Jiehe Chen
jiehchen@pdx.edu

SCA 2D Simulation: Instrumented Growth Visualization

Project Overview

This project is inspired by Modeling Trees with a Space Colonization Algorithm (Runions, Lane & Prusinkiewicz, 2007). SCA is used to generate geometric models that simulate the branching structures of real trees and plants. The key idea is to let branches compete for “attractor points” in space to drive growth, rather than relying on traditional recursive splitting models.

The project will implement an interactive 2D version simulation system of SCA. Users can set a movable seed and attractor points on a fixed-size canvas. Attractor points support both one-time generation and dynamic point placement via mouse during runtime. The simulation advances in two-second intervals, executing multiple evolution steps and refreshing the display at each tick. It provides controls for start, stop, and reset operations, real-time parameter tuning, and displays basic runtime metrics such as node and edge counts and performance statistics.

UI components

SCA Related Buttons

- Preset distribution shapes for attraction points: rectangle, ellipse, etc.
- Attraction point distribution density: N
- Influence distance: d_i, float
- Kill distance: d_k, float
- Extrude step: d_s, float
- Tropism: g
- K-th closet: k, integer, see SCA workflow for detail
- (Competitor Advantage) K-th evaluation type for influence, bool, default true
    - True: evaluate locally
    - False: evaluate globally
- (Resource Durability) K-th evaluation type for kill, bool, default false
    - True: evaluate locally
    - False: evaluate globally

Simulation Related Buttons

- Start
- End
- Reset

Visualization

- Canvas for tree generation - two second intervals for grow
- Logging

SCA workflow

Each attraction point only affects the nearest tree node.
Each tree node may be attracted by multiple attraction points simultaneously.

- 1. Initialization
    - Create the initial tree node. It grows nodes and competes for resources with new nodes.
    - Distribute attraction points randomly in space. Each attraction point represents a potential "resource" or "target direction".
- 2. Attraction Phase
    - For each attraction point:
        i. Calculate the distance from the attraction point to all tree nodes.
        ii. If the distance is less than an influence radius d_i, the attraction point is considered to "attract" that tree node.
        iii. Record the attraction relationship.
- 3. Growth Phase
    - For each attracted tree node:
        i. Calculate the attraction direction by averaging the vectors pointing from that node to all the attraction points that attract it.
        ii. Obtain the growth direction
        v_dir = normalize(normalize(sum(normalize(direction_vectors)))+g).
        iii. Generate a new node along this direction:
        new_pos = current_pos + d_s * v_dir,
        where d_s is a fixed growth step size.
        iv. Connect the nodes by creating an edge between the original node and the new node.
- 4. Update Phase
    - For each attraction point:
        i. If the distance is less than a killing radius d_k, the attraction point is considered "occupied" and removed from the list.
- 5. Iteration & Termination
    - Repeat the "attract → grow → update" steps until all attraction points are removed (all have been "absorbed") or the maximum number of iterations is reached.

Variation:
– Each attraction point is associated with the k-th nearest competing node.
– Each attraction point is eliminated when it enters the influence region of its k-th nearest competitor.

two modes for the k-nearest evaluation:

- Globally, performs the search over all tree nodes
- Locally, restricts the search to nodes within the range defined by d_i and d_k

Concerns

The plan is to use eframe/egui, but I'm unsure about the feasibility of dynamically scattering attraction points using the mouse during the simulation.
Performance issues, if each node globally traverses every tree node, the complexity would be O(N*M).

Reference

Runions, A., Lane, B., & Prusinkiewicz, P. (2007). Modeling trees with a space colonization algorithm. In D. Ebert & S. Mérillou (Eds.), Eurographics Workshop on Natural Phenomena (pp. 63–70). The Eurographics Association.
https://algorithmicbotany.org/papers/colonization.egwnp2007.pdf

Github Repository

https://github.com/JH12333/SCA-2D-Simulation