Recognizing Plants Using Stochastic L-Systems

Ashok K. Samal
University of Nebraska-Lincoln, asamal1@unl.edu

Brian Peterson
University of Nebraska-Lincoln

David J. Holliday
University of Nebraska-Lincoln

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

Part of the Computer Sciences Commons

http://digitalcommons.unl.edu/cseconfwork/37

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.
ABSTRACT

Recognizing naturally occurring objects has been a difficult task in computer vision. One of the keys to recognizing objects is the development of a suitable model. One type of model, the fractal, has been used successfully to model complex natural objects. A class of fractals, the L-system, has not only been used to model natural plants, but has also aided in their recognition [1]. This research extends the work in plant recognition using L-systems in two ways. Stochastic L-systems are used to model and generate more realistic plants. Furthermore, to handle the complexity of recognition, a learning system is used that automatically generates a decision tree for classification. Results indicate that the approach used here has great potential as a method for recognition of natural objects.

1. INTRODUCTION

Natural objects have much complexity and variation, making their recognition with a computer a difficult task. One of the keys to recognizing natural objects is to define a suitable representation, or model, for that class of objects. A model must account for the complexity and variation inherent in the objects. One type of model that is particularly well suited for modeling natural shapes is the fractal. Fractals have many interesting properties that make them attractive to model natural objects. Their use in computer graphics to generate realistic natural scenes shows their usefulness [2, 3, 4].

Fractals have also been used in computer vision, although not as widely, to model and recognize natural scenes [5, 6, 7]. However, the use of fractal models for recognition of single objects, such as plants and flowers, has been relatively unexplored.

One type of fractal, the L-system, has been used to model trees and tree-like structures [4]. It can accurately capture the prominent features of trees and plants. This model has also been used successfully to model plant growth. Our previous work has shown that the L-system can not only be used to model plants and tree-like structures, but can also aid in their recognition [1, 8]. Building on that work, a more robust and accurate plant model, the stochastic L-system, is used to model tree-like structures. These structures are then classified using a decision tree that is automatically computed with the aid of a learning system. This research shows that robust and accurate recognition of complex tree-like structures, such as those generated by stochastic L-systems, is feasible.

This article is organized as follows. First, L-systems are described, along with a more sophisticated version called the stochastic L-system. Next, the features used to describe the L-system objects are discussed. Then, the use of a learning system to generate the decision tree for classifying the objects is presented. Finally, the implementation details and the results are discussed.
2. L-SYSTEMS

The L-system is a particular type of fractal that can be used to model trees and plants. It was first conceived as a mathematical model to describe the growth and interaction of cells within plant structures [9]. Only later, when graphical applications of L-systems were proposed, could they be used to model higher plant structures, such as branches, leaves, and the plants themselves [10].

The way the objects are generated makes the L-system an ideal candidate for modeling trees and plants [2]. There are many types of L-systems, e.g. context-free, context-sensitive, and stochastic. One of the simplest is the context-free L-system. Each of these types is capable of generating trees and tree-like objects. Certain types, for instance, a stochastic L-system, are able to represent a tree or plant more accurately than other types.

The classes of L-systems form a hierarchy, much like the hierarchy for formal grammars introduced by Chomsky [11]. The Chomsky hierarchy contains, among other types, context-free and context-sensitive grammars. The classes of L-systems are very similar to the corresponding classes of Chomsky's hierarchy. There are two important ways, however, in which L-systems differ from Chomsky grammars. First, the distinction between terminals and nonterminals in an L-system is not as great as in formal grammars. Symbols that are used as nonterminals in the application of productions may be treated as terminals when the image corresponding to the L-system is actually generated. Second, all nonterminal symbols in an L-system are replaced in parallel when a production is applied. That is, no choice can be made about which nonterminal in a string may be replaced. Finally, the strings generated using the L-system is graphically interpreted using an object called a turtle.

A context-free L-system is formally defined as a 4-tuple \( G = < S, P, d, \delta > \), where \( S \) is the starting symbol (often called the axiom), \( P \) is the set of productions, \( d \) is the unit length, and \( \delta \) is the unit angle. More complicated classes of L-systems have additional components, but these four suffice to describe a context-free L-system.

As in string grammars, objects in L-systems are derived using the rewrite rules. The generation of an object from an L-system can be thought of as treating the symbols in the derived string as commands and then interpreting them to move a turtle around a coordinate system, possibly drawing while moving. The turtle, which can draw straight lines and adjust its orientation, has a state associated with it. This state is a 3-tuple \((x, y, \alpha)\), where \((x, y)\) is the current location in the coordinate system and \(\alpha\) is the turtle's current orientation.

The symbols that may appear in an L-system production describe how a turtle's state is manipulated. Two symbols, "F" and "f", specify that the turtle move forward while drawing and move forward without drawing, respectively. Two of the commands alter the turtle's orientation. The command "+" turns the turtle \(\delta\) degrees to the left, while "-" turns the turtle \(\delta\) degrees to the right. Finally, there are two branching commands. The command "[" starts a branch, pushing the turtle's state onto a stack and "]" ends a branch, popping the state of the turtle from the stack.

A simple L-system grammar and the graphical representation of the objects derived from it at several iterations are shown in Figure 1.

![Image of L-system with 2, 3, and 4 iterations]

Figure 1: An L-system grammar and some objects generated by it.
Stochastic L-systems: Context-free L-systems are adequate for modeling a large class of plants. However, due to their deterministic nature, they cannot represent the minor variations that is essential to model the plants accurately. Stochastic L-systems overcome this drawback by assigning probabilities to various patterns of growth.

Stochastic L-systems are almost identical to the L-systems previously discussed. They differ in that multiple productions can exist for a single letter; the production applied is determined by a predefined probability. For example, one production may be applied to a letter 50% of the time, another production 30%, and still another the rest. Figure 2 shows a stochastic L-system grammar and three instances of objects generated using it, all at five iterations.

Figure 2: Three objects generated using a stochastic L-system.

Due to their probabilistic nature, stochastic L-systems have an advantage over regular L-systems when modeling natural trees. Since different productions may be used at different times, it is possible to generate many different variations of the plant at the same iteration. This allows for the creation of plant classes, all sharing similar overall characteristics, but differing in minor details. For instance, the plants in Figure 2 all appear similar, but are not identical.

3. OBJECT RECOGNITION USING FRACTAL MODELS

Natural objects of a given class come in a variety of shapes and sizes. The objects can be, in effect, categorized into different types. A model must to be developed for each class of objects. Here, a different L-system grammar is used for each class of plant or tree. The next step is to select the features to be used for recognition. These features should be selected with an eye towards ease of recognition of objects and robustness and ease of computation.

Once specific features are decided upon, the recognition process is divided into two stages. The first stage is known as the training stage. In this step useful features for recognition are determined and then their values are computed for each model. In the testing stage, the identity of an unknown object is determined. The underlying recognition strategy is based on matching features of the unknown object with those of the known object classes. Therefore, the values for the features selected in the training stage are also computed for the unknown object. The feature values are then compared to those of the known models. The result of the matching process determines the identity of the unknown object.

There are a variety of mechanisms available to classify objects by matching features, including template matching, clustering, and minimum distance classifiers. The approach taken here is to use a decision tree classifier. By traversing the tree from its root, the possible classes an unknown object can belong to is narrowed down. This is done until the class the object belongs to is uniquely determined.

Features: Many general properties of objects are used for classification. These include the principal axis, the moments of inertia, and symmetry. In addition, several new properties of plants are defined and used. The lengths of selected branches as well as the separation between them are some examples of such properties. Figure 3(a) illustrates the definitions used for the branch features in an L-
system object. Figure 3(b) illustrates some of the branch features used by this system. Branch features used include branch lengths, starting position of branches, and the separation between them.

Figure 3: L-system object definitions and branch features.

**Building Decision Trees**: In order to classify an unknown object, its features must be compared with the features of known object classes. This process can be done easily and efficiently using a decision tree. The non-terminal nodes of a decision tree consist of boolean expressions, and the terminal nodes consist of class names. To classify an unknown object, we start at the root of the tree. Based upon the outcome of the boolean test at the node, the test is moved down either the right or left branch. This process continues until we reach a terminal node. The object is assigned the class specified in that node.

In previous research[8], the decision tree was created manually with some aid from the KBVision Constraint System, a commercial computer vision package. With this system, histograms of each feature can be observed, and the best splitting value for the decision tree can be found. For example, it may be found that half the plants have values less than 4.2 for feature \( f_1 \), while the other half clearly have greater values. This feature can then be used to form a node in a decision tree, with the test \( f_1 < 4.2 \) used to split the training cases into two groups. This process is repeated until all training cases have been classified.

In this research, a commercial learning system called C4.5 was used to build the decision tree. C4.5 is a series of programs that construct classification models by discovering and analyzing patterns found in records of information[12]. Data from a set of discrete classes are given to C4.5 as a training set. This training set is analyzed, and a decision tree is derived by the system. Test cases can then be provided to determine the accuracy of the created decision tree. Using a learning system to build decision is clearly advantageous since the process of deriving a decision tree for a large training set by manual methods is tedious, time consuming, and prone to human error.

**4. IMPLEMENTATION AND RESULTS**

The system to recognize the plants and trees using the stochastic L-system model is implemented on a SUN Sparcstation. The KBVision system is used for various image processing operations. Plants are generated using stochastic L-systems and are used both for training and testing. The training set contained fifty instances of each plant, with a total of 7,500 plants. The test set contained twice as many plant instances, with a total of 15,000 plants. It was decided that data sets containing three to five iteration plants would be used. The features from the plants in the training set are given to the learning system which then generates the best decision tree.

Overall, C4.5 was able to build decision trees that correctly classified the test cases with approximately 93% accuracy using shape features only and with 94% using both shape and branch information. Table 1 shows the summary of the results.

Many interesting conclusions can be made from our results. First, plants generated from stochastic L-systems can be accurately classified, even though variations do exist between individual members of a class. Next, shape information can be used with results almost as good as shape and branch features combined. It was expected that the branch statistics would not classify instances
of stochastic L-systems very accurately. This is due to the randomness in the branch lengths, locations, and frequencies. However, it was not entirely expected that the shape features alone could be used to accurately classify the plants.

Classifying objects generated using stochastic L-system grammars accurately with only shape features implies that the recognition of natural plants and trees with only shape information may be possible as well. Since the branches in natural plants are difficult to determine in general, shape features may provide a good direction for recognition. However, some shape characteristics, such as the backbone and trunk lengths, moments of inertia, and orientation, require finding major branches first.

5. SUMMARY

Recognizing plants and trees with a computer is an extremely difficult task. To aid this process, L-systems have been used to model natural plants. Previous research has shown that classifying plant-like structures generated by regular L-system is possible. Our research shows that classification is also possible with a more complicated and realistic model, the stochastic L-system. Further, it may be possible to classify natural plants and trees using their shape features only. This could prove advantageous to future research, because branches are difficult to extract from natural plant images.

6. REFERENCES


