EUROGRAPHICS 2002

Tutorial TH2: Modeling and Rendering of Synthetic Plants

Oliver Deussen, Dresden University of Technology

Published by
The Eurographics Association
ISSN 1017-4565

The European Association for Computer Graphics
23rd Annual Conference

EUROGRAPHICS 2002

Saarbrücken, Germany
September 2–6, 2002

Organized by

EUROGRAPHICS
THE EUROPEAN ASSOCIATION
FOR COMPUTER GRAPHICS

INFORMATIK
Max-Planck-Institut
für Informatik
Saarbrücken, Germany

Universität des Saarlandes
Germany

International Programme Committee Chairs
George Drettakis (France)
Hans-Peter Seidel (Germany)

Conference Co-Chairs
Frits Post (The Netherlands)
Dietmar Saupe (Germany)

Honorary Conference Co-Chairs
Jose Encarnação (Germany)
Wolfgang Straßer (Germany)

Tutorial Chairs
Sabine Coquillart (France)
Heinrich Müller (Germany)

STAR Report Chairs
Dieter Fellner (Germany)
Roberto Scopignio (Italy)

Lab Presentation Chairs
Günther Greiner (Germany)
Werner Purgathofer (Austria)

Industrial Seminar Chairs
Thomas Ertl (Germany)
Bernd Kehrer (Germany)

Günter Enderle Award Committee Chair
François Sillion (France)

Conference Game Chair
Nigel W. John (UK)

John Lansdown Award Chair
Huw Jones (UK)

Conference Director
Christoph Storb (Germany)

Short/Poster Presentation Chairs
Isabel Navazo (Spain)
Philipp Slusallek (Germany)

Local Organization
Annette Scheel (Germany)
Hartmut Schirmacher (Germany)

Modelling and Rendering of Synthetic Plants

Oliver Deussen

Faculty of Computer Science
Dresden University of Technology, Germany

deussen@inf.tu-dresden.de
www.inf.tu-dresden.de/cgm, www.greenworks.de

Introduction

Plant geometries:

→ important for many indoor and outdoor scenes
→ hard to generate and to manipulate
→ tricky to convert
→ today: seldomly used
→ tomorrow: standard of most animation systems

Content

→ Plant modelling, a survey
→ Modelling of ecosystems
→ Rendering ecosystems efficiently
→ Non-realistic rendering of plants

What to learn?

→ Modelling methodologies
→ Geometry generation and manipulation
→ Advanced rendering techniques
→ Some practical aspects

Modelling of Plants

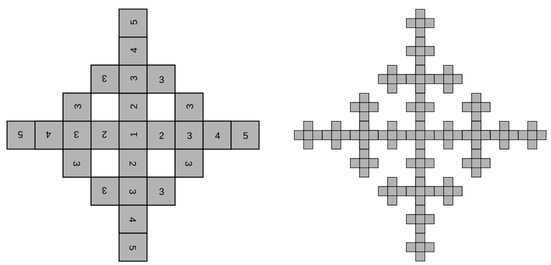
Modelling methodologies

Two different modelling methodologies:

1. **Procedural methods** (Algorithms)
   intuitive, simple parameters, special solutions for single plants
2. **Rule-based Systems** (formal grammars)
   local rules, abstract parameterization, general approach

The Beginning

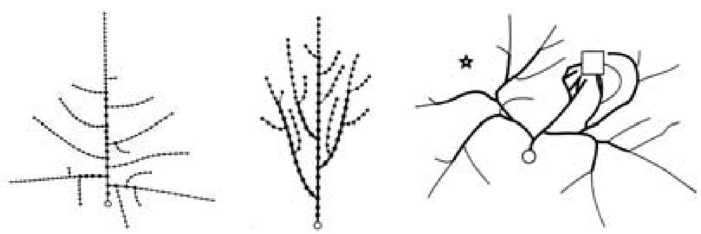
→ Ulam, 1966: cellular automatons used for branching structures
→ Cell states are changed due to given rules



Branching structure produced by Ulam

A Continous Model

→ Cohen, 1967 (Biologist): creation of a continous model
→ Models are programmed by Fortran programs
(one program for each structure)



Branching structures produced by Cohen

Model of Cohen:

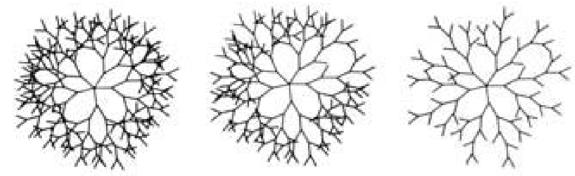
- growth takes place only at the end of branches
- strength and angle of growth is determined by current direction, a density field plus its gradient sa well as resistence against changes of angle.
- branching tendency is modulated by a probabilistic value, computed by distance to branching place and local density field.

Modelling Branching Regulation

→ Fisher and Honda (1979), two biologists, worked on branching e.g. how trees manage not two overlap their branches

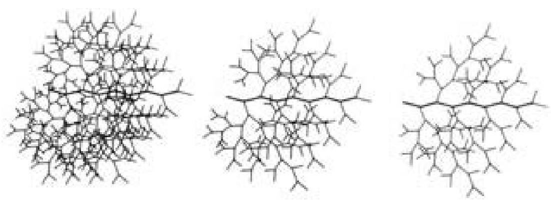
→ two-dimensinonal models, two methods of branching regulation:

1. branching is performed only at place whith a minimum distance to all other branching places.



Verzweigungsbildung nach Honda

2. during branching successors receive different growth rates in dependence to the local situation



Branching structures by Fisher and Honda

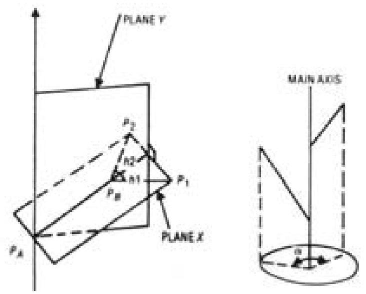
Modelling Issues

→ efficient modelling of branching structures

→ generation of a variety of tree models

Model of Aono and Kunii (1984):

1. binary branching, generated trees have a monopodial or sympodial shape
2. length and diameter of branches are reduced by a constant factor, branching angles are constant for all branching levels.
3. child branches are directed into plane defined by father branch and its maximal gradient.
4. simultanous branching at all end points of the branches



(a) (b)

Parameters in Aono und Kunii's model:

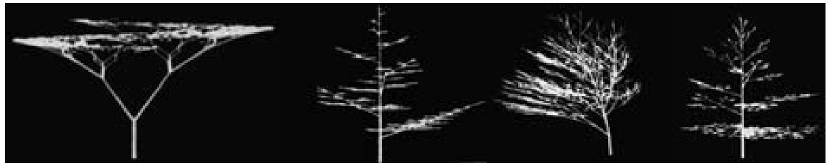
a) Parameterisation of branches; b) Divergence angle

→ later approaches mostly use elements of this branching model

Drawbacks of the implemented approach:

→ only very simple leaves

→ tree skeleton is only sketched by lines of different width



Branching structures by Aono and Kunii

Particle Systems

→ Reeves and Blau (1985) visual models needed for a movie

→ trees: simple recursive branching model

→ post processing: introduction of randomness

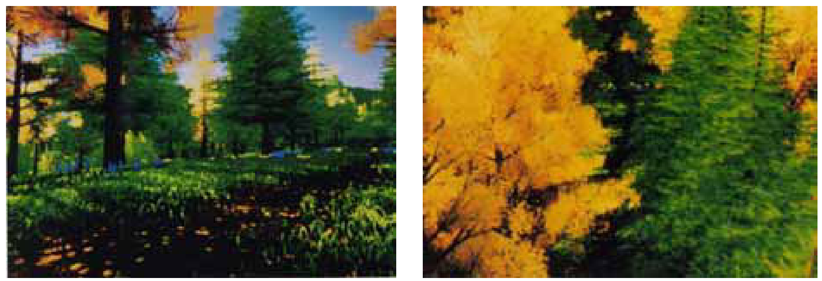
→ leaves: small balls with color and orientation

→ important for rendering: good colors and shadow

What to learn?

→ simple models can generate very nice images

Results:



Oriented particle systems by Reeves und Blau

A Fractal Tree Model

→ Oppenheimer (1986): inspired by Mandelbrot

→ recursive procedure creates self similar trees

→ Used parameters:

- → Branching angle
- → Ratio between size of father and child branches
- → Amount of taping along stem and branches
- → Number of branches per segment
- → Deviation angle


```
procedure fractaltree()
 begin
   draw current branch segment
   if (small enough)
       then drwa leaf
       else begin
           transform for current branch
           fractaltree()
           repeat
             begin
               transform for branching
               fractaltree()
             end
 end end
```


Result:

Fractal tree by Oppenheimer

Modelling the bark:

horizontal saw teeth function + Brownian noise

$$\text{bark}(x, y) = \text{sawteeth}(N * (x + R * \text{noise}(x, y)))$$

→ noise(x, y): periodical in $x-$ and $y-$ direction

Geometric Modelling

→ Bloomenthal (1985): tries to improve tree geometry

Modelling method:
→ control points by recursive algorithm
→ connection of points by Spline interpolation (C2)
→ generation of surface by circular shapes perpendicular to spline

→ main problem: natural looking branches.
→ solution: saddle surface at branches
→ bark: Bump-Mapping (source is real bark)

Result:

Tree geometries, created by Bloomthal

A Botanic Approach

→ de Reffye (1988): simulate biological growth in 3-d
→ discrete model (one step per growth period)
→ along a sprout segments are placed
→ each sprout can rest or die

A bud (top of a sproud) has three probability values:

1. probability to die
2. probability to rest
3. probability to branch

Growth simulation by de Reffye

Used Parameter:

- tree age
- growth speed of branches in different levels
- number of buds in each level
- probabilities for dying, resting, and branching

```
procedure budtree()
begin
    for each time step do begin
        for jede living bug do begin
            if bud does not die and does not rest
            then begin
                generate segment
                generate apical leaf
            end
            for each bud do
                if bud branches
                then generate branch
        end
    end
end
```


Results:

Branching structures by de Reffye

A Morphologic Idea

→ Leonardo da Vinci: tree skeleton is a combination of strands
each strand connects a leaf to a small root

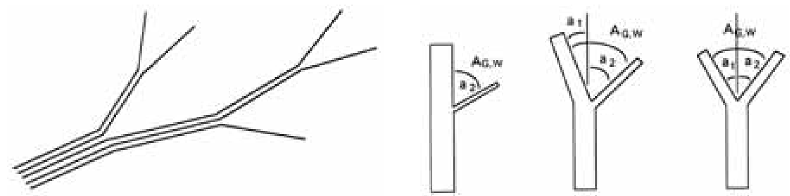
→ at branching: strands are divided and generate children

→ geometric result: section area of father equals sum of section areas of children

→ Holton (1994): modelling method based on strands

→ Number of strands determines thickness and length of branches, number of leaves and branching angle

Strand model:



Strand model by Holton

Results:

Example trees by Holton

Another procedural Method

→ Weber and Penn (1995): tree generator using 50 parameters

→ very nice trees, model very complicated

Trees created by Weber and Penn

Voxel Plants

→ Greene (1989): interaction between plants and environment

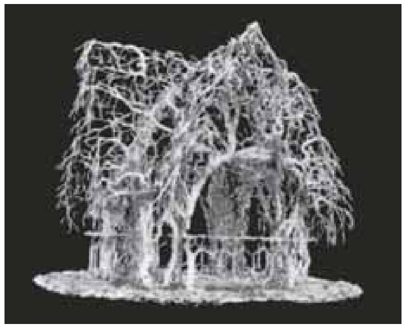
→ in this case: plants that grow on walls etc.

→ for each voxel element: determine how much sunlight is introduced

→ starting from a manually given seedpoint a probabilistic algorithm lets the plant grow

→ the algorithm searches the next voxel that recieves enough light and directs the plant into that direction

Results:



Example plants by Greene

Results (cont.):

Example plant by Greene

Rule-based Modelling of Plants

→ a formal rule base transforms a given initial state into an final state

→ extremely compact data description for complex objects (data amplification)

→ data amplification is a time-consuming process

→ creation is mostly based on local generation rules

Data Amplification

→ replacement rules modify local portions of the data description

Examples:

- **Graphs**
  Edges or nodes are replaced by given sub graphs
- **Array**
  Field values or combinations of values are replaced by other values
- **Strings**
  Characters in a string are replaced by strings
- **Geometric Objects**
  Geometric primitives are replaced by more complex ones

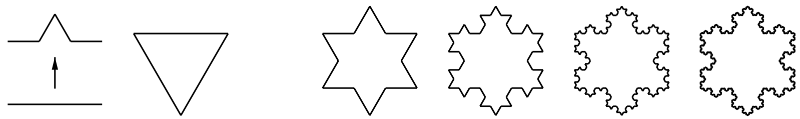
Geometric Replacement

The von Koch curve, a classical example:

→ lines are replaced by a set of lines (generator)

→ initial state is a triangle (initiator)

→ successive replication of lines generates a complex figure



Lindenmayer-Systemes

→ a Lindenmayer-System or L-System uses string replacement

→ it is defined by a formal grammar $G = (V, \omega, P)$:

V: alphabet

ω: axiom, a non-empty string

P: set of productions

→ per time step all possible replacements are performed in parallel

→ main difference to Chomsky grammars

An Example

Alphabet: $V = \{f, F, +, -\}$

Axiom: $w = F - -F - -F$

Rule base: $P = \{F ::= F + F - -F + F\}$

Amplification (two replacements):

$F - -F - -F$

$F + F - -F + F - -F + F - -F + F - -F + F - -F + F$

$F + F - -F + F + F + F - -F + F - -F + F - -F + F + F + F - -F + F -$
$-F + F - -F + F + F + F - -F + F - -F + F - -F + F + F + F - -F +$
$F - -F + F - -F + F + F + F - -F + F$

Graphical Interpretation

→ string is converted into a graphical description

→ Turtle-metaphor (Przemislaw Prusinkiewicz)

→ Turtle is moved over a drawing plane or within the 3-d space

→ state of turtle: vector (x, y, α) position plus angle

→ commands of the string (characters) change these local parameters

Drawing commands in 2-d

F move turtle in current direction α with distance d and draw line state: $(x, y, \alpha) \rightarrow (x + d\cos\delta, y + d\sin\delta, \alpha)$

f move turtle in current direction α with distance d without drawing state: $(x, y, \alpha) \rightarrow (x + d\cos\delta, y + d\sin\delta, \alpha)$

$+$ increase angle about δ: $(x, y, \alpha) \rightarrow (x, y, \alpha + \delta)$

\- decrease angle about δ: $(x, y, \alpha) \rightarrow (x, y, \alpha + \delta)$

Branching Structures

→ at a branch one needs to store state of the turtle

→ extension of L-System description: a state machine

Pushdown-Automaton:

→ a stack data structure stores the turtle states

→ access to stack: push and pop

corresponding characters:

[store current state (x, y, α) on stack

] load current state (x, y, α) from stack

Examples

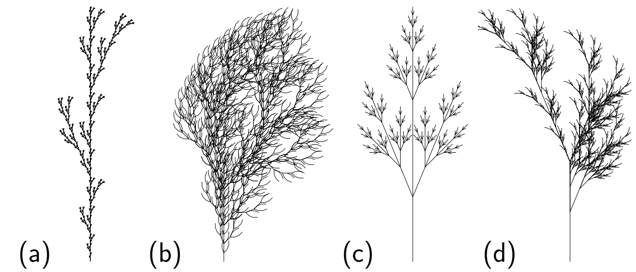


Figure	n	δ	w	P
(a)	5	25,7°	F	{ F ::= F[+F]F[-F]F }
(b)	4	22,5°	F	{ F ::= FF-[-F+F+F]+[+F-F-F] }
(c)	7	25,7°	X	{ X ::= F[+X][-X]FX, F ::= FF }
(d)	5	22,5°	X	{ X ::= F[[X]+X]+F[+FX]-X, F ::= FF }

Drawing commands in 3-d

→ turtle: now position in 3-d plus three rotation angles (α, β, γ)

→ differential angle δ may change each of the three angles

→ turtle state: $(x, y, z, \mathbf{M})$ with $\mathbf{M}$ rotation matrix

drawing commands:

- F move turtle in current direction α with distance d draw line, change state
- f move turtle in current direction α with distance d without drawing, change state
- $+$ increase current angle γ by δ, change state
- \- increase current angle γ by δ, change state

- & increase current angle β by δ, change state
- $\wedge$ decrease current angle β by δ, change state
- $\backslash$ increase current angle α by δ, change state
- / decrease current angle α by δ, change state
- | turn around

→ additional commands for colors, width of branches

→ filled geometry: { and } define start and end of path that is triangulated later

Example

$A ::= [\&FL!A]/////'[\&FL!A]///////'[\&FL!A]$
$F ::= S/////F$
$S ::= FL$
$L ::= ['''\wedge\wedge\{-f+f+f-|-f+f+f\}]$

Bush, defined by Prusinkiewicz and Lindenmayer

Stochastic and Parametric Systems

Stochastic L-Systems:

→ introduce randomness into plant generation

→ for a character more than one replacement rule is defined

→ for each rule an application probability is defined

Parametric L-Systems:

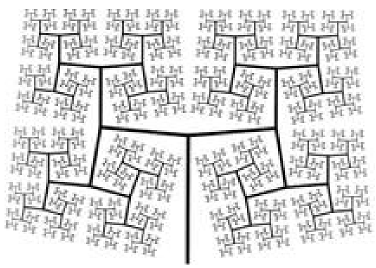
→ parameterization of commands:
$F(\mathbf{w})$: move into current direction about $\mathbf{w}$

→ allows to change values during expansion

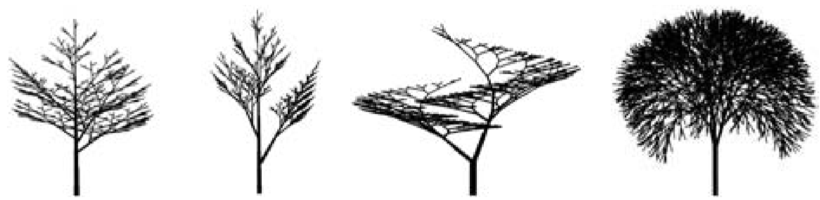
Example

Binary tree:

let be $n = 10$, $\delta = 85°$, $R = 1.456$, axiom $w = A(1))$
only one rule: $A(s) ::= F(s)[+A(s/R)][-A(s/R)]$



More Examples



Plants from Prusinkiewicz, Lindenmayer: The algorithmic beauty of plants

Modelling Phyllotaxis using L-Systems

Production: $A(n) ::= +(137.5)[f(\sqrt{n}) \sim D]A(n+1)$

→ D: sub-system to define a small circle

→ complete head of a sun flower is defined by sub L-systems for seeds S, and different kinds of blossoms R R, M, N, O, P

→ conditional L-System

$$
\begin{array}{ll}
A(n) ::= +(137.5)[f(\sqrt{n})C]A(n+1) & \\
C(n) : n \leq 440 & ::= \sim S \\
C(n) : 440 < n \leq 565 & ::= \sim R \\
C(n) : 565 < n \leq 580 & ::= \sim M \\
C(n) : 580 < n \leq 595 & ::= \sim N \\
C(n) : 595 < n \leq 610 & ::= \sim O \\
C(n) : 610 < n & ::= \sim P
\end{array}
$$

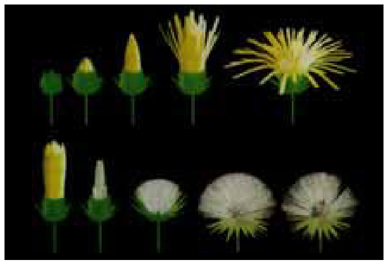
Results:

Plants from Prusinkiewicz, Lindenmayer: The algorithmic beauty of plants

Animation with L-Systems

→ differential L-Systems

→ continous growth is described by differential equations



Animation of plant growth, courtesy of P. Prusinkiewicz

Plant Interaction with Environment

→ environmental sensitive L-Systems

Two plant interactions, courtesy of P. Prusinkiewicz

Combined procedural and rule-based modelling

Plant Modelling

so far:

→ procedural plant modelling methods

- → algorithms that can be parameterised
- → more or less specific
- → often intuitive parameters (e.g. age, vigour)
- → example: AMAP/Bionatics

→ rule-based plant modelling methods

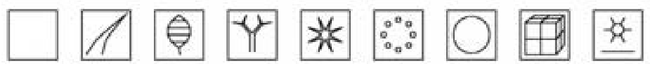
- → L-Systems, iterated function systems (plant images)
- → general modelling scheme
- → abstract formal representation for a plant

xfrog - A New Modelling Paradigm

→ collaboration with Bernd Lintermann (ZKM Karlsruhe)

→ plant is represented by a structure graph

Nodes: components that encapsulate data and algorithms



Component types:

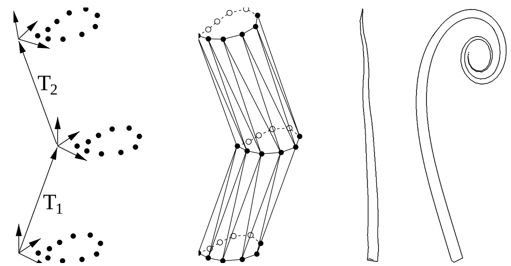
1. generation of geometry
2. multiplication of geometry
3. global modelling

Links: generation rules

Generation of Geometry

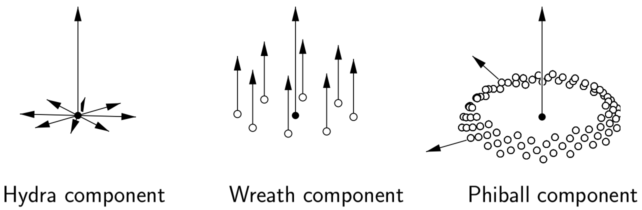
→ leaves: textured surfaces (triangles, quads or triangulated surfaces)

→ branches: generalized cylinders (inspired by Todd and Latham):



Multiplication of Geometry

→ three types of multiplication:



→ problem: plant structure is represented by graph structure and by multiplication components

⇒ geometry generation in two stages

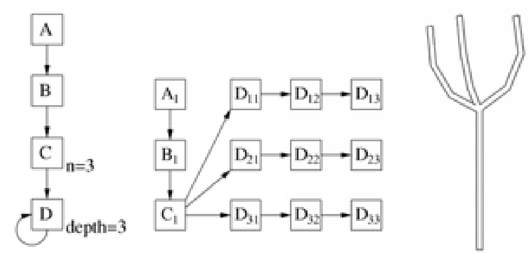
Multiplication of Geometry (cont.)

A root component

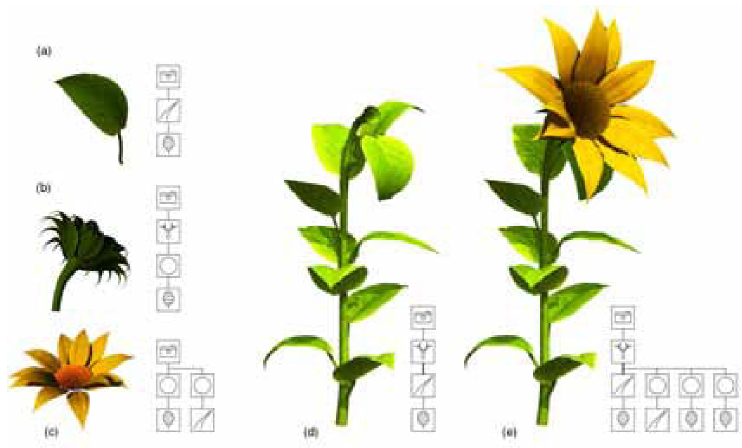
B component that generates a cylinder

C multiplication component, number of copies: 3

D component that generates a cylinder, recursion depth: 3



Plant Example I



Plant Example II

Plant Example III

Further Examples I

Further Examples II

Further Examples III

Video

Breaking of regularity

→ problem: so far generated plants are too regular
→ and modelling depends on local rules only

Solutions:
→ introduction of randomness
→ modelling of exceptions
→ global modelling (FFDs)

Functional modelling

→ in multiplication components, many parameters are stored as ranges.
→ for each multiplied component, an indivdual value is determined by interpolation
→ this mechanism can be modified by applying functions
→ part of these functions can be iteration numbers (affects bending, see right)

Exceptions

→ similar mechanism to functional modelling
→ each multiplied component receives its own identification number
→ in each component it can be stored if its geometry is generated according to the identification number
→ application: dead branches and twigs

Freeform-deformation

→ a special component allows to deform the growing space
→ this can be done for all generated vertices or only for the tree skeleton

center: needles too large, right: better result (only skelton deformed)

Tropisms

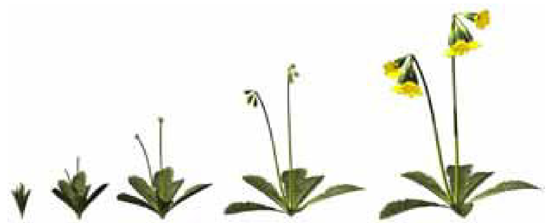
→ allow to direct growth into a desired direction
→ linear, circular or freely given directions can be used

Pruning

→ growth is restricted to a given volume
→ can be implemented like cutting the branches or as a tendency to stop growth

Animation

→ the set of parameters describes the model completely
→ different sets of parameters can describe a plant for different ages (keyframes)
→ interpolation of the parameter values allows to animate plant growth



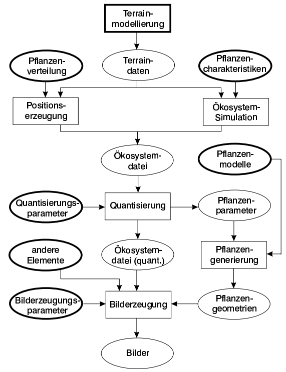
Modelling ecosystems

General outline

→ cooperation: Stanford University, University of Calgary

Layout of the project:

- parametrized plant models
- specification of terrain and ground paramters
- specification of plant positions and parameters
- reduction of generated geometry
- efficient rendering



Video

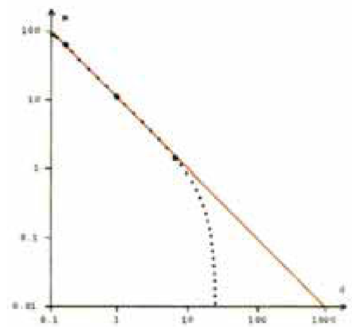
Specification by Simulation

→ random initial positions

→ simulation of self thinning
Yoda et al.:
$\log(m) = -\frac{3}{2}\log(d) + const$
m: dry weight of plants
d: plant density

→ population below curve
⇒ number increases

→ growth (increase of weight)
⇒ selection



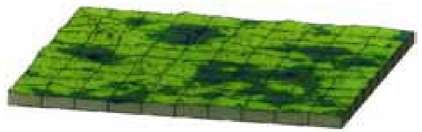
Video

Simulating Different Plant Species

Complex model:

→ self thinning, domination, continous seeding

→ environmental factors (water, soil)



Ground Water

Plant Distribution

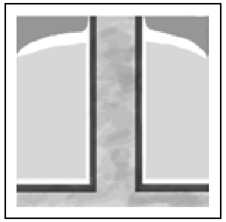
Resulting image (Premyslaw Prusinkiewicz):

Graphical Specification

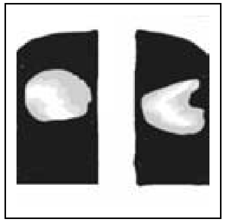
- user draws populations (density and other parameters)
- plant positions computed by variant of halftoning
- each plants recieves individual parameters

Example

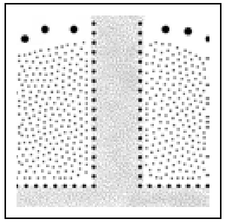
→ gray-scale images drawn by the user



Density



Prosperity



Positions

Reduction of Geometry

→ variant of traditional instancing
(replace repeated objects by instances of one representative)

→ here: replace similar objects by a set of representatives
⇒ **Approximate Instancing**

→ 10-15 representatives per species are enough for cheating a large visual complexity.

Video

Examples I

Examples II

Examples III

Image: Bernd Lintermann

Examples IV

Examples V

Image: Bernd Lintermann

Real-time rendering of virtual landscapes

joint work with:

Carsten Coldiz, Dresden Technical University

Marc Stamminger, University of Weimar (Germany)

George Drettakis, Reves/INRIA, Sophia Antipolis (France)
www-sop.inria.fr/reves/index.gb.html

→ will be published in IEEE Visualization 2002 (Boston)

Problems With Complex Plant Scenes

- unnecessary amount of variety
  - → approximate instancing
    - → reduces the amount of different models needed in a scene
- over-specification
  - → simple primitives like points and lines
    - → reduces the amount of vertices to be processed
  - → visibility culling/occlusion culling
    - → unseen models are not transferred to the graphics processor
- too complex plant models
  - → level-of-detail algorithms
    - → each plant is shown with a complexity that is sufficient for its actual size

An Example

→ 16,7 million polygons

→ about one million pixels

⇒ bunches of polygons per pixel

LOD-Representation for Plants

→ static level-of-detail:

- → model is represented by several different representation that are blended
- → visible popping artefacts can occur
- → all representations must be stored

→ dynamic level-of-detail:

- → model representation is computed individually for each size on the screen
- → popping artefacts are reduced
- → memory efficient
- → representation must be computed for each image

Representation by Points and Lines

→ compact objects (leaves, petals) ⇒ point representation

→ thin objects (branches, special leaves) ⇒ line representation

→ point and line representations are shuffled

→ ... and stored in vertex arrays

⇒ variing amount of data can be used to represent a plant

⇒ blending between polygonal model and approximation is done successively on the basis of individual leaves

A Point Representation

Polygones 13,000 Points 6,500 3,250 1,625

Single Plant Example

→ small triangles: vertex rate counts

→ standard way of approximation: replace polygons by points if projected area can be displayed more efficiently by point approximation

→ display can be tuned by point splat size and replacement threshold

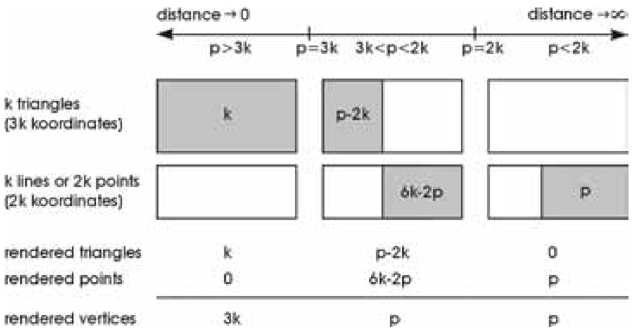
standard display

Single Plant Example (cont.)

threshold reduced: points visible

threshold reduced, splat size increased

Reduction principle



Line Representation

1,500 plants, original model size 12 million triangles, now displayed with 10-15 Hz

Video

Importance Reduction

→ important parts of plants (blossoms, petals) are not so much reduced

⇒ quality can be increased without much more data

Complex Scene Examples

100 million polygones/8 Hz

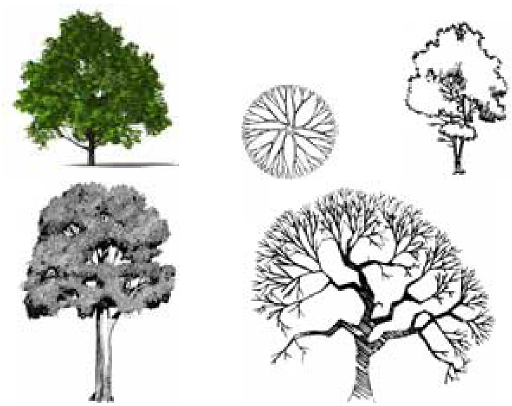
70 million polygones/4 Hz

→ here: scene hierarchy used to enhance LOD representation

Video

Nonrealistic Rendering of Plants

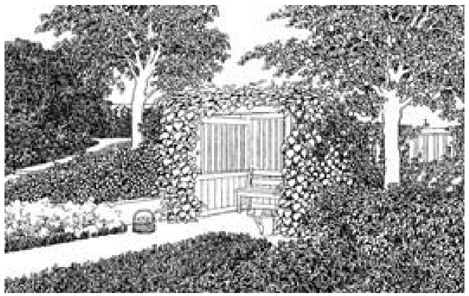
Images of Trees



Drawings: Larry Evans

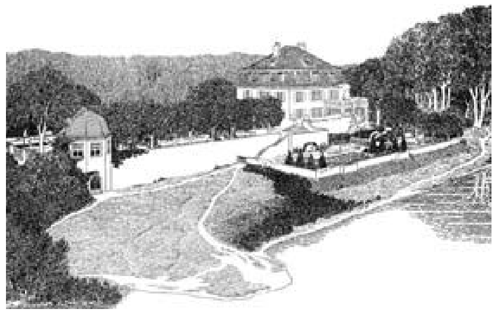
Example I

A garden, drawn at beginning of 20th centruy:



Drawing: Larry Evans

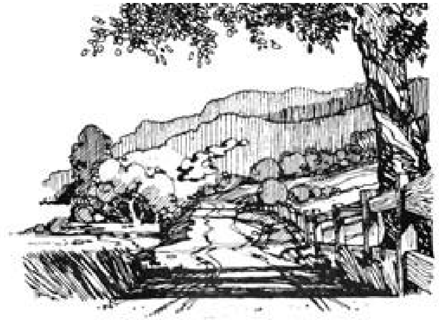
Example II



Drawing: Larry Evans

Modern illustrations:

Drawing: Larry Evans

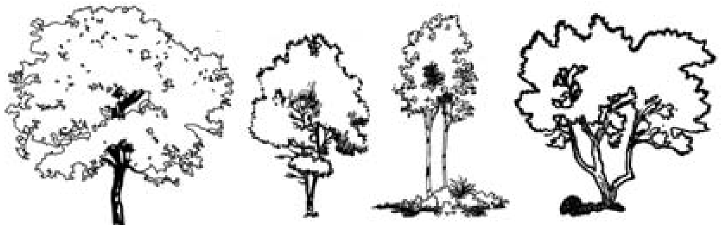


Drawing: Larry Evans

Synthetic Plant Sketches

→ How do artists work?

→ first observation: abstraction of shape

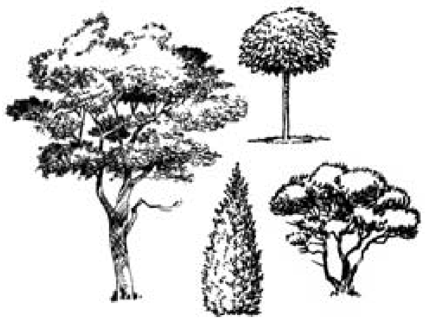


Drawing: Larry Evans

→ shape by agglomeration of small entities

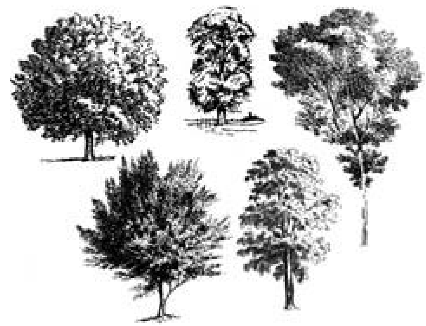
Drawing: Larry Evans

→ light and shadow by adding detail



Drawing: Larry Evans

→ light and shadow by hatching



Drawing: Larry Evans

Synthetic Illustrations

→ starting from realistic plant models (xfrog)
→ tree skeleton and leaves must be handled differently

Data representation:

- tree skeleton: thinned geometry
- leaves: oriented point cloud

→ this enables flexible, fast and coherent illustration

Algorithm

Tree skeleton:
conventional algorithms of NPR (silhouettes + hatching)

Leaves:
→ abstract drawing primitives represent foliage
→ depth differences control detail of drawing
→ drawing styles by variaton of abstract drawing primitives

Example

→ a large tree

Example (cont.)

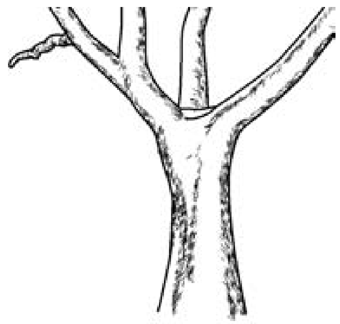
→ representation of tree skeleton

Example (cont.)

→ representation of foliage (point cloud)

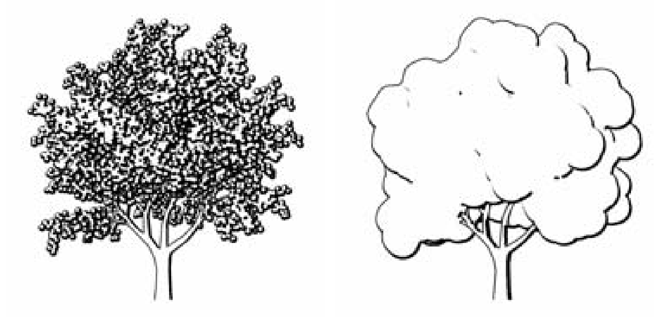
Example (cont.)

→ hatching of stem



Example (cont.)

→ two representations of the same tree

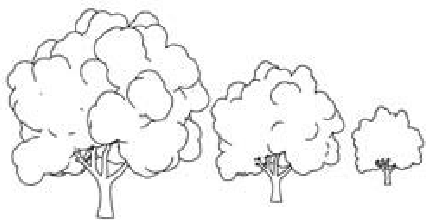


(size=0.15, DDT=1000) (size=0.7, DDT=2000)

Video

Example (cont.)

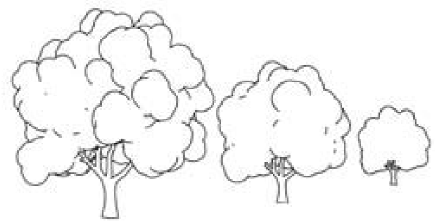
→ non-linearity of depth buffer
⇒ LOD can be achieved automatically



→ in the background differences are relatively larger

Example (cont.)

→ additionally: enlarge primitives in the background

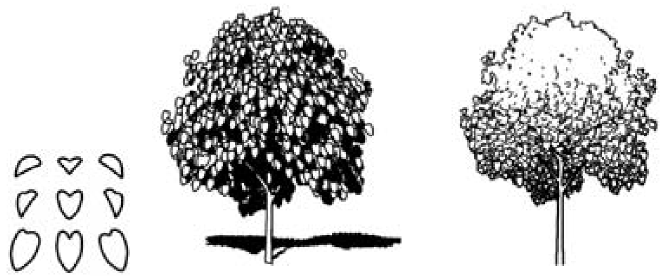


Video

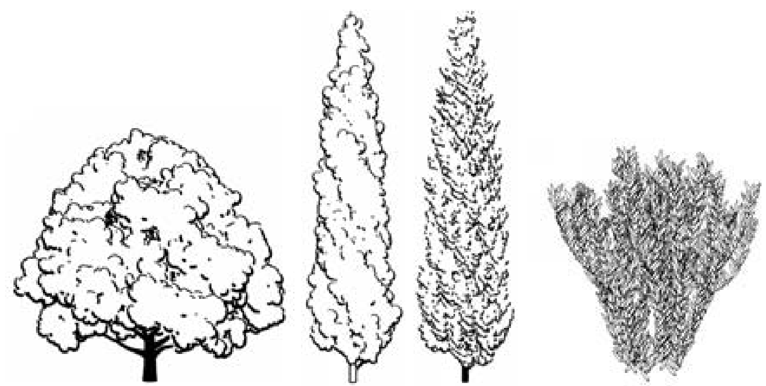
Example (cont.)

→ variation of primitive shape

→ performed by interpolation due to normal vector



Results



A larger example

Video

Further Readings

→ B. Lintermann, O. Deussen:
Interactive Modelling of Plants
IEEE Computer Graphics and Applications, 19(1), pp. 56–65, 1999

→ O. Deussen, P. Hanrahan, M. Pharr, B. Lintermann, R. Měch, P. Prusinkiewicz:
Realistic Modelling and Rendering of Plant Ecosystems
SIGGRAPH 98 Conference Proceedings, pp. 275–286

→ O. Deussen, C. Colditz, M. Stamminger, G. Drettakis:
Interactive Visualization of Complex Plant Ecosystems
IEEE Visualization 2002, Boston, in print,
see: www.inf.tu-dresden.de/cgm

→ O. Deussen: **Computergenerierte Pflanzen**
Springer-Verlag 2002 (in print)

References

[1] H. Abelson and A. diSessa. *Turtle geometry*. MIT-Press, Cambridge, 1982.

[2] I. Adler. A model of contact pressure in phyllotaxis. *Journal of Theoretical Biology*, 45:1–79, 1974.

[3] M. Aono and T. L. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, May 1984.

[4] R. A. Armstrong. A comparison of index-based and pixel-based neigh-borhood simulations of forest growth. *Ecology*, 74(6):1707–1712, 1993.

[5] G. Baranoski and J. Rokne. An algorithmic reflectance and transmittance model for plant tissue. In *EUROGRAPHICS '97, Computer Graphics Forum*, 1997.

[6] M. Barnsley. *Fractals everywhere*. Academic Press, 1988.

[7] M. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. In *Proceedings of the Royal Society*, Atlanta, Georgia, 1984. School of mathemetics, Georgia Institut of Technology.

[8] M. F. Barnsley, A. Jacquin, F. Malassenet, L. Reuter, and A. D. Sloan. Harnessing chaos for image synthesis. In J. Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 131–140, August 1988.

[9] J. Bloomenthal. A representation for botanical trees using density distributions. In *Intl. Conf. on Engineering and Computer Graphics*, pages 571–575. Beijing, China, August 1984.

[10] J. Bloomenthal. Modeling the mighty maple. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 305–311, July 1985.

[11] Bruce M. Blumberg and Tinsley A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, pages 47–54. ACM SIGGRAPH, Addison Wesley, August 1995.

[12] D. Brown and P. Rothery. *Models in Biology: mathematics, statistics and computing*. John Wiley, 1993.

[13] A. Brownbill. Reducing the storage required to render l-system based models. Master's thesis, University of Calgary,, 1996.

[14] N. Chiba, K. Muraoka, A. Doi, and J. Hosokawa. Rendering of forest scenery using 3d textures. *The Journal of Visualization and Computer Animation*, 8:191–199, 1997.

[15] N. Chomsky. Three models for the desciption of languages. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

[16] V. Claus, H. Ehrig, and G. Rozenberg. *Graph grammars and their application to computer science; First international workshop*. LNCS 73. Springer-Verlag, 1979.

[17] D. Cohen. Computer simulation of biological pattern generation processes. *Nature*, (216):246–248, October 1967.

[18] L. da Vinci. Notizbücher (Bemerkung Nr. 394).

[19] P. de Reffye, C. Edelin, J. Francon, M. Jaeger, and C. Puech. Plant models faithful to botanical structure and development. In J. Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 151–158. ACM SIGGRAPH, August 1988.

[20] S. Demko, L. Hodges, and B. Naylor. Construction of fractal objects with iterated function systems. *Computer Graphics (Proceedings of SIGGRAPH 85)*, 19(3):271–278, July 1985. Held in San Francisco, California.

[21] O. Deussen. Pixel-oriented rendering of line drawings. In T. Strothotte, editor, *Computational Visualization: Graphics, Abstraction and Interactivity*, pages 105–120. Springer Verlag, 1998.

[22] O. Deussen, S. Hiller, K. van Overveld, and T. Strothotte. Floating points: A method for computing stipple drawings. *Computer Graphics Forum*, 19(4):40–51, Eurographics 2000 Conference Proceedings.

[23] O. Deussen and T. Strothotte. Computer-generated pen-and-ink illustration of trees. *Computer Graphics*, 34(4):13–18, SIGGRAPH 2000 Conference Proceedings.

[24] D. Ebert, K. Musgrave, P. Peachey, K. Perlin, and S. Worley. *Texturing and Modeling: A Procedural Approach*. AP Professional, 1994.

[25] L. Evans. *The New Complete Illustration Guide: The Ultimate Trace File for Architects, Designers, Artists, and Students*. Van Nostrand Reinhold Company, 1996.

[26] F. G. Firbank and A. R. Watkinson. A model of interference within plant monocultures. *Journal of Theoretical Biology,*, 116:291–311, 1985.

[27] J.B. Fisher and H. Honda. Computer simulation of branching pattern and geometry in terminalia (combretaceae), a tropical tree. *Botanical Gazette*, 138:377–384, 1977.

[28] J.B. Fisher and H. Honda. Branch geometry and effective leaf area: a study of of terminalia-branching pattern. 1 theoretical ideas. *American Journal of Botany*, 66:633–644, 1979.

[29] D. R. Fowler, J. Hanan, and P. Prusinkiewicz. Modelling spiral phyllotaxis. *Computers and Graphics*, 13(3):291–296, 1989.

[30] D. R. Fowler, P. Prusinkiewicz, and J. Battjes. A collision-based model of spiral phyllotaxis. In E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 361–368. ACM SIGGRAPH, July 1992.

[31] M. Gardner. Mathematical games: On cellular automata, self-reproduction, ther garden of eden and the game "life". *Scientific American*, 224(2):112–117, 1971.

[32] M. Gervautz and C. Traxler. Representation and realistic rendering of natural phenomena with cyclic csg-graphs. *The Visual Computer*, 12:62–74, 1996.

[33] D. G. Green. Modelling plants in landscapes. In M. T. Michalewicz, editor, *Plants to ecosystems. Advances in computational life sciences I*, pages 85–86, Melbourne, 1997. CSIRO Publishing.

[34] N. Green. Voxel space automata: Modelling with stochastic growth processes in voxel space. *Computer Graphics*, 23(3):175–184, 1989.

[35] J.W. Haefner. *Modelling Biological Systems*. Kluwer Academic Publishers, 1996.

[36] F. Hallé, R.A.A. Oldeman, and P.B. Tomlinson. *Tropical Trees and Forests*. Springer-Verlag, 1978.

[37] S. I. Higgins and D. M. Richardson. A review of models of alien plant spread. *Ecological Modelling*, 87:249–265, 1996.

[38] M. Holton. Strands, gravity and botanical tree imagery. *Computer Graphics Forum*, 13(1):57–67, 1994.

[39] H. Honda. Description of the form of trees by the parameters of a tree-like body: effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theroetical Biology*, 31:331–338, 1971.

[40] H. Honda, P.B. Tomlinson, and J.B. Fisher. Computer simulation of branching interaction and regulation by unequal flow rates in botanical trees. *American Journal of Botany*, 68(4):569–585, 1981.

[41] B. Hosgood, S. Jacquemoud, G. Andreoli, J. Verdebout, G. Pedrini, and G. Schmuck. Leaf optical properties experiment 93. Technical report, Joint Research Center, European Comission, Institute for remote sensing Applications, EUR 16095 EN, 1995.

[42] D. H. House, G. S. Schmidt, S. A. Arvin, and M. Kitagawa-DeLeon. Visualizing a real forest. *IEEE Computer Graphics and Applications*, 18(1):12–15, 1998.

[43] S.E Jorgensen and G. Bendoricchio. *Fundamentals of Ecological Modelling*. Elsevier Science Publishers, 2001.

[44] Timothy L. Kay and James T. Kajiya. Ray tracing complex scenes. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 269–278, August 1986.

[45] M. Kowalski, L. Markosian, J. D. Northrup, L. Burdev, R. Barzel, L. Holden, and J. F. Hughes. Art-based rendering of fur, grass, and trees. In *SIGGRAPH '99 Conference Proceedings*. ACM SIGGRAPH, August 1999.

[46] T. Ligget. *Stochastic interacting systems: Contact, Voter and Exclusion Processes*. Springer-Verlag, New-York, 1999.

[47] A. Lindenmayer. Mathematical models for cellular interactions in development, i. filaments with one-sided inputs. *Journal of Theoretical Biology*, 18:280–299, 1968.

[48] A. Lindenmayer. Mathematical models for cellular interactions in development, ii. simple and branching filaments with two-sided inputs. *Journal of Theoretical Biology*, 18:300–315, 1968.

[49] A. Lindenmayer. Developmental systems without cellular interaction, their languages and grammers. *Journal of Theoretical Biology*, (30):455–484, 1971.

[50] B. Lintermann and O. Deussen. A modelling method and user interface for creating plants. *Computer Graphics Forum*, 17(1):73–82, 1998.

[51] Lucasfilm Ltd. The adventures of andré and wally b. Film, 1984.

[52] B. Mandelbrot. *Fractals: Form, Chance and Dimension*. W.H. Freeman and Co., San Fransisco, 1977.

[53] B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Co., San Fransisco, 1983.

[54] D. Marshall, D. S. Fussel, and A. T. Campbell. Multiresolution rendering of complex botanical scenes. In *Proceedings of Graphics Interface 97*, pages 97–104, May 1997.

[55] N. Max. Hierarchical rendering of trees from precomputed multi-layer Z-buffers. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 165–174. Eurographics, Springer-Verlag, Vienna, June 1996.

[56] N. Max, O. Deussen, and B. Keating. Hierarchical image-based rendering using texture mapping hardware. *Eurographics Rendering Workshop 1999*, pages 57–62, June 1999.

[57] N. Max, C. Mobley, B. Keating, and E. Wu. Plane-parallel radiance transport for global illumination in vegetation. *Eurographics Rendering Workshop 1997*, pages 239–250.

[58] N. Max and K. Ohsaki. Rendering trees from precomputed Z-buffer views. In *Eurographics Rendering Workshop 1995*. Eurographics, June 1995.

[59] A. Meyer and F. Neyret. Interactive volumetric textures. In *Eurographics Rendering Workshop 1998*, pages 157–168, New York City, NY, June 1998. Eurographics, Springer Wien.

[60] F. Kenton Musgrave, Craig E. Kolb, and Robert S. Mace. The synthesis and rendering of eroded fractal terrains. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 41–50, July 1989.

[61] F.K. Musgrave. *Methods for Realistic Landscape Imaging*. PhD thesis, Yale University, 1993.

[62] R. Měch and P. Prusinkiewicz. Visual models of plants interacting with their environment. In H. Rushmeier, editor, *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 397–410. ACM SIGGRAPH, ACM Press, 1996.

[63] J. von Neumann. Theory of self-reproducing automata. Univeristy of Illinois Press, Urbana, 1966.

[64] F. Neyret. Synthesizing verdant landscapes using volumetric textures. In Xavier Pueyo and Peter Schröder, editors, *Eurographics Rendering Workshop 1996*, pages 215–224, New York City, NY, June 1996. Eurographics, Springer Wien.

[65] F. Neyret. Modeling, animating, and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):55–70, January-March 1998.

[66] P. E. Oppenheimer. Real time design and animation of fractal plants and trees. In D.C. Evans and R.J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 55–64. ACM SIGGRAPH, August 1986.

[67] F. Perbert and M. Cani. Animating prairies in real-time. In *2001 ACM Symposion on interacive 3D Graphics*, pages 103–110, March 2001.

[68] Ken Perlin. An image synthesizer. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 287–296, July 1985.

[69] Matt Pharr, Craig Kolb, Reid Gershbein, and Pat Hanrahan. Rendering complex scenes with memory-coherent ray tracing. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*. ACM SIGGRAPH, Addison Wesley, 1997.

[70] E. Pielou. *Mathematical Ecology*. John Wiley, 1977.

[71] P. Prusinkiewicz. Modelling and visualization of biological structures. In *Proceedings of Graphics Interface '93*, pages 128–137, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.

[72] P. Prusinkiewicz and M. S. Hammel. A fractal model of mountains with rivers. *Proc. Graphics Interface '93*, pages 174–180, 1993.

[73] P. Prusinkiewicz, M. S. Hammel, and E. Mjolsness. Animation of plant development. *Computers Graphics (SIGGRAPH '93 Proceedings)*, pages 351–360, 1993.

[74] P. Prusinkiewicz, P. James, and R. Měch. Synthetic topiary. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 351–358, 1995.

[75] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.

[76] Przemyslaw Prusinkiewicz. Graphical applications of l-systems. *Graphics Interface '86*, pages 247–253, May 1986.

[77] Przemyslaw Prusinkiewicz and Glen Sandness. Koch curves as attractors and repellers. *IEEE Computer Graphics & Applications*, 8(6):26–40, November 1988.

[78] W. T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2):91–108, April 1983. Held in USA.

[79] W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 313–322, July 1985.

[80] R. E. Ricklefs. *Ecology*. W. H. Freeman, New York, 1990.

[81] J. Rossignac and P. Borrel. Multi-resolution 3D approximations for rendering complex scenes. In B. Falcidieno and T.L. Kunii, editors, *Geometric Modeling in Computer Graphics*, pages 455–465. Springer Verlag, Genova, Italy, 1993.

[82] S. Rubin and T. Whitted. A three-dimensional representation for fast rendering of complex scenes. In *Computer Graphics (SIGGRAPH '80 Proceedings)*, volume 14, pages 110–116, 1985.

[83] A. Salomaa. *Formal Languages*. Academic Press, 1990.

[84] D. Saupe. Point evaluation of multi-variable random fractals. In H. Jürgens and D. Saupe, editors, *Visualisierung in Mathematik und Naturwissenschaften*, pages 114–126. Springer-Verlag, 1989.

[85] J.W. Shade, D. Lischinski, D. Salesin, T. DeRose, and J. Snyder. Hierarchical image caching for accelerated walkthroughs of complex environments. In *Proceedings of SIGGRAPH 96*, pages 75–82, August 1996.

[86] A. R. Smith. Plants, fractals and formal languages. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18(3):1–10, July 1984.

[87] B. Smits, J. Arvo, and D. Greenberg. A clustering algorithm for radiosity in complex environments. *Computer Graphics (SIGGRAPH '94 Proceedings)*, 28(3):435–442, 1994.

[88] John M. Snyder and Alan H. Barr. Ray tracing complex models containing surface tessellations. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 119–128, July 1987.

[89] C. Soler and F. Sillion. Hierarchical instantiation for radiosity. *Eurographics Rendering Workshop 2000*, pages 173–184.

[90] M. Stamminger and G. Drettakis. Interactive sampling and rendering for complex and procedural geometry. In S. Gortler and C. Myszkowski, editors, *Rendering Techniques 2001*, pages 151–162. Eurographics, Springer-Verlag, Vienna, 2001.

[91] J. Stewart. Hierarchical visibility in terrains. In *Eurographics Rendering Workshop*, pages 217–228. Springer Verlag (Buchtitel: Rendering Techniques '97), 1997.

[92] A. Takhtajan. *Florostic regions of the world*. Berkeley, 1986.

[93] S. Todd and W. Latham. *Evolutionary Art and Computers*. Academic Press, London, 1992.

[94] S. Ulam. Pattern of growth of figures: Mathematical aspects. In G. Keps, editor, *Module, Proportion, Symmetry, Rhythm*, pages 64–74. Braziller, New York, 1966.

[95] H. Vogel. A better way to construct the sunflower head. *Mathematical Biosciences*, 44:179–189, 1979.

[96] R. Voss. Fractals in nature: from characterization to simulation. In H.O. Peitgen and D. Saupe, editors, *The science of fractal images*, pages 21–70. Springer-Verlag, 1988.

[97] J. Weber and J. Penn. Creation and rendering of realistic trees. In R. Cook, editor, *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 119–128. ACM SIGGRAPH, August 1995.

[98] K. Yoda, T. Kira, H. Ogawa, and K. Hozum. *J. Biol. Osaka Cy Univ*, 1963.