Simulating Branching Geometries in Random Trees

K.D.S. Jinasena and D.U.J. Sonnadara
Department of Physics, University of Colombo, Sri Lanka

Abstract

Bifurcation branching geometry, which is one of the fundamental branching geometries, was used to simulate the development of random trees using a recursive algorithm. Probability concepts were introduced into the model to generate and control the growth of sub-branches, to replace the use of a fixed set of plant specific parameters. The structural changes of random trees have been studied with the addition of a stochastic nature to the parameters used in the model.

1. Background

Simulation of living organisms has been a difficult task due to the complexity of the random forms and patterns found within them. However, it has been shown that most natural phenomena have the self similar property integrated within their formation and therefore it is possible to model such structures using recursive algorithms.

Most methods for modeling trees are based on the assumption that trees have a repetitive, recursive structure. This assumption is consistent with the established notion of architectural tree models identified by Halle, Oldeman, and Tomlinson [1], where the form of a tree is largely determined by its branching pattern. They have categorized tropical trees into 23 different branching types and recorded the characteristics of those tree architectures.

Several past studies [2-4] have attempted to model botanical trees in geometric forms. The representations for the most part are focused on specific plant types and have not been flexible enough to use as a general model for simulating trees. Moreover, these studies have not investigated the stochastic nature of trees, which is one of the most important parameters of natural objects.

The main objective of this work is to develop a simulation model which is independent of plant specific parameters. This would allow us to study the impact of internal and external factors on the development of trees.

2. Methodology

Binary tree data structures were used in order to implement the bifurcation branching geometry computationally. A typical binary tree consists of nodes and edges in which each node has at most two child nodes. An edge refers to the link from the parent to the child.

In this work, Honda's model [5] was used to simulate trees with bifurcation branching geometry. Three-dimensional positions of the end-points of the branches were calculated using geometrical assumptions which included two parameters; the branching angle and the relative ratio of the branch lengths. There are two models developed by Honda, known as Honda's H Model and Honda's I-Model. Honda's I-model was chosen to generate the tree structure in this work mainly due to its ability to model non planar branching structures.

Unlike a complete binary tree, every branching point of a natural tree does not divide into two sub branches. In real plants, the development of branches is a rather complex process and in most cases it is dependent on a number of internal and external factors such as the amount of nutrition and water transferred to the branches, the availability of sunlight, space for the branch to grow, wind, etc. In reality, the decisive factor for developing branch segments is a combination of all the factors. It is hard to implement a model which incorporates all the internal and external factors affecting the branch development. Therefore the statistical probability of branches occurring was considered instead of considering the internal and external factors directly.

In the Monte-Carlo simulation, the following initial step was taken to introduce statistical variations. Two independent probabilities (P_l, P_r) were assigned for the generation of the left and right sub branches. It was assumed that the growth of a sub branch segment will occur only if a random number $R \in [0,1]$ is less than P_l or P_r.

$$R[0,1] \begin{cases} > P_{(l,r)} \in [0,1] & \text{dormant node} \\ < P_{(l,r)} \in [0,1] & \text{active node} \end{cases}$$

3. Results

A prototype Tree Simulator program was developed using MatLab® which allows the user to change the modeling parameters interactively. The tree structure was generated by adding random factors to the branching angles, the branch lengths and the thickness of the branches. The user was given the freedom to change a total of eight parameters that can influence the development of a random tree.

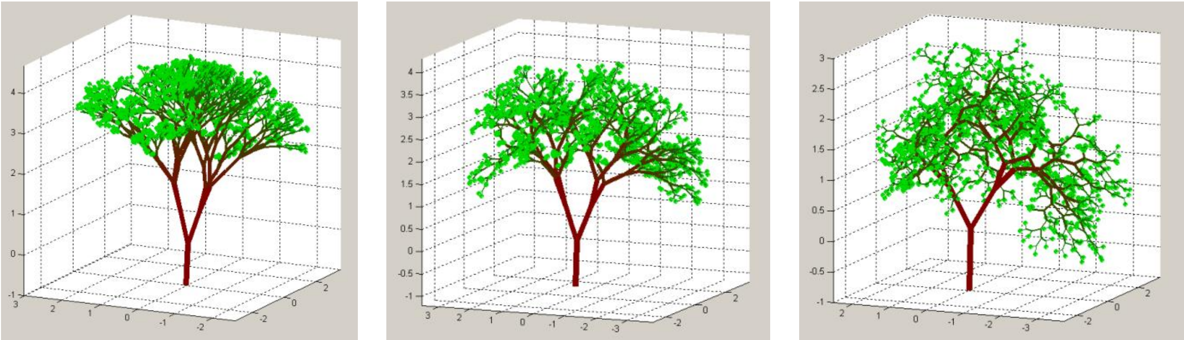
The outcome of the developed model was tested by assigning values to the branch angles, the inclined angles and the scaling factor for branch segments (see Table 1), which were extracted from a study of branching patterns of real trees in Terminalia [6].

Table 3: Parameters and default values used in the simulation of random trees

Parameter	Parameter Description	Default value	Max random variation
h	Level of branching	10	-
θ_1	Branching angle of left branch segment	16^0	+ 0.5
θ_2	Branching angle of right branch segment	17^0	+ 0.5
Δ	Angle of inclination in branching plane	20^0	+ 0.5
S_l	Scaling factor for branch length	0.80	± 0.2
S_t	Scaling factor for branch thickness	0.86	- 0.2

In natural trees, the internal and external effects influence branch development. Work is still underway to find mathematical representation to mimic such effects and to study the structural changes. However visual inspection of the generated tree patterns reveal that tree structures similar to natural trees could be generated with the above parameters, with minimum usage of computer graphic effects.

It was noticed that simulated tree structures deviate from their natural appearance when the branching angles (θ_1, θ_2) are increased beyond a certain limit. By considering the visual similarities between simulated tree structures and natural trees, the optimum range for branching angles which output the best natural appearance was identified to stretch out between 15^0- 25^0 (see Figure 1).



$\theta_1 = 16^0$ $\theta_2 = 17^0$ | $\theta_1 = 25^0$ $\theta_2 = 25^0$ | $\theta_1 = 40^0$ $\theta_2 = 40^0$

Figure 1: Simulated trees for different branching angles $\theta_{(1,2)}$. Default values given in table 1 were used as other model parameters.

4. Conclusions

In this research work, a Monte-Carlo simulation model which uses probability concepts to determine the growth of branches of random trees was developed. Although the present study is limited to using a set of selected probabilities to determine the occurrence of branches, in the case of natural trees there are a large number of internal and external factors that can influence the development of a tree. We are in the process of extending the model to study the effect due to internal and external factors influencing the growth of a tree, and to compare the results with the data obtained from natural trees.

References

[1] Halle F., Oldeman R.A.A. & Tomlinson P.B. (1978). *Tropical trees and forests: An architectural analysis.* Springer-Verlag, Berlin-Heidelberg-New York

[2] Fisher J.B. & Honda H. (1977) Computer Simulation of Branching Pattern and Geometry in Terminalia (Combretaceae), a Tropical Tree. *Botanical Gazette* 138 (4): 377-384

[3] Honda H. & Fisher J.B. (1978). Tree Branch Angle: Maximizing Effective Leaf Area. *Science, New Series* 199 (4331): 888-890

[4] Honda H., Hatta H. & Fisher J.B. (1997) Branch Geometry in *Cornus Kousa* (Cornaceae): Computer Simulations. *American Journal of Botany* 84 (6):745-755

[5] Honda H. (1971). Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology* 31(2): 331-338

[6] Fisher J.B. & Honda H. (1979). Branch geometry and effective leaf area: A study of Terminalia-Branching pattern. 2. Survey of real trees. *American journal of Botany* 66 (6): 645-655.

Object Tracking using Optical Flow

G.D. Illeperuma and D.U.J. Sonnadara
Department of Physics, University of Colombo, Sri Lanka

Abstract

Tracking objects in real-time has a variety of applications in many fields. Optical flow based tracking is one such tracking mechanism which can track moving objects even under complex backgrounds and different light conditions. The research presented in this paper discusses the feasibility of using optical flow to track moving objects captured in a camera view, to extract basic information related to the objects. The motion of a simple pendulum, a ball falling through a viscous medium and fish swimming in a fish tank were used to demonstrate the validity of the method.

1. Background

In computer vision, tracking objects is an important area. Although there are many algorithms and methods to track, to count or to identify objects, the most commonly used methods use background subtraction and color based tracking. In this research, optical flow is used to track objects. Optical flow is the amount of image movement within a given time period. This method of tracking can be easily converted into parallel processing and it is much faster than conventional tracking methods.