8. L-Systems

L-Systems

Christoph Traxler 1

L-Systems

Christoph Traxler 2

8. L-Systems

L-Systems

- Introduced 1968 by the biologist Aristid Lindenmayer as a framework for studying the development of simple multicellular organisms

- Subsequently applied to investigate higher plants and plant organs
- L-systems initialy generated the topology of plants as a result of discrete development

Christoph Traxler 3

L-Systems

- Prusinkiewicz extended L-systems in various ways:
  - Parametric L-systems
  - Timed L-systems
  - Open L-systems
  - Simulation of ecosystems

- The incorporation of geometric features allows detailed modeling and a realistic visualization of plants

Christoph Traxler 4

8. L-Systems

L-Systems

- L-systems are parallel rewriting systems operating on strings of symbols
- An L-system is an ordered triplet $\langle A,\omega,P \rangle$, where
  - A is a finite set of symbols called alphabet
  - $\omega \in A^+$ is a non empty word called axiom
  - $P \subset A \times A^*$ is a finite set of productions

Christoph Traxler 5

L-Systems

- A production is written as $a \to v$ and means that each occurence of the symbol a in the current string is rewritten by v
- In a derivation step all symbols of the current string are replaced simultaneously by the proper production
- Difference compared with Chomsky-systems: Only one symbol is replaced in each derivation step

Christoph Traxler 6

8. L-Systems

L-Systems

- If there exists no production for a symbol a then the identic production a→a is applied
- The set of all strings that can be derived from an L-system is called formal language of the L-system
- An L-system is called deterministic (dL-system) if and only if there exists for each $a \in A$ exactly one $v \in A^*$ so that $a \to v$

Christoph Traxler 7

L-Systems

- Example: {a,b,c};a;{p_1:a→ab,p_2:b→ac}

  ω_0 = a

  ω_1 = ab

  ω_2 = ab ac

  ω_3 = abacabc

  ω_4 = abacabcabacc
- How are derived strings transformed into a graphic ?

Christoph Traxler 8

8. L-Systems

Turtle Graphics

- The turtle is a drawing tool for the geometric interpretation of a derived string
- When reading the string the turtle performs specific commands for certain symbols and changes its state
- Its state is defined by the triplet (x,y,α), where x,y defines its position on the plane and the angle α its orientation

Christoph Traxler 9

Turtle Graphics

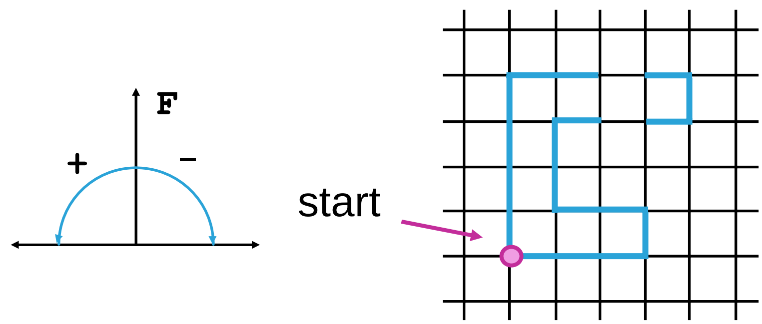
- The turtle obeys to the following commands:
  - `F` draw a line of length d, the state changes to $(x+d\cos\alpha, y+d\sin\alpha, \alpha)$
  - `f` move forward a step of length d without drawing
  - `+` turn left by angle δ, the state changes to $(x,y,\alpha+\delta)$
  - `-` turn right by angle δ, the state changes to $(x,y,\alpha-\delta)$

Christoph Traxler 10

8. L-Systems

Turtle Graphics

- Starting conditions for the turtle: Initial state, d and δ
- Example: **FFFF-FFfF-F-FfF+FF+FF-F-FFF**, d=1, δ =90°



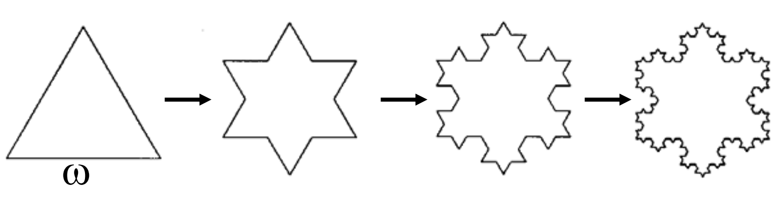
Classic Fractals by L-Systems

- Using the turtle commands as alphabet
- Construction of Koch's island:

  δ = 60°, d depends on derivation length

  ω = **F--F--F** //Initiator

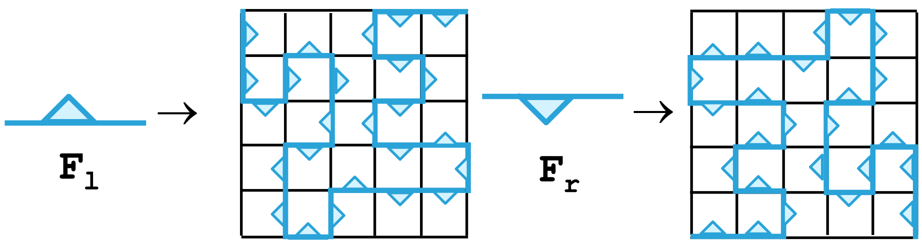
  F → **F+F--F+F** //Generator



8. L-Systems

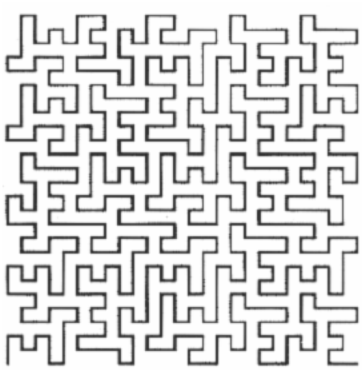
Classic Fractals by L-Systems

- Space Filling Curves
  - Edge Replacement: Using different symbols with the same geometrical semantic to controll derivation
  - Construction of the E-curve

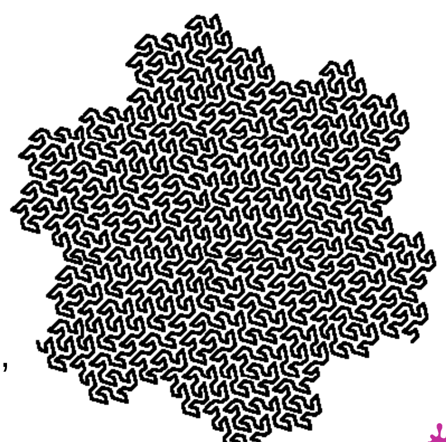


Christoph Traxler 13

Classic Fractals by L-Systems



E-curve, n=2, δ=90°

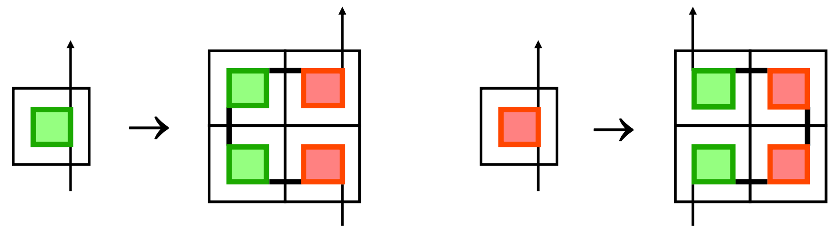


Hexagonal Gosper curve, n=4, δ=60°

Christoph Traxler 14

Classic Fractals by L-Systems

- Space Filling Curves
  - Node Replacement: Using additional symbols which are ignored by the turtle to controll derivation
  - Construction of the Hilbert curve



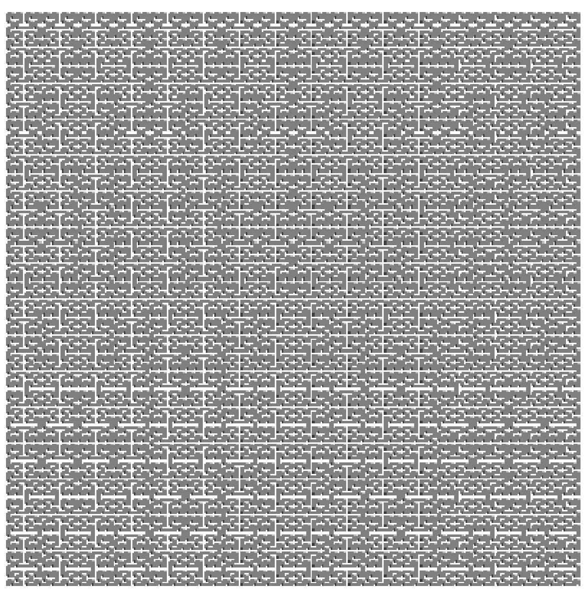
L→+RF−LFL−FR+   R→−LF+RFR+FL−

ω:L

Christoph Traxler 15

Classic Fractals by L-Systems

- Hilbert curve



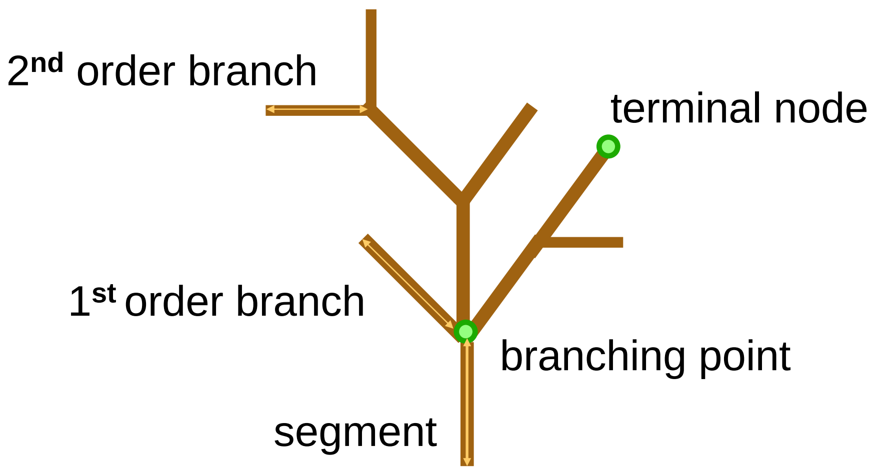
Christoph Traxler 16

Branching Structures

- The turtle constructs objects incrementaly, thus there must be a mechanism to move it back to the branching points
- This is achieved by additional commands:
  - [push the state of the turtle onto a stack
  -] pop the last state from the stack and make it to the current state of the turtle
- Can be used to store and restore other information as well

Christoph Traxler 17

Branching Structures

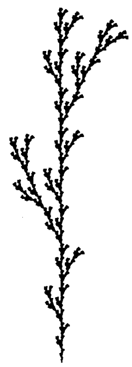


F[+F][-F[-F]F]F[+F[+F][-F][-F]

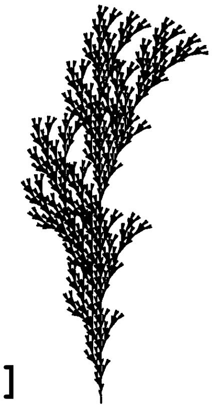
Christoph Traxler 18

8. L-Systems

Branching Structures



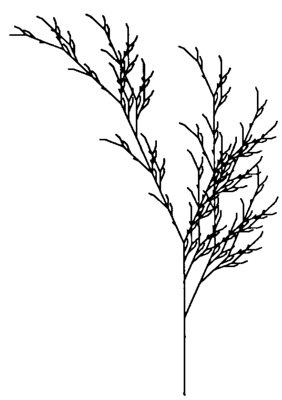
n=5, δ=25.7°
ω:**F**
F→F[+F]F[-F]F



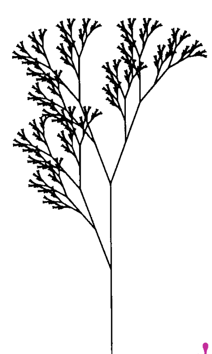
n=5, δ=20°
ω:**F**
F→F[+F]F[-F][F]

Branching Structures

- Node replacement



n=7, δ=20°
ω:**X**
X → F[+X]F[-X]+X
F → FF



n=5, δ=22.5°
ω:**X**
X → F-[[X]+X]+F[+FX]-X
F → FF

Non Deterministic L-Systems

- At least one symbol has more than one production
- There must be a mechanism which selects one of the productions for each symbol during derivation:
  - Stochastic L-systems
  - Context sensitive L-systems
  - Parametric L-systems

Stochastic L-Systems

- Different productions for a symbol are selected randomly
- During parallel replacement a new random number is used for each symbol
- A probability is assigned to each production
- Used to generate variations among individuals of a species

Stochastic L-Systems

ω: F

p_1: $F \xrightarrow{0.33} F[+F]F[-F]F$

p_2: $F \xrightarrow{0.33} F[+F]F$

p_3: $F \xrightarrow{0.33} F[-F]F$

Christoph Traxler 23

Stochastic L-Systems

Christoph Traxler 24

8. L-Systems

Context Sensitive L-Systems

- The selection of a production for a symbol depends on the adjacent symbols in the current string
- A context-sensitive production is written as **A*<s>A*** $\rightarrow$ **A+**, example:

```
xy < a > z  → ab
x  < a > yz → aa
xyazabyxayzab ⇒ xyabzabyxaayzab
```


Context Sensitive L-Systems

- Is used to simulate the propagation of signals (hormones, nutrients) between parts of a plant

```
ω:  baaaaaaa          baaaaaaa
p1: b<a → b           abaaaaaa
p2:   b → a           aabaaaaa
                      aaabaaaa
                      aaaabaaa
                      aaaaabaa
                      aaaaaaba
```

8. L-Systems

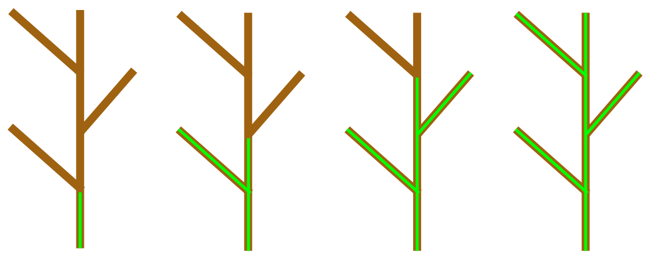
Context Sensitive L-Systems

- Branching structures: Topology has to be considered when searching for the context

`# ignore +-`

ω: $F_b[+F_a]F_a[-F_a][+F_a]F_a$

p: $F_b < F_a \rightarrow F_b$



Christoph Traxler 27

Parametric L-Systems

- Symbols are assocciated with a finite set of parameters
- Parameter values are used to select productions and to controll turtle geometry
- Incorporation of geometrical features, → the final shape is a result of the derivation
- Important for plant modeling, because the geometry of a plant is the result of its developmental process

Christoph Traxler 28

8. L-Systems

Parametric L-Systems

- A parametric L-system (pL-system) is defined as ordered quadruplet $\langle A,\Sigma,\omega,P\rangle$, where
  - A is an alphabet of symbols
  - Σ is a finite set of parameters
  - $\omega \in (A \times R^*)^+$ is the axiom
  - $P \subset ((A \times \Sigma^*) : C(\Sigma) \rightarrow (A \times E(\Sigma))^*$ is the set of productions
- $C(\Sigma)$ denotes a logical and $E(\Sigma)$ an arithmetic expression with parameters from Σ

Christoph Traxler 29

Parametric L-Systems

- PL-systems operate on strings of modules which are parametric symbols, i.e. $(A \times R^*)$
- Derivation: A production p matches a module m of a string if:
  - The symbols of m and of the predecessor of p are the same
  - The number of real values in m is equal to the number of parameters in the predecessor of p
  - The condition of p is true

Christoph Traxler 30

Parametric L-Systems

- Modules are replaced as follows:
  1. All parameters in the production are set to the real values of the module
  2. The condition is evaluated
  3. If the condition is true, then the arithmetic expressions in the successor are evaluated
  4. The new modules with the resulting real values are filled into the new string

Parametric L-Systems

```
ω:   B(2)A(4,4)
p1:  A(x,y): y<=3  → A(x*2,x+y)
p2:  A(x,y): y>3   → B(x)A(x/y,0)
p3:  B(x)  : x<1   → C
p4:  B(x)  : x>=1  → B(x-1)
```

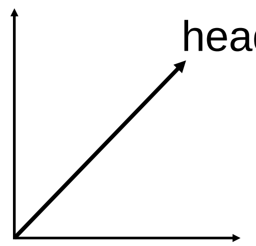
Result:

```
B(2)A(4,4)
B(1)B(4)A(1,0)
B(0)B(3)A(2,1)
C   B(2)A(4,3)
C   B(1)A(8,7)
C   B(0)B(8)A(1.142,0)
```

8. L-Systems

The 3D-Turtle

- Virtual construction tool in space
- Its state is defined by a local coordinate system (transformation matrix)
- Within pL-systems the commands depend on parameters



The 3D-Turtle Commands

F(d) Position a cylinder of length d along the h-axis, the turtle is translated to the endpoint of the cylinder

F(d,r) Position a cylinder with of length d and radius r

f(d) Move a step of length d along the h-axis

8. L-Systems

The 3D-Turtle Commands

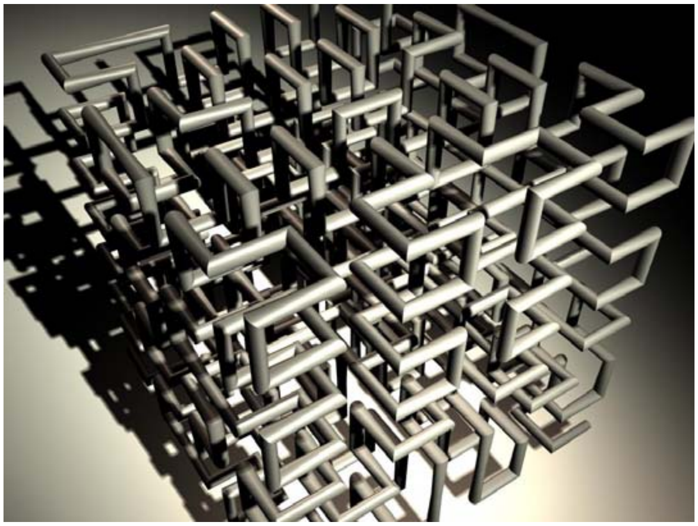
+(δ)	Rotate around the u-axis by angle δ
&(δ)	Rotate around the r-axis by angle δ
/(δ)	Rotate around the h-axis by angle δ
|	Turn around by 180°, - the same as +(180)
[ ]	Have the same effect as for the 2d-turtle

- Turtle commands can be extended by arbitrary parameters

Christoph Traxler 35

The 3D-Turtle

- A 3D-extention of the Hilbert curve



Christoph Traxler 36

8. L-Systems

A Tree Model

- Monopodial branching structures

$$\omega : A(1,10)$$

$$p_1 : A(l,w) \rightarrow F(l,w)[\&(a_0)B(l*r_2,w*w_r)]/(d)A(l*r_1,w*w_r)$$

$$p_2 : B(l,w) \rightarrow F(l,w)[+(-a_1)\$C(l*r_2,w*w_r)]C(l*r_1,w*w_r)$$

$$p_3 : C(l,w) \rightarrow F(l,w)[+(a_1)\$B(l*r_2,w*w_r)]B(l*r_1,w*w_r)$$

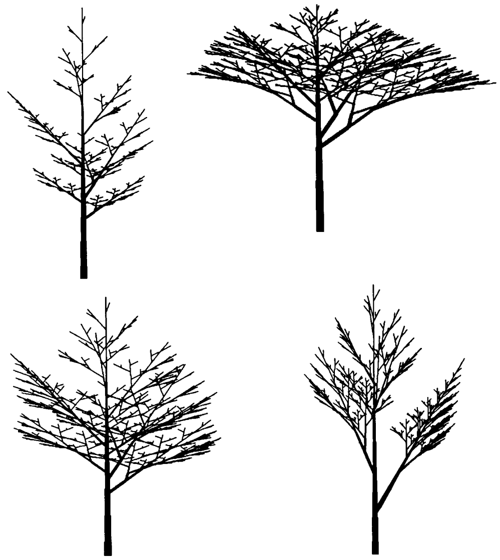
- The $ command turns the turtle back into a horizontal position

A Tree Model

- Parameters of the model:
  - r_1: Contraction ratio for the trunk
  - r_2: Contraction ratio for the branches
  - a_0: Branching angle from the trunk
  - a_1: Branching angle for lateral axes
  - d: Divergence angle
  - w_r: Width decrease rate

8. L-Systems

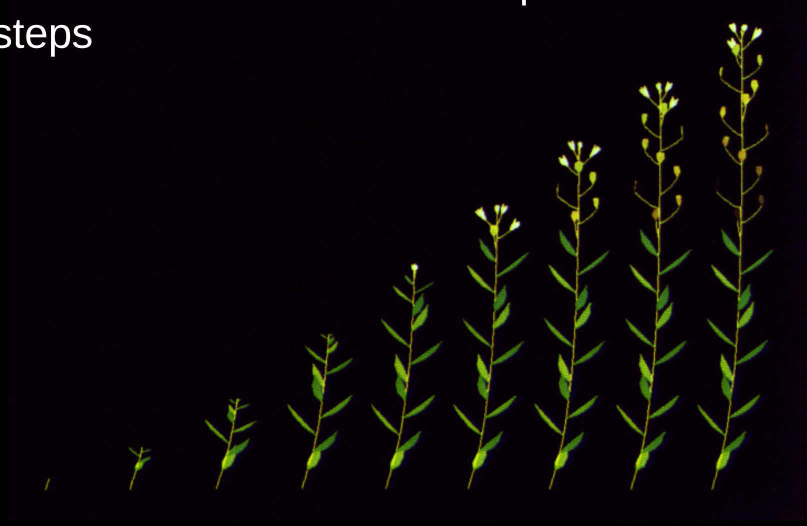
A Tree Model



Christoph Traxler 39

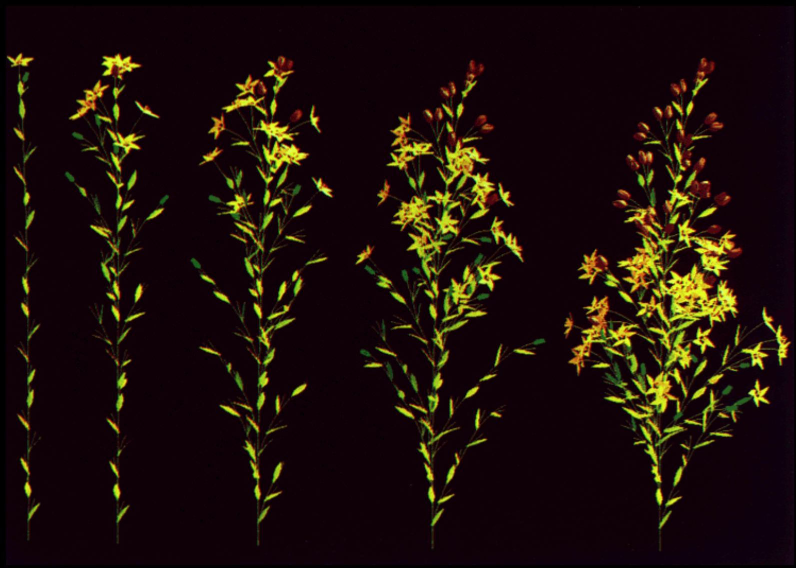
Development of Inflorescences

- Correct simulation of development in discrete steps



Christoph Traxler 40

Development of Inflorescences



Phyllotaxis

- Patterns of arrangements (leaves, seeds, ...)
- Connected to Fibonacci numbers

Phyllotaxis

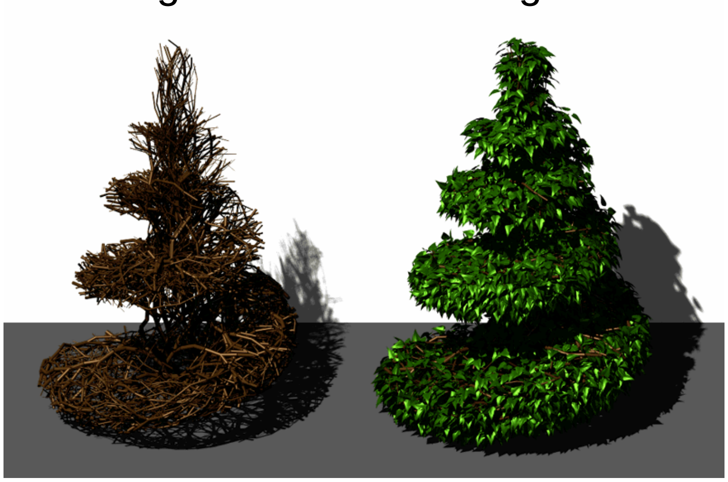
Environment Sensitive Growth

- Green's voxel space automata
- Not possible with pL-systems

8. L-Systems

Environmentally-Sensitive L-Systems

- The propagation of signals is used to prune branches against a surrounding surface



Christoph Traxler 45

Environmentally-Sensitive L-Systems

- Query modules check the environment
- They are evaluated after each derivation step to deliver components of the turtle's state
- These values can be used to select productions (e.g. trigger a signal, terminate growth)

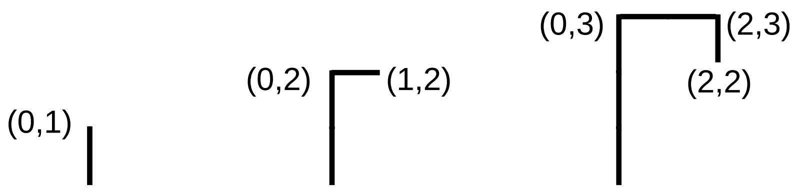
Christoph Traxler 46

8. L-Systems

Environmentally-Sensitive L-Systems

```
ω:   A;  p1:    A→ F(1)?P(x,y)-A;  p2:F(k)→
                F(k+1)

F(1)?P(*,*)-A
F(1)?P(0,1)-A
F(2)?P(*,*)-F(1)?P(*,*)-A
F(2)?P(0,2)-F(1)?P(1,2)-A
F(3)?P(*,*)-F(1)?P(*,*)-F(1)?P(*,*)-A
F(3)?P(0,3)-F(1)?P(2,3)-F(1)?P(2,2)-A
```

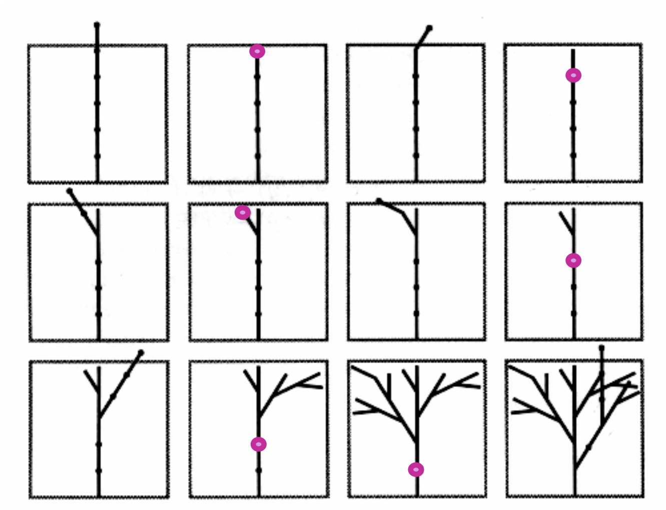


Synthetic Topiary

- Pruning of a 2D branching structure against a square

```
ω: FA?P(x,y)    // first segment
// continue growth
p1: A>?P(x,y):!prune(x,y) →. @F/(180)A
// prune last segment
p2: A>?P(x,y): prune(x,y) →. T%
p3: F>T → S
// signal back propagation
p4: F>S → SF
p5: S   → ε      // delete signal
p6: @>S → [+FA?P(x,y)] // triggers growth
```

Synthetic Topiary



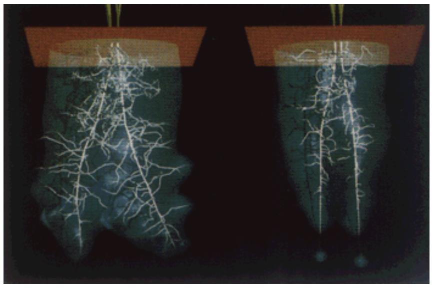
Synthetic Topiary

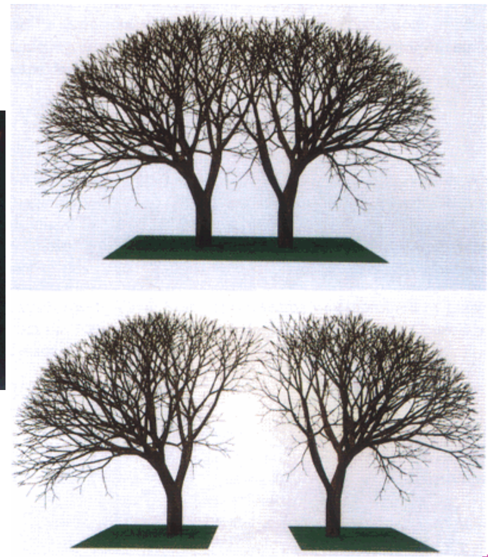
Synthetic Topiary

Open L-Systems

- Extension to general query modules and auxilary data structures
- Check for self intersection and avoid them
- Green’s voxel space automata can be realized
- Check for biological relevant conditions (soil composition, water, light, wind, ...)

Open L-Systems





Fractals vs. Graftals

- L-systems are a more general modeling technique for recursive defined objects, which were called graftals by Smith (1984)
- Plants are not fractals, because:
  - They do not have infinite detail
  - They only have a very limited degree of self similarity
  - They have not the scaling properties of fractals

Fractals vs. Graftals

- PL-Systems are suitable for modeling objects with a repetitive structure:
  - Plants
  - Terrains (fractal landscapes)
  - Linear fractals (IFS-attractors)
  - Architecture
  - Seashells
  - Particle systems

Efficient Rendering of pL-Systems

- Use Directed Cyclic Graphs (DCG) as object representation
- Object instancing: Avoids to build up the whole scene in memory
- Similar to Hart and deFanti's method for ray tracing of linear fractals and Kajiya's method for fractal terrain
- Problem: L-Systems are usually not contractive

8. L-Systems

Efficient Rendering of pL-Systems

- Components
  - Selection node: Each symbol is transformed into a selection node, which joins all the productions of the symbol and selects the proper one in each cycle
  - Transformation node: Map into local coordinate system
  - Calculation node: Modify parameters, which affect production selection and transformations

Christoph Traxler 57

Efficient Rendering of pL-Systems

- PL-System for Sierpinsky tetrahedron

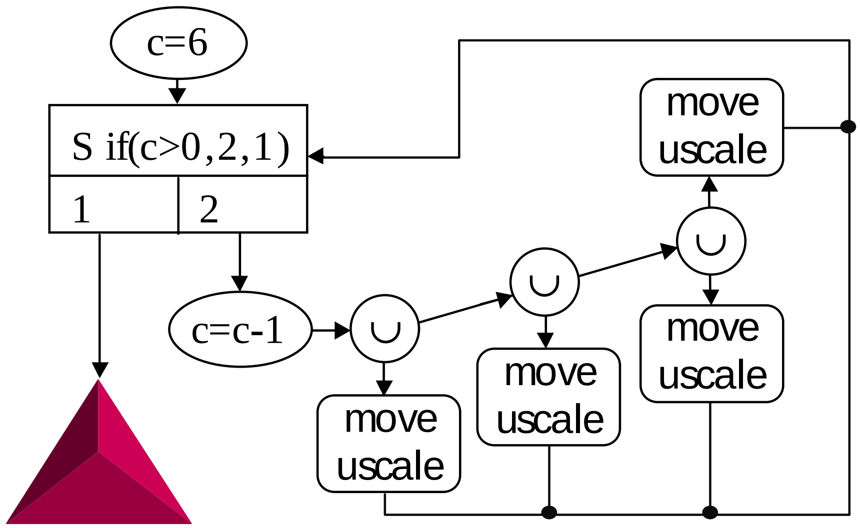
```
{c=6} S if(c>0,2,1)
1: S → tetrahedron
2: S → {c=c-1}(move(0.5,0.5,0.5)
               uscale(0.5)S ∪
               move(-0.5,-0.5,0.5)
               uscale(0.5)S ∪
               move(0.5,-0.5,-0.5)
               uscale(0.5)S ∪
               move(-0.5,0.5,-0.5)
               uscale(0.5)S)
```

Christoph Traxler 58

8. L-Systems

Efficient Rendering of pL-Systems

- DCG for Sierpinsky tetrahedron



Christoph Traxler 59

Efficient Rendering of pL-Systems

- PL-System for a simple branching structure

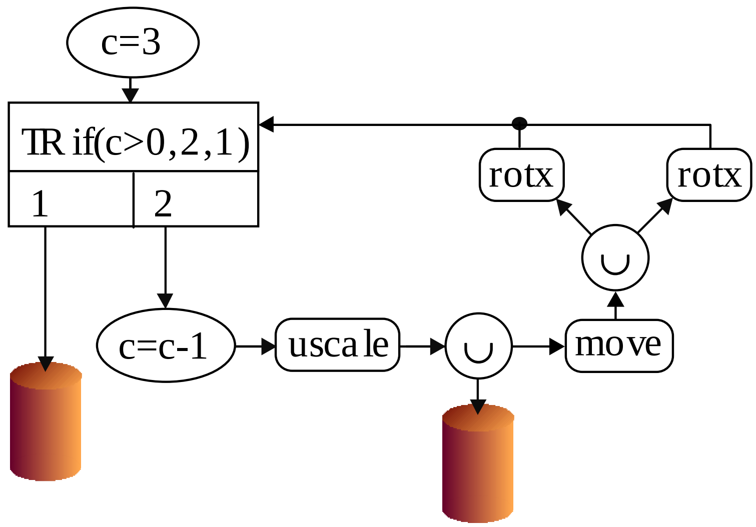
```
{c=6} TR if(c>0,2,1)
1: TR →     cylinder
2: TR →     {c=c-1}
            (uscale(0.9) cylinder ∪
             move(0,0,0.9)
            (rotx(26)TR ∪ rotx(-32)TR))
```

Christoph Traxler 60

8. L-Systems

Efficient Rendering of pL-Systems

- DCG for a simple branching structure



Christoph Traxler 61

Efficient Rendering of pL-Systems

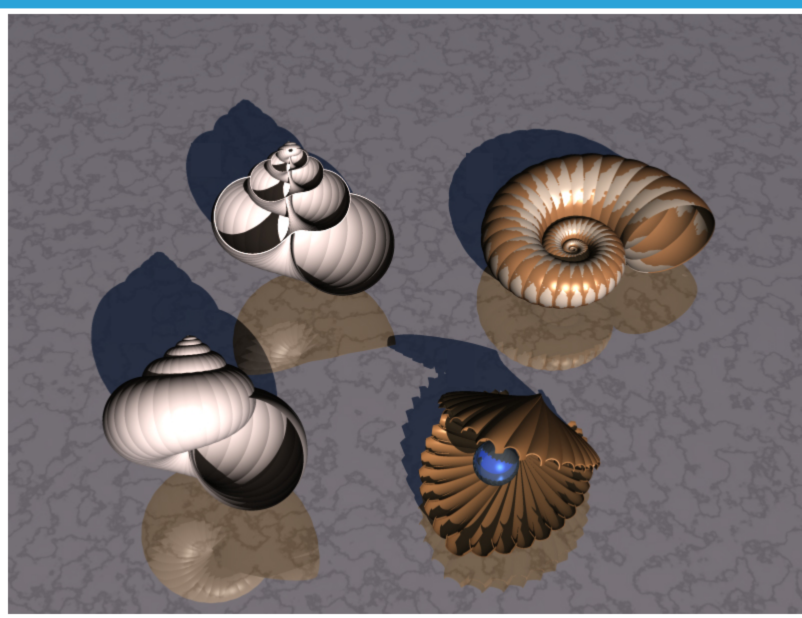
- Ray Tracing:
  - Traverse graph for each ray
  - Evaluate condition of selection nodes and follow corresponding path
  - Evaluate calculation nodes
  - Apply inverse transformations of transformation nodes to ray and accumulate them
  - Intersect ray with primitive object
  - Back transform normal vector

Christoph Traxler 62

Efficient Rendering of pL-Systems

- Optimization not trivial because of non contractive mappings
- For contractive mappings the methods is equivalent with those of Hart and deFanti
- Accumulate bounding boxes for transformation nodes in pre-processing and use their union for each traversal depth
- Fill a grid during pre-processing
- Grid resolution is used to balance memory consumption and rendering performance

Efficient Rendering of pL-Systems



8. L-Systems

Web Sites

- www.cg.tuwien.ac.at/research/rendering/csg-graphs/index.html
- algorithmicbotany.org
- www.xfrogdownloads.com

Christoph Traxler 65