Virtual Climbing Plants Competing for Space

Paper by Beďrich Beneš & Erik Uriel Millàn

Presented by Alan Litteneker for CS275

Virtual Plants

Very simple question . . .

Can we virtually mimic real plants?

- Highly complex natural systems, interacting in counterintuitive ways
- Incredible variation of form and structure
- True simulation would require infeasible cellular morphology modelling

Lindenmayer Systems

Early models lacked real complexity, until . . .

- A botanist named **Aristid Lindenmayer** proposed a formal language system in 1968
- A small number of simple rules can allow for a high level of complexity and variation
- Can describe anything from **Algae** to **Sierpinski Triangles** to **Oak Trees**

L-system Examples

Open L-Systems

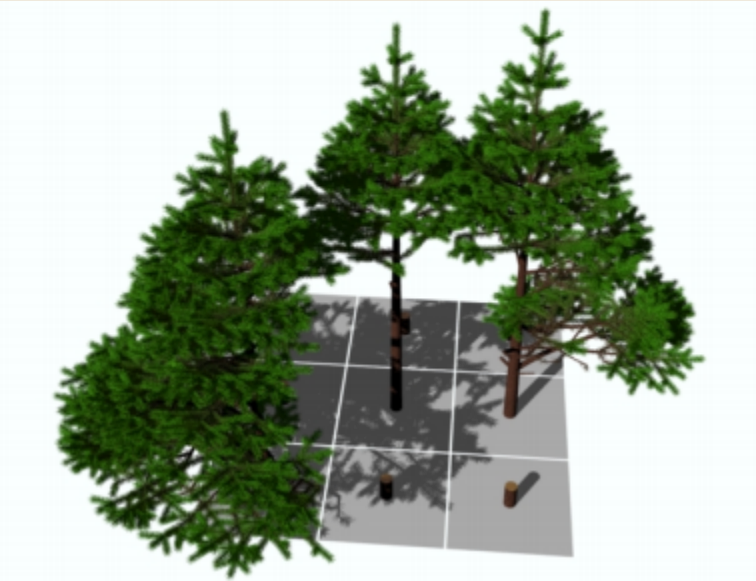
Developed by **Przemyslaw Prusinkiewicz**

Allow for interaction with environment by adding general query to global data structures

For example:

- Limited water, nutrients, etc. in ground
- Competition for light and space
- Avoidance of external and self intersection
- Structural support

Open L-system Examples



Problems with L-Systems

- Non-intuitive for human programming
- Difficult to determine structure of final shape from initial language and state
- Almost impossible to find an L-system to describe a particular structure
- Programmatically complex to allow any interaction with environment
- Computationally expensive

Is there something “better”?

Inherent difficulty with L-systems suggests domain specific simplifications

- If we are interested in a specific type of plant, can we develop a simpler system specifically for that type of plant?
- Can a more specialized system allow easier interaction with the environment?

Particle Plant Systems

- Proposed in the mid 80’s
- Generally plant type specific
- Describes plant as connected graph of simple particles
- Growth comes from human programmed data structure production rules
- Easier integration with environment system

Particle Systems

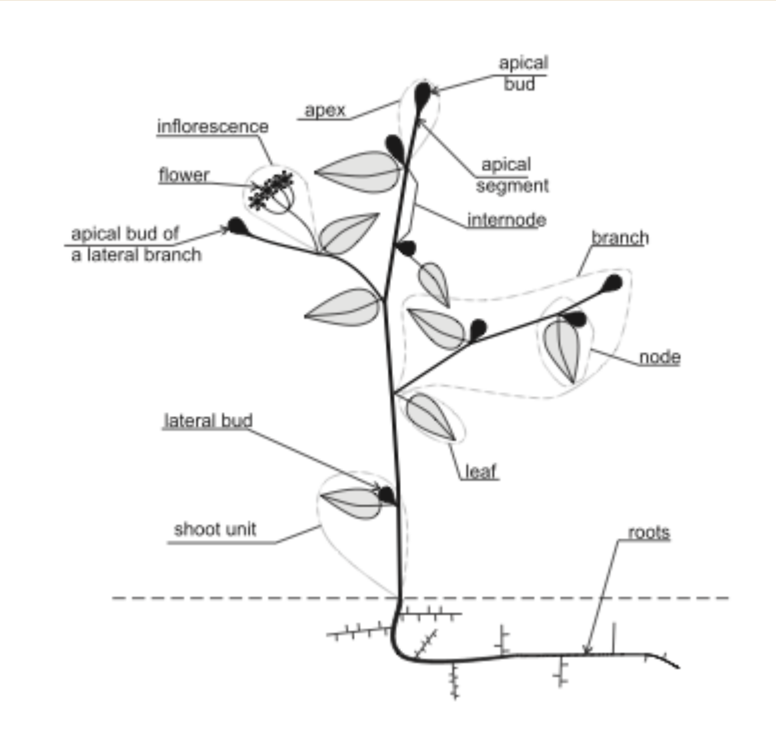
Pros:

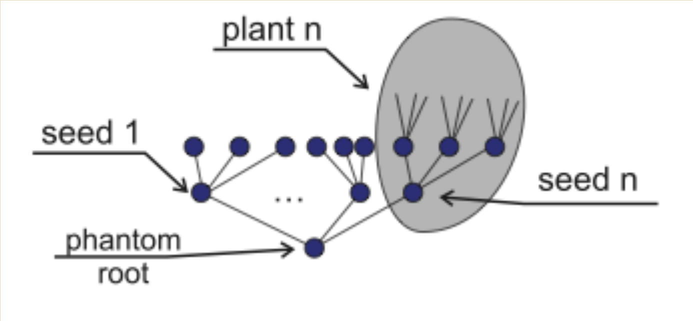
- Easy to program
- Cheap(er) to run
- Simpler environment interaction

Cons:

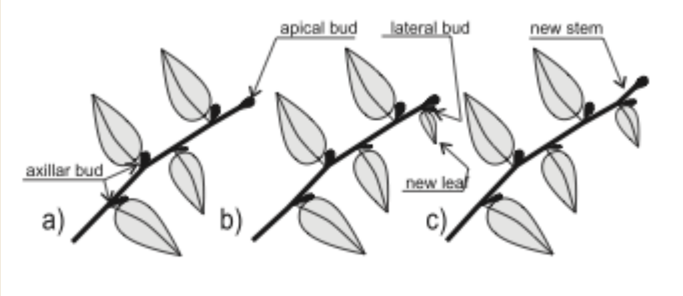
- Limits possible complexity and variation
- Doesn’t sound as cool

A Useful Simplification





Basic Process



1. Initial apex particles are seeded
2. Random walk to find new optimal bud position according to fitness function
3. Ensure new positions do not intersect with plant or existing environment
4. If unable to grow, perform traumatic reiteration
5. Occasionally seed lateral branches as fall backs
6. Go back to step 2, and repeat

Algorithm Specifics

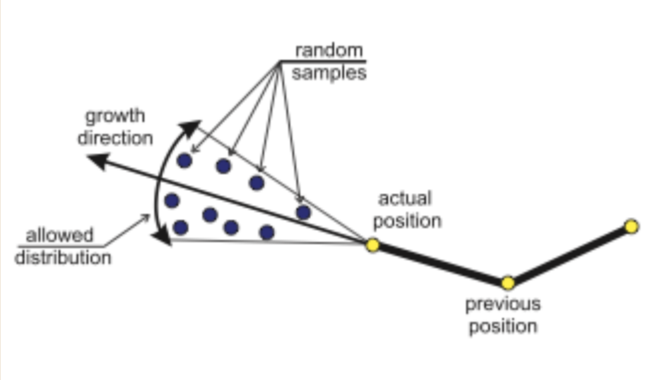
1. Put all seeds into the list of active buds
2. While the list is not empty:
3. Perform the following actions in parallel for every bud from the list of active buds:
    a. Generate *n* sample positions
    b. Evaluate the fitness function for every sample
    c. Pick the best sample
    d. Is the best position viable?

       YES: continue growing
        i. Grow there
        ii. Sometimes generate lateral buds and do not put them into the list of active buds
        iii. Put every *i*-th lateral bud into the list of active buds to achieve branching
    e. NO: perform traumatic reiteration
        i. Remove the bud from the list of active buds
        ii. Find the closest bud down on the same branch
        iii. Put this bud into the list of active buds

Fitness Function

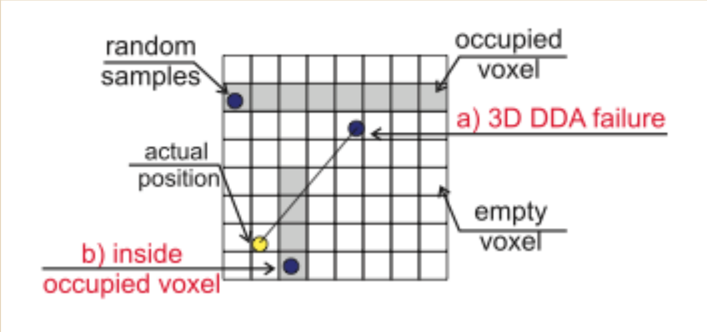
Directed random walk attempts to:

- Maximize amount of light received by node
  - Sum of number of light sources visible to bud point
  - Tested using single ray tracing through scene
- Minimize distance to non -plant objects
  - Optimize structural support



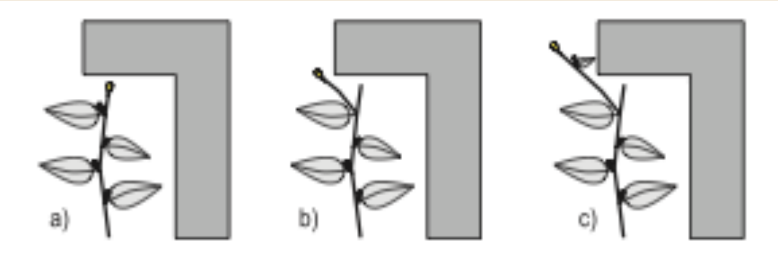
Collision Detection

- High resolution voxel map
- Voxelize the initial environment
- Add to voxel map as plant grows
- DDA line traversal for collision detection



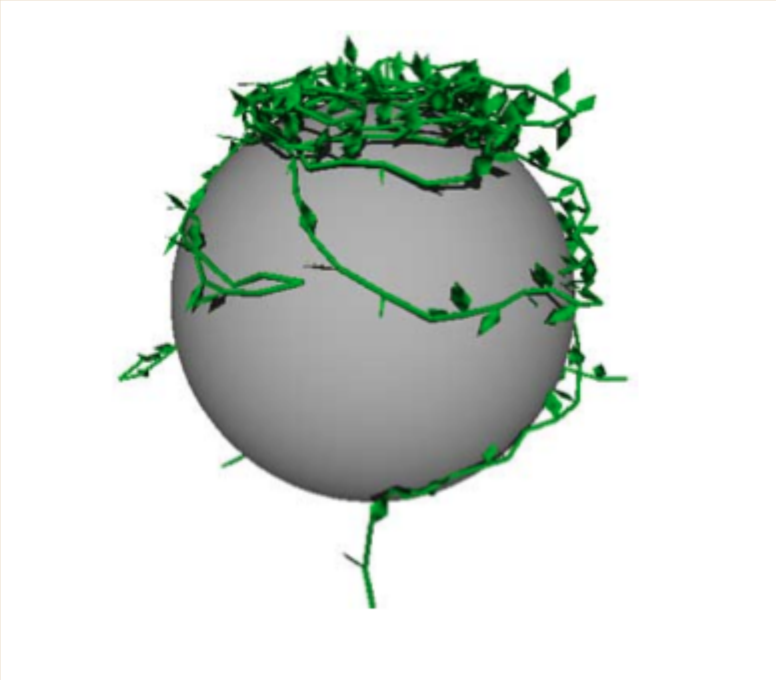
Traumatic Reiteration

- Growth in a particular direction can result in lack of space for further growth
- **Traumatic Reiteration** solves this by promoting a lower lateral bud to a growth bud whenever a growth node dies
- Simplest mechanism for growth optimization

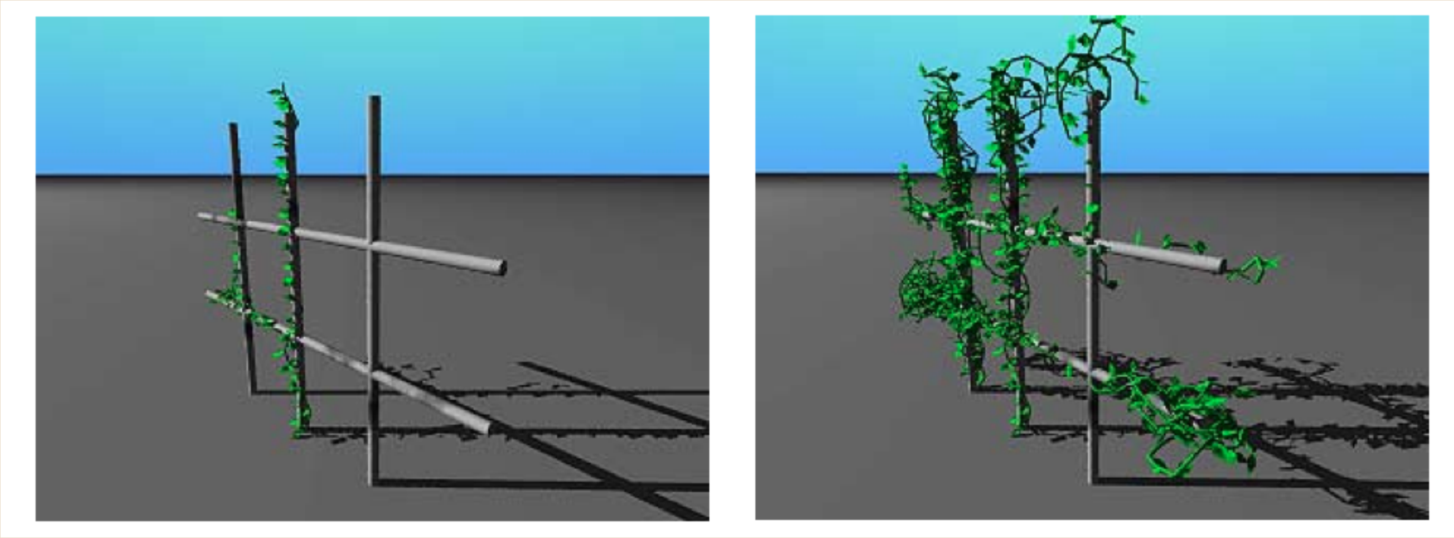


Pictures!

Maximizing Light Coverage
on a Complex Surface

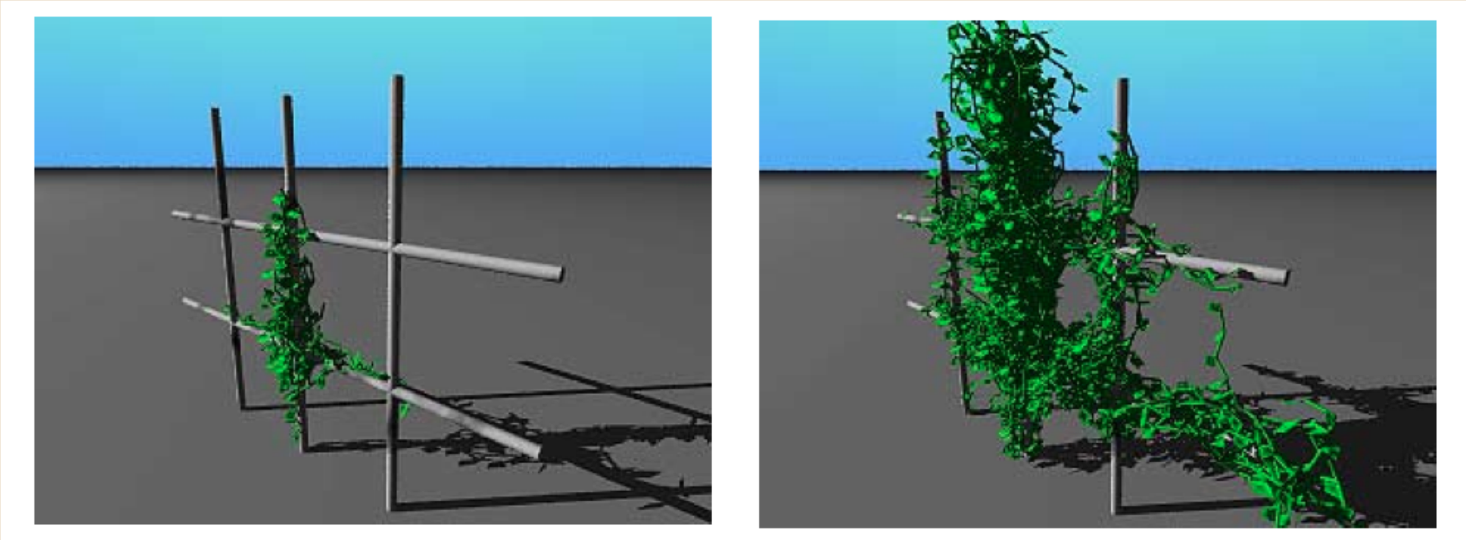


Pictures!

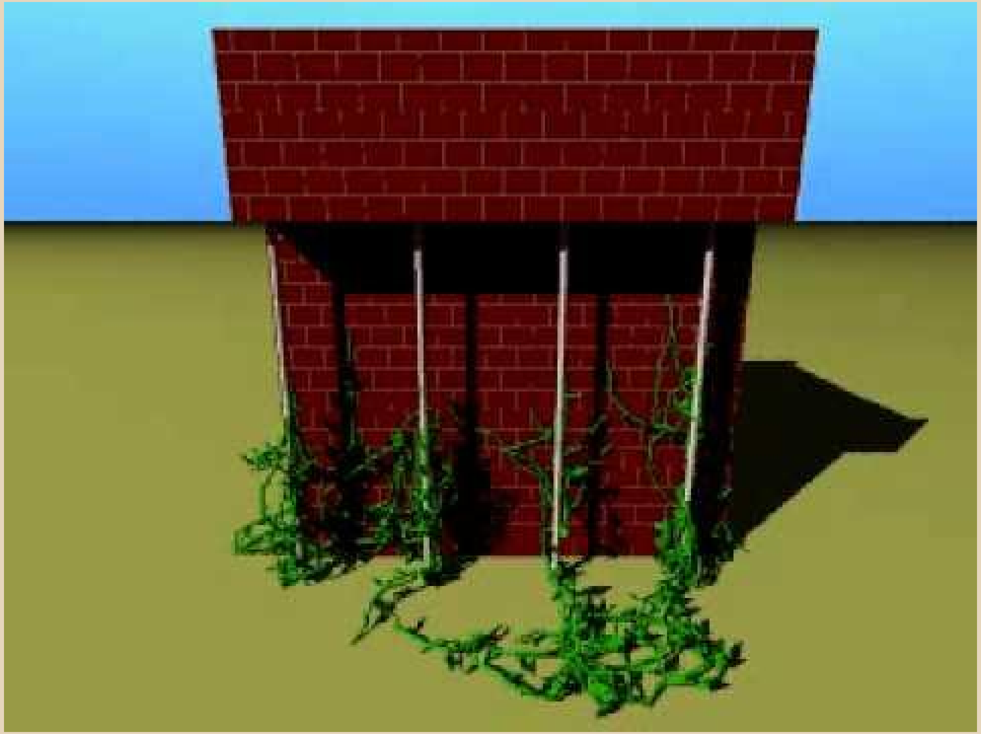


Notice problems with lack of gravity simulation.

Pictures!



Video!



https://youtu.be/F2aYwU23XBY

Questions?

The End!