New York: Springer-Verlag)
[10] Mishra J and Mishra S 2007 L-System Fractal (Netherland: Elsevier)