Lecture12: The Evolution of L-System Plants

1. Examples of L-Systems
2. Evolving L-Systems to Generate Virtual Plants
3. Summary

L-Systems

- Lindenmayer proposed a model of development based on *rewriting rules* or *productions*. This model, known as L-systems, originally provided a formal description of the development of simple multicellular organisms and was later extended to higher plants.
- Central to L-systems, is the notion of rewriting, where the basic idea is to define complex objects by successively replacing parts of a simple object using a set of rewriting rules or productions. The rewriting can be carried out recursively.

Rewriting Rules

- Consider strings built of two letters a and b;
- Rewriting rules: $a \to ab$, $b \to a$;
- The rewriting process starts from a distinguished string called the axiom. Assume the axiom consist of a single letter b.
- The first step of rewriting: a;
- The second step of rewriting: ab;
- The third step of rewriting: aba

Geometric Interpretation of L-Systems

F : Move forward a step of length d. The state of the turtle changes to (x', y', a), where $x' = x + dcos(a)$ and $y' = y + dsin(a)$. A line segment between points (x, y) and (x', y') is drawn.

f : Move forward a step of length d without drawing a line. The state of the turtle changes as above.

$-$: Turn left by angle b. The next state of the turtle is $(x, y, a - b)$.

$+$: Turn left by angle b. The next state of the turtle is $(x, y, a + b)$.

All other symbols are ignored by the turtle (the turtle preserves its current state).

Creating the Koch Island

Axiom: $F + F + F + F$.

Production: $F \longrightarrow F - F + F + FF - F - F + F$.

Wildwood: The Evolution of L-System Plants

1. Wildwood generates a population of L-system representations at random.
2. The representations are rendered to the screen as plant individuals and a fitness function is applied to each individual.
3. The most fit plants survive and the plants for the next generation are constructed by combining the L-system representation of fit parent individuals from the current generation.
4. The process continues until the user intervenes.

Bracketed L-Systems

- The turtle interprets a character string as a sequence of line segments, connected “head to tail” to each other. The resulting figure remains just a single line.
- Brackets have been intriduced to model branching behavior:

  [: Push current angle/position on stack;

  ]: Pop and return to state of last push.
- In Wildwood, the rotation angle b was preset to 30 degrees.

Random String Generation

- A random rule length is first computed. For simplicity, a length between 4 and 20 was selected. Then, terms from the set $f, F, [,], +, -, A$ were selected at random to fill the string.
- To ensure balanced brackets, the probability of a] increases proportionally with respect to the remaining slots in the string and the number of right brackets necessary to balance the production rule.

Genetic Operations

- Crossover: randomly select valid substrings from each parent, and then swap them.
- Mutation: select a valid substring and then replacing the substring with a randomly generated string.

Evaluation

- Hand bred plants: fitness decided by the users. For example, how interesting plants look.
- Environment simulation: fitness produced by the simulated environment. For example, a plant's survival may depend upon its height, root depth, leaf size, ability to withstand wind, etc.

$$F(plant) = Width(plant) + S/Height(plant)$$

where S is parameter.

Summary

1. Simple geometric rules can lead to complex structures in fractals, while simple dynamical rules can lead to complex behaviour in CAs.
2. L-systems are simple program fragments that can define plantlike and other fractals.
3. Wildwood represents exploratory technology created with the intent to enrich virtual worlds.

References:

1. Gary William Flake, *The Computational Beauty of Nature*, MIT Press, 1998, pp. 59–92.
2. Mock, Kenrick. (1998). "Wildwood: The Evolution of L-System Plants for Virtual Environments." International Conference on Evolutionary Computing (ICEC'98) available from http://www.math.uaa.alaska.edu/ afkjm/