# Self-organizing tree models for image synthesis

A method for generating realistic models of temperate-climate trees and shrubs

Jan Havlíček
Attila Hencz

# Introduction

# Modeling trees

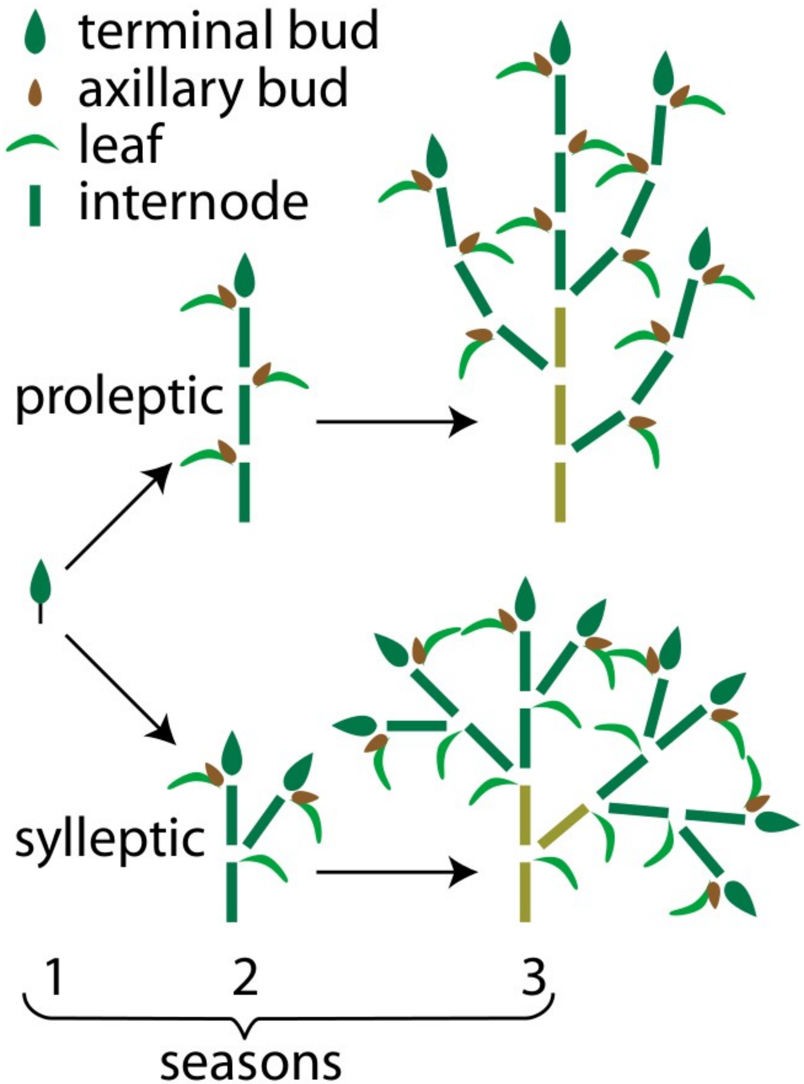
- Assumption (most methods):

  trees have a repetitive, recursive structure
  (branching pattern → form of a tree)
- However: distribution of buds is similar in all trees
- Many buds do not develop into lasting branches
- Arrangement of branches locally irregular, but well-balanced as a whole

## Complementing architectural tree models

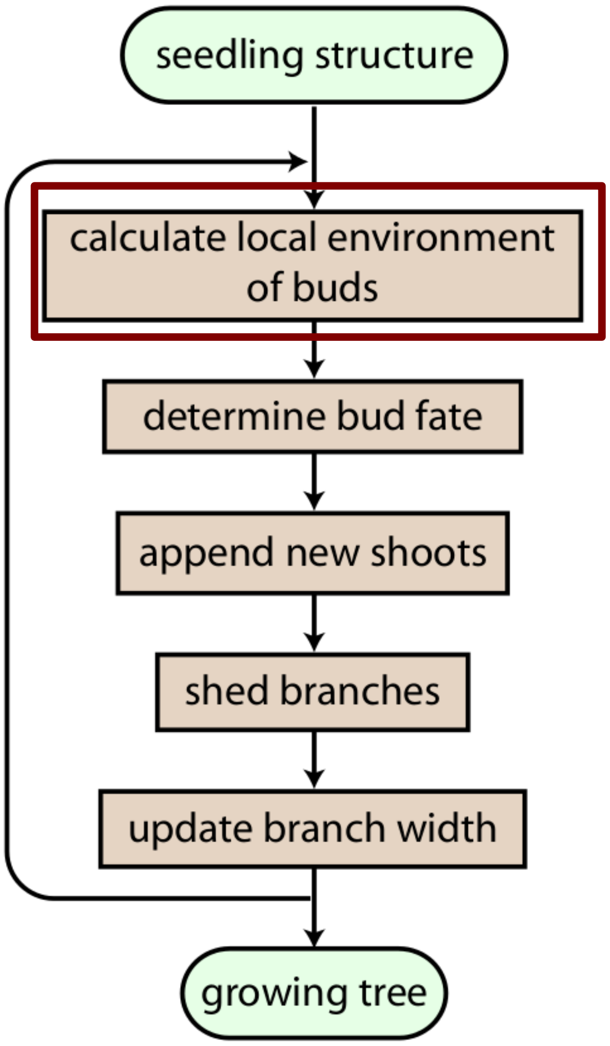
- Self-organizing character of tree development
- Every bud and branch compared with alternatives for the same role in overall tree structure
- Integrate elements of development:
  - local control of branching geometry,
  - competition of buds and branches for space or light,
  - and regulation of this competition through an internal signaling mechanism

# The modeling method




- Creating a tree structure by simulating its development
- Cycle of interactions between the tree and its environment

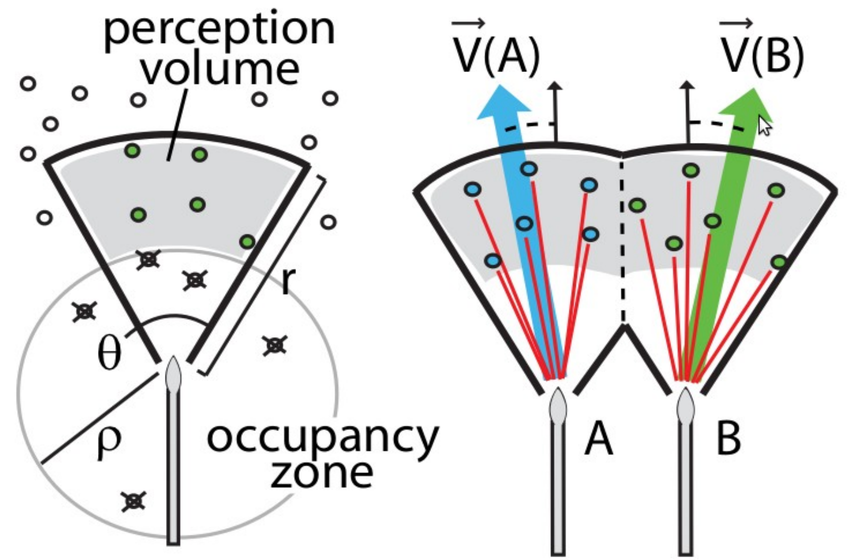
# Growth cycle: 1-Calculation of the environmental input




- Estimate the availability or quality of the space surrounding each bud (a number Q) and the optimal direction of shoot growth (a vector V ).
- 2 simple methods
  - Space colonization
  - Shadow propagation

# Environmental input: Space colonization

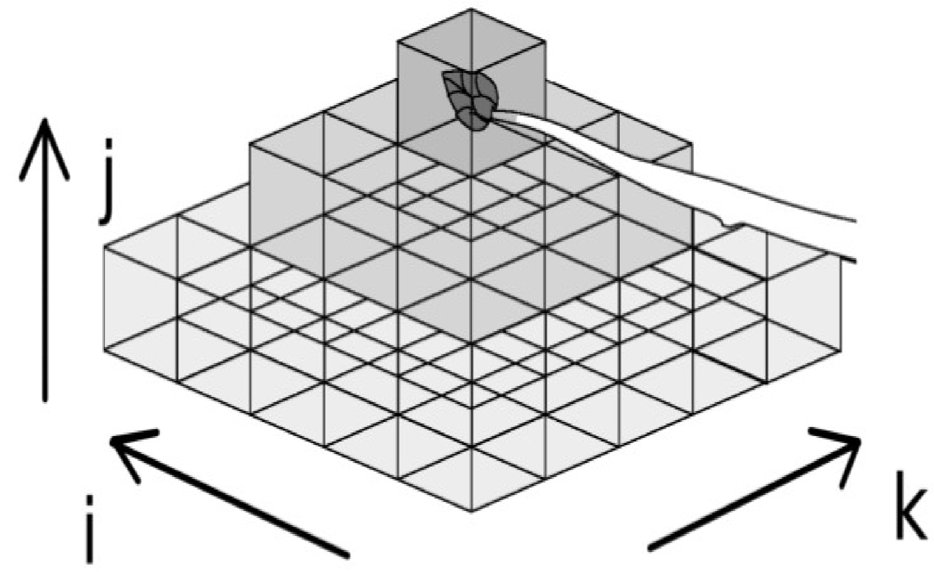
- Branches ← only from buds
- Each bud has a...
  - spherical occupancy zone (radius ρ)
  - conical perception volume (angle θ and distance r)




- Available space:
  - Init.: set of pre-generated marker points (e.g. uniform dist.)
  - In each iteration:
    - markers deleted from occupancy zones
    - buds then compete for the remaining points
      |A|>0 → Q=1, |A|=0 → Q=0; V=norm'd sum of norm'd marker vecs

# Environmental input: Shadow propagation

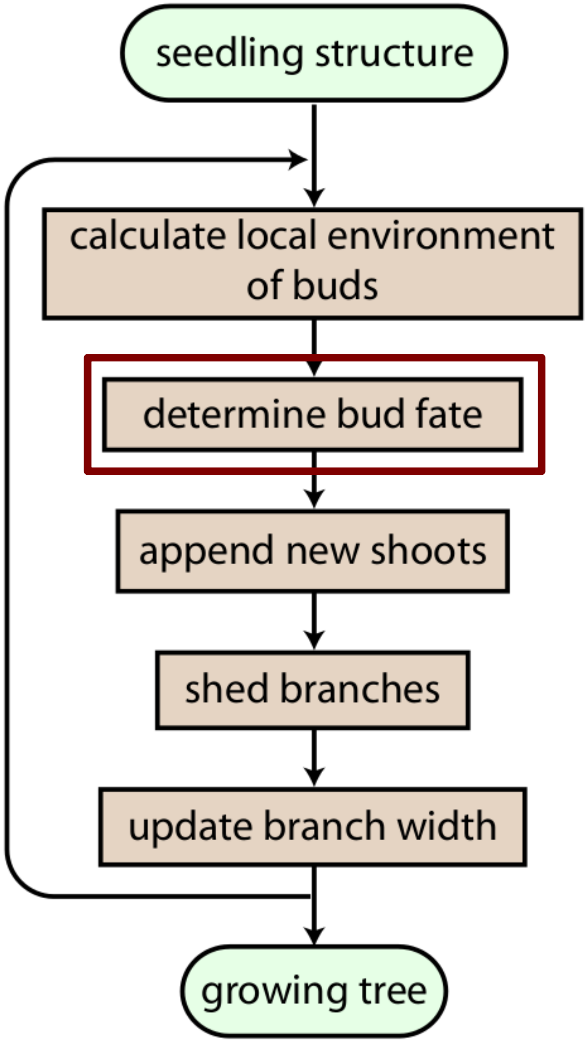
- To estimate: exposure of each bud to light
- Space → grid of voxels with "shadow value" s
  - initially s=0,
  - then: $\Delta s = ab^{-q}, q = 0, 1, \ldots, q_{max},$ $a > 0$ & $b > 1$ user-defined)




- Light exposure Q of a sample bud in a voxel:

  $Q = max(C - s + a, 0)$ , C – constant representing full exposure
- Growth dir. V: (a) neg. gradient of s; (b) voxel with lowest s among neghboring in the perception volume of the bud

# Growth cycle: 2-Calculation of bud fate

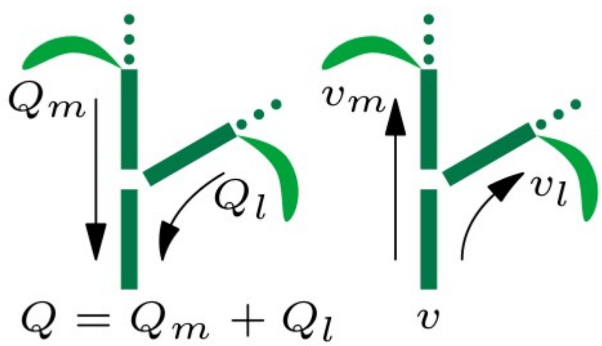


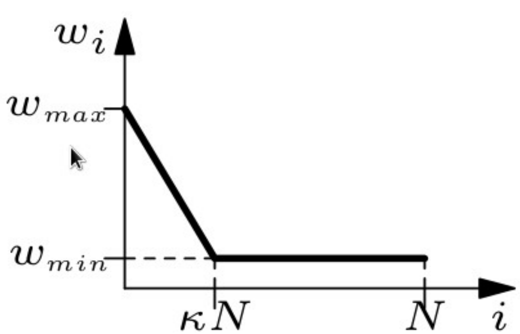

- Env. input (avail. space/light) → which buds produce how large shoots
- “Apical” control (suppression of lateral branch growth)
- 2 simple resource alloc. models:
  - Extended Borchert-Honda (BH) model
  - Priority model

# Bud fate: Extended Borchert-Honda model

$$v_m = v \frac{\lambda Q_m}{\lambda Q_m + (1-\lambda) Q_l}$$

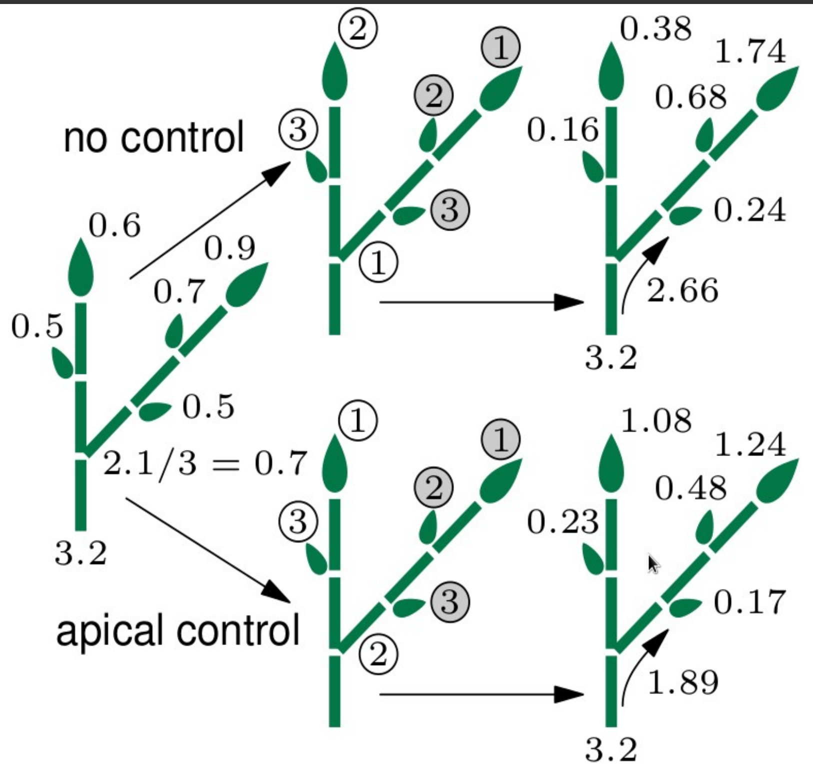
$$v_l = v \frac{(1-\lambda) Q_l}{\lambda Q_m + (1-\lambda) Q_l}$$




- Purely endogenous – distribution of a growth-inducing resource to buds
- New: distribution by amount of light received by the buds
- 2 passes:
  - Light (Q) collection toward the “root”
  - Redistribute toward the branches
- Internodes:
  - Number: n = floor(v)
  - Length: l = v/n

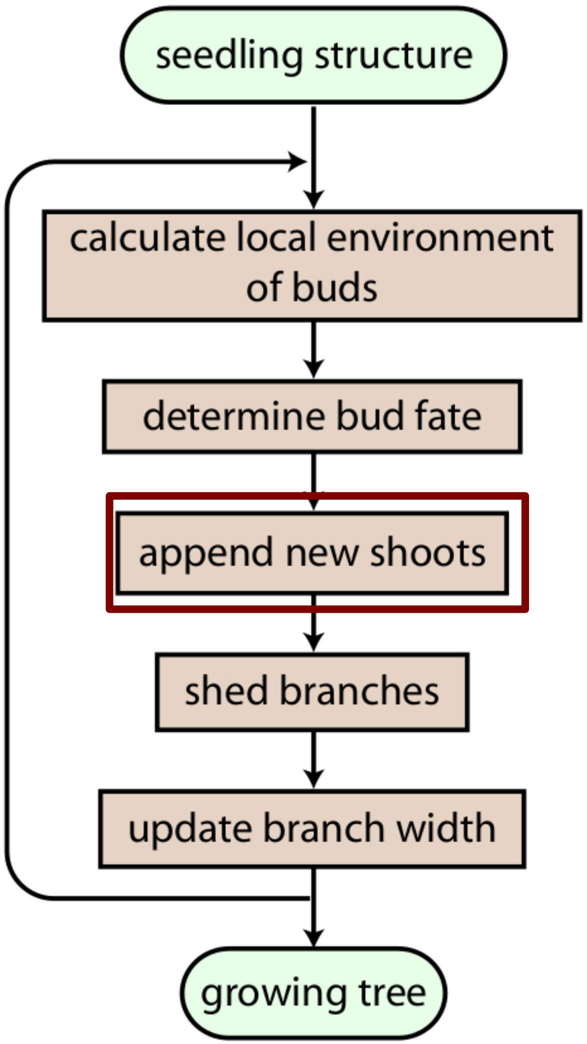
# Bud fate: Priority model

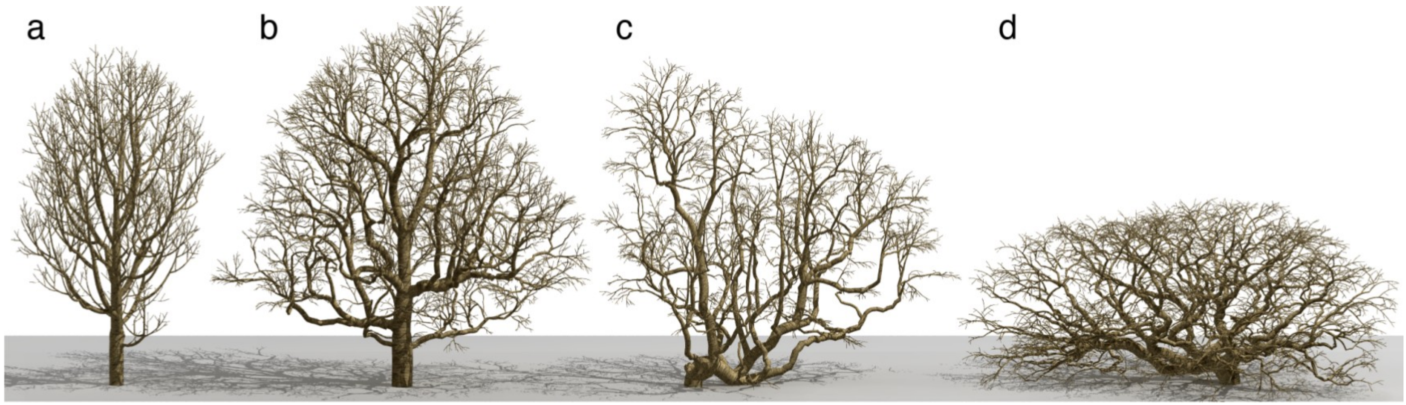



$$v_i = v \frac{Q_i w_i}{\sum_{j=1}^{N} Q_j w_j}, i = 1, 2, \ldots, N.$$

- Collect resources (light, #buds)
- Order buds by priority (avg.)
- Apical/No control (perm./dyn.)
- Distribute resources

# Growth cycle: 3-Addition of new shoots

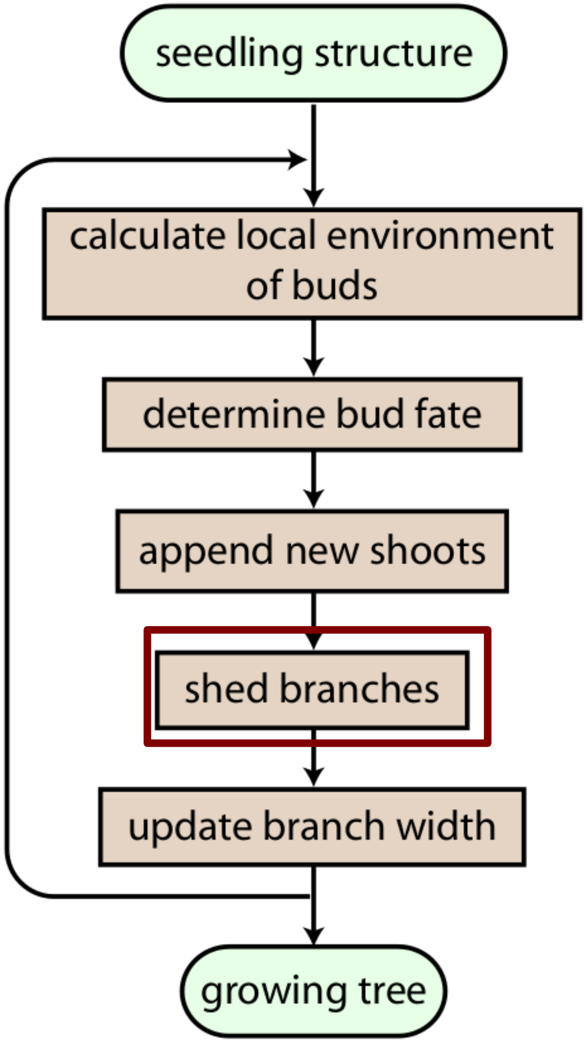



- New shoots issued in direction of buds
- Modified by optimal growth direction (V) and by tropism vector

$$\vec{d}_{shoot} = \vec{d}_{def.} + \xi \vec{V} + \eta \vec{T}$$

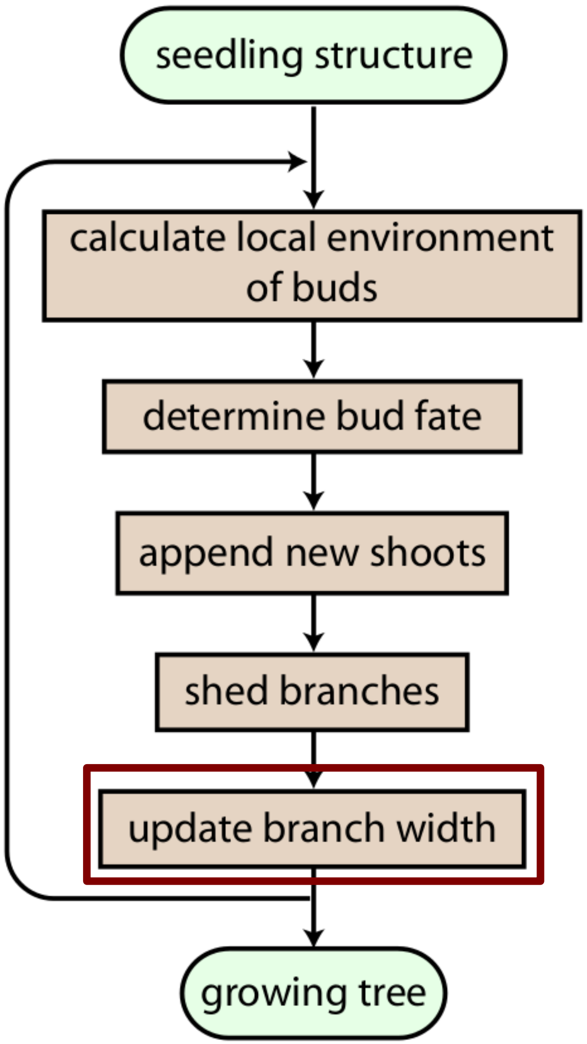
# Growth cycle: 4-Shedding of branches




- Important component of crown self-organization
- When ratio of internodes / Q in a branch falls below threshold, drop the branch
- Space col. – problem: Q binary

# Growth cycle: 4-Shedding of branches

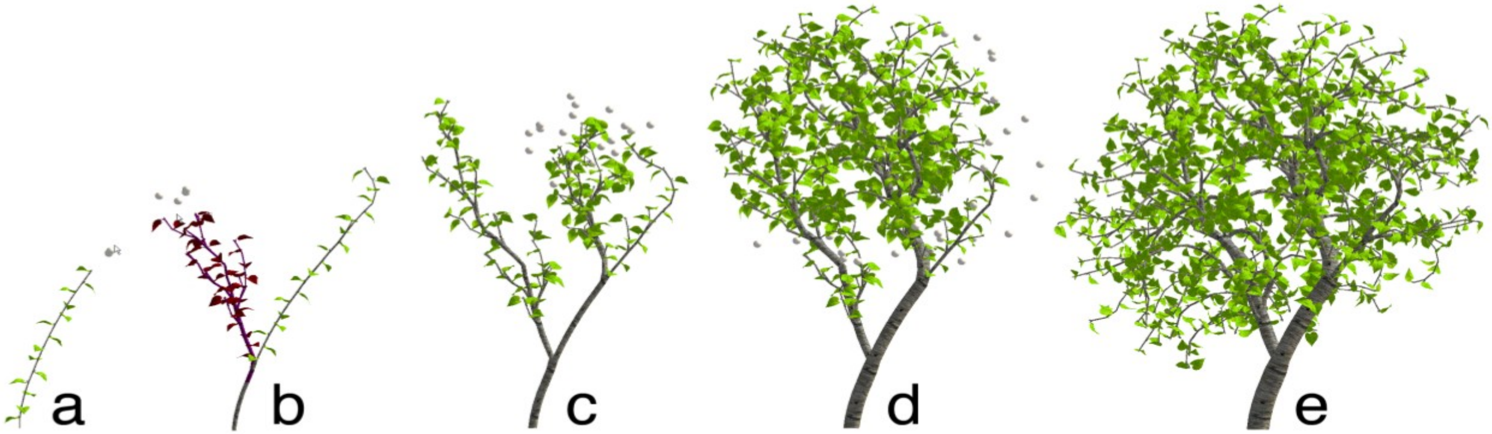
# Growth cycle: 5-Calculation of branch diameter




- Important factor affecting natural appearance
- Pipeline model (each leaf contributes, toward "root")
- Branching point: $d^n = d_1^n + d_2^n$
  n – user defined (e.g. 2-3)
- (!) d stays after shedding (memory)

# Interactive control

- Tree modification: bending & pruning
- Environment control:
  - space col.: modifying the marker point set with 3D proc. Brush)
  - shadow prop.: change light
- Enables more precise control over shape while keeping natural appearance



# Implementation

- Simple to implement (couple hundred lines of L+C and C++ for L-studio
- Tree complexity: 1000(young)-700 000(old) metamers
- Voxel grid: 200 x 200 x 200
- Markers: interactive: 1-1000; non-int.: may reach 1000000

# Performance

- Times:

| Figure | Steps | Internodes | Gen. time |
|---|---|---|---|
| left | 106 | 700 000 | 82 sec. |
| center | 90 | 642 000 | 60 sec. |
| right | 68 | 225 000 | 21 sec. |

# Conclusion

# Results

- Self-organization simplifies the modeling process
  - well-balanced branch distributions emerge automatically
- Realistic (biologic roots)
- Simple, only few variables
- Interactive modeling using environment, not tree itself

Overall, the proposed method makes it possible to generate a wide range of highly realistic trees, and control their form using a small number of parameters or interactive manipulations.

# Eye candy: sample pictures