The 3D construction set for plant modelling
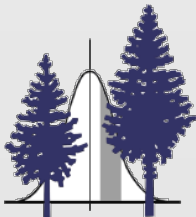
integrated in XL
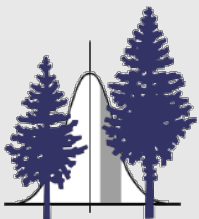
Michael Henke

Göttingen, den July 13, 2007

mhenke@uni-goettingen.de

http://fbi.forst.uni-goettingen.de/

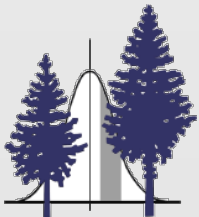Institut für Forstliche Biometrie und Informatik

Concept

- rule-based object generation

- created form Deussen and Lintermann, realised in the commercial software xFrog

- combination of two modelling concepts:

  1. procedural modelling
     - "objects will be programmed" - parameteriseable algorithm
       e.g. particle systems, trees of strands, fractal models
  2. rule based modelling
     - "objects will be result" - reapplication of rules to an initial state
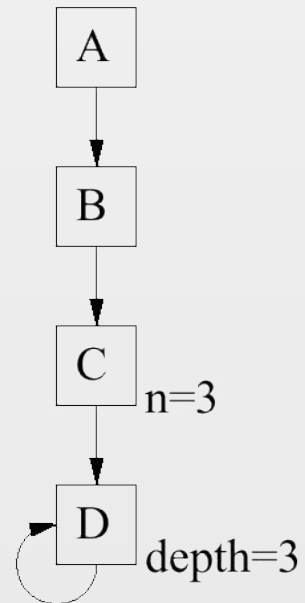   e.g. L-systems, CSG-Modelling, iterate function systems
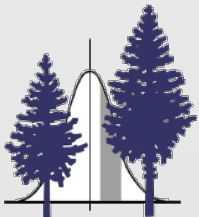
Concept - implementation

- rule system is defined by an directed graph

- procedural part is provided by the nodes

$\Rightarrow$ a model is described by two parts:
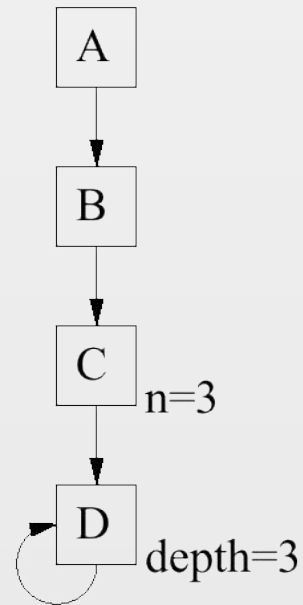
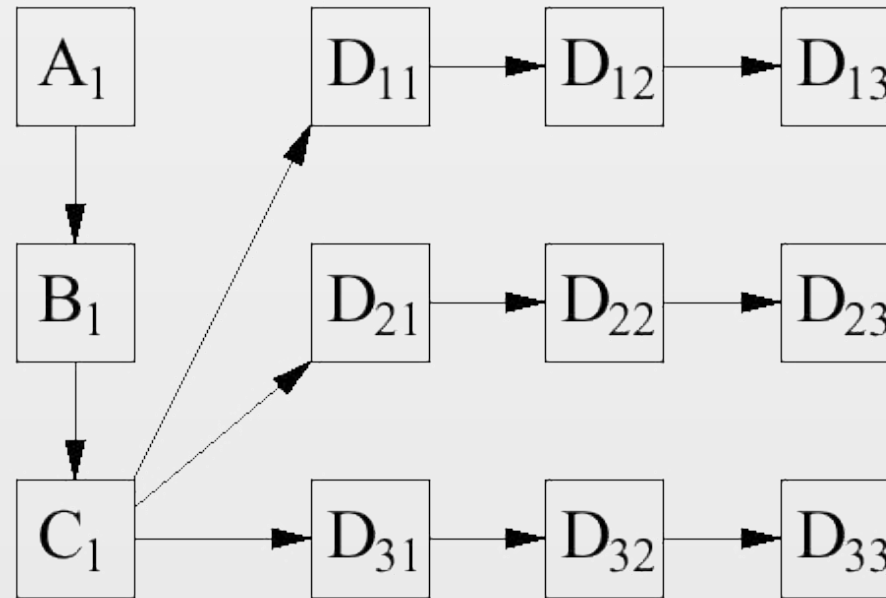 – graph structure
  – set of parameters
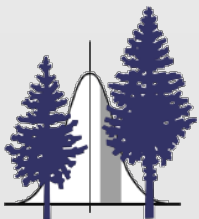
Working procedure
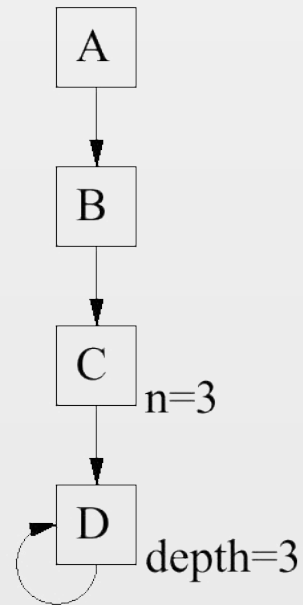

p-graph

Working procedure

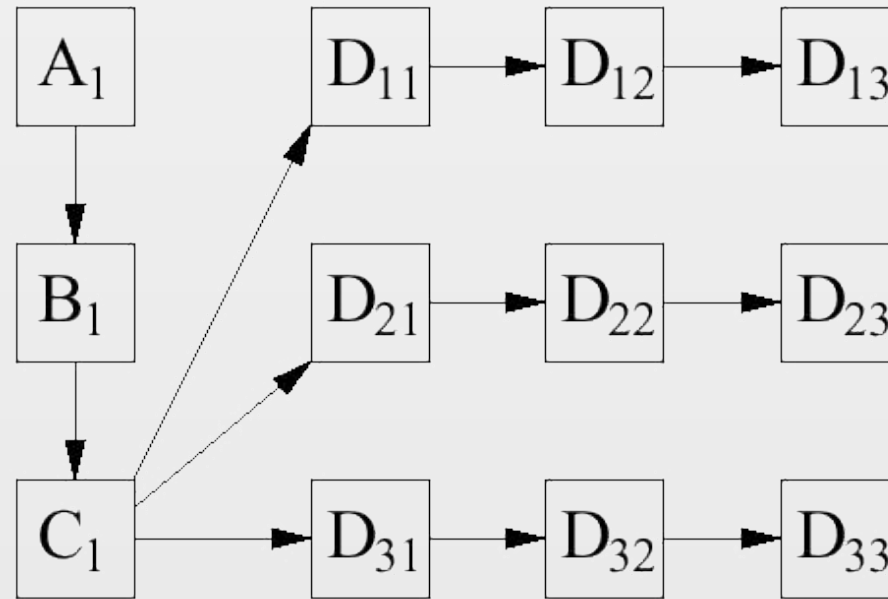p-graph   instantiation-tree
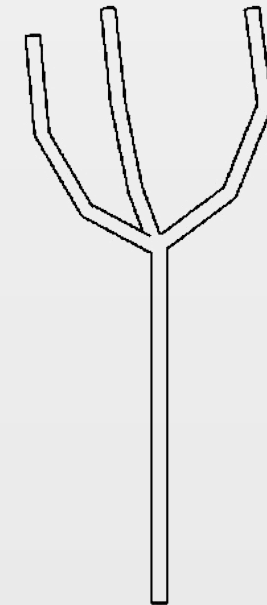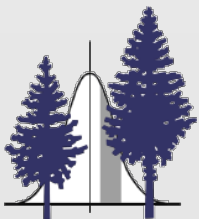
Working procedure

p-graph   instantiation-tree   geometry

Construction set – "old"

Tree Basic component for trees, creates the geometry of a stem and multiplies subsequent components as branches. Parameters are the distribution of branches, their scale, angle etc.

Construction set – "old"

Tree Basic component for trees, creates the geometry of a stem and multiplies subsequent components as branches. Parameters are the distribution of branches, their scale, angle etc.

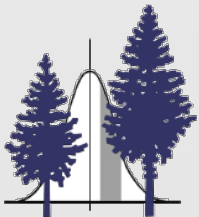**Horn** A component that places other components on a user-defined curve. It is used for stalks, twigs, etc.
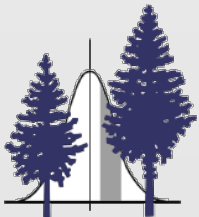
Construction set – "old"

Tree Basic component for trees, creates the geometry of a stem and multiplies subsequent components as branches. Parameters are the distribution of branches, their scale, angle etc.

Horn A component that places other components on a user-defined curve. It is used for stalks, twigs, etc.

Hydra Multiplies subsequent components on a curve with any direction to the direction of the parent component.

Institut für Forstliche Biometrie und Informatik

Construction set – "old"

Tree Basic component for trees, creates the geometry of a stem and multiplies subsequent components as branches. Parameters are the distribution of branches, their scale, angle etc.

Horn A component that places other components on a user-defined curve. It is used for stalks, twigs, etc.
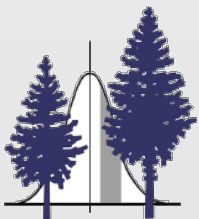
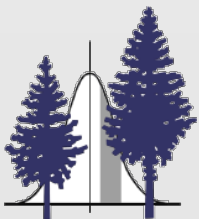**Hydra** Multiplies subsequent components on a curve with any direction to the direction of the parent component.

Wreath The functionality is integrated in the Hydra block.

Institut für Forstliche Biometrie und Informatik
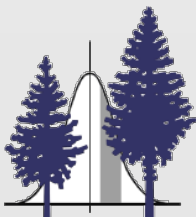
Construction set – "old"

Tree Basic component for trees, creates the geometry of a stem and multiplies subsequent components as branches. Parameters are the distribution of branches, their scale, angle etc.

Horn A component that places other components on a user-defined curve. It is used for stalks, twigs, etc.

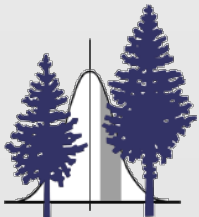**Hydra** Multiplies subsequent components on a curve with any direction to the direction of the parent component.

Wreath The functionality is integrated in the Hydra block.

PhiBall Multiplier that distributes all connected structures on a section of an ellipsoid according to the golden angle.
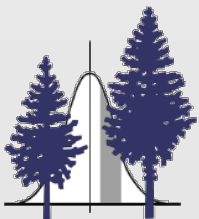
Construction set - "new"

Arrange Main block for arranging large numbers of instances on an area according to user-defined terrain data.

Construction set - "new"

Arrange Main block for arranging large numbers of instances on an area according to user-defined terrain data.

Variation Allows you in combination with an multipling-block to variate the generated instances, like sequential, spread, exceptions or random.
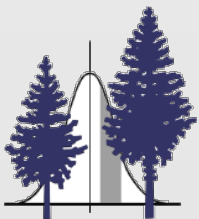
Construction set - "new"

Arrange Main block for arranging large numbers of instances on an area according to user-defined terrain data.

Variation Allows you in combination with an multipling-block to variate the generated instances, like sequential, spread, exceptions or random.

BlockScale Scaling block, enables scaling depending on internal variables.

BlockColor Enables coloring depending on internal variables.

Hydra block - attributes

XL-Code:

```
1  Hydra(<x>).(...)
2
3
4  setNumber(<x>)
5
6
7  setTrajectory(<BSplineCurve>)
8
9
10 setRadius(<x>), setRadius(<x1>, <x2>)
11
12
13 setTwist1(<x>), setTwist2(<x>),
14 setTwist(<x1>, <x2>)
15
16
17 setSpin1(<x>), setSpin2(<x>),
18 setSpin(<x1>, <x2>)
19
20
21 setScale1(<x>), setScale(<x>),
22 setScale(<x1>, <x2>)
23
24
25 setSlopeFunction(<FloatToFloat>)
```
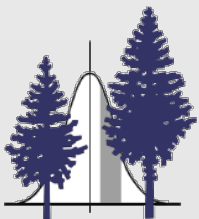
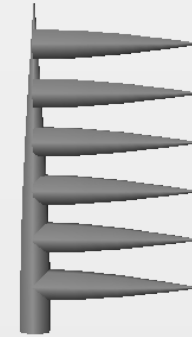$$x, x1, x2 \in [Strgin, double]$$

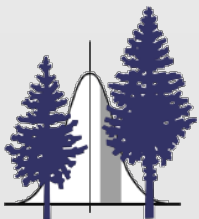attribute window

Edge types

- **MULTIPY-edge:**

  – standard edge

- **CHILD-edge:**

  – is only used by the tree and horn block

- **NEXT-edge:**

  – identifies different part for the variation block
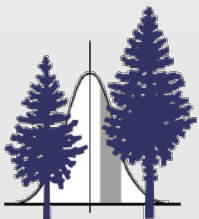
MULTIPY-edge

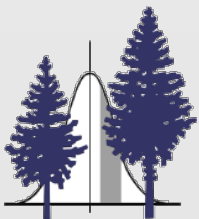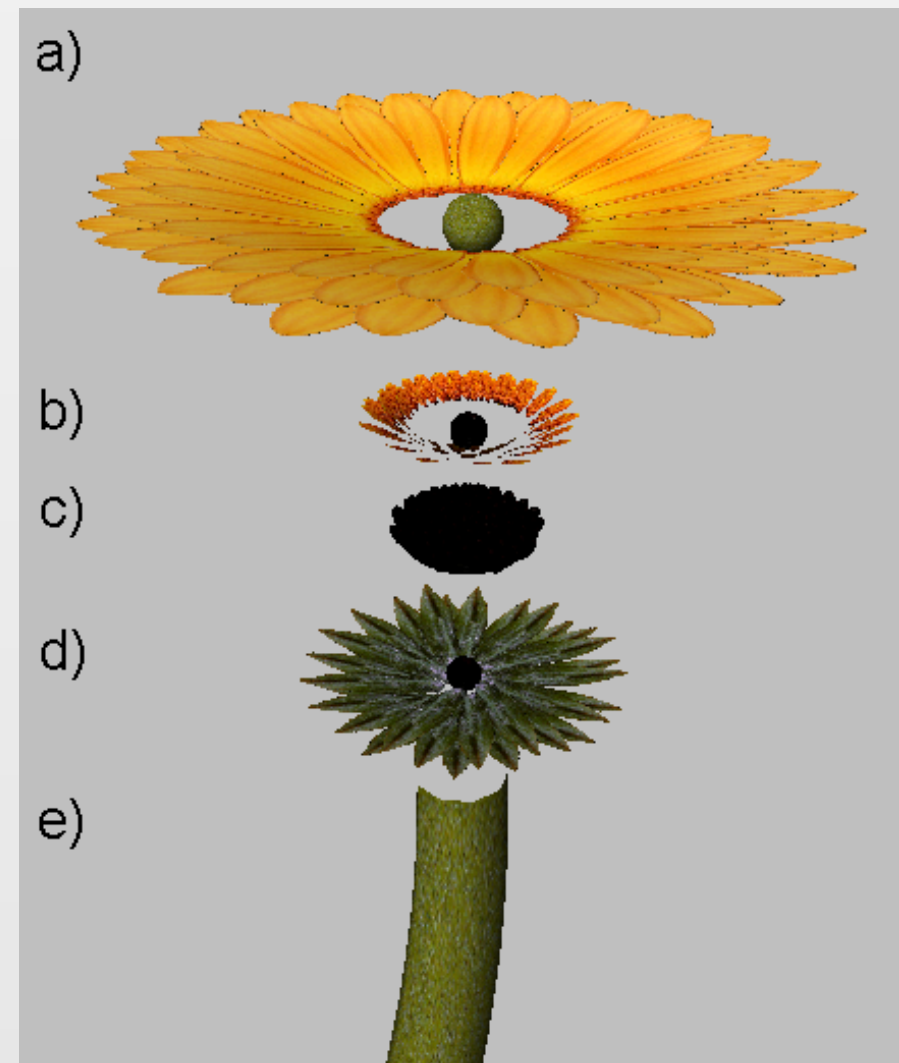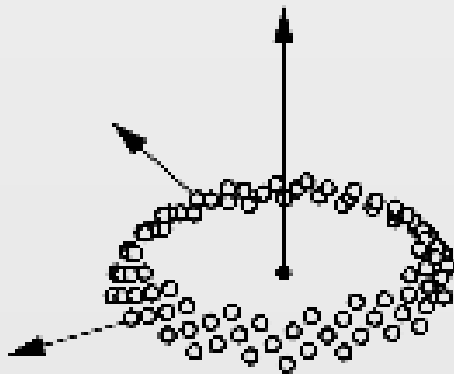CHILD-edge

Example

natural gerbera

GroIMP-model

Example construction

- construction:

  – horn block generates the shaft
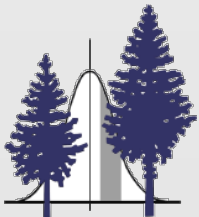 – every ring of petals is produced by a phiball block

Example source code

```
 1  const ShaderMat petalMat1 = shader("Lambert");
 2
 3  Parallelogram petal1 () {
 4      TMatrix4d m = new TMatrix4d();
 5      m.rotZ(Math.PI/2);
 6      return leaf(5f, 2f).(setMaterial(petalMat), setTransform(m));
 7  }
 8
 9  Ring1 ==> PhiBall(75).(setRadius(1.5, 1.5, 0.6), setFan(1.5, 1.35),
10          setScale1(0.8)) -MULTIPLY-> petal1;
11
12  shaft ==> Horn(15).(setMaterial(shaftMat), setLength(0.375), setRotZ1(0.05),
13          setRotX(-0.05, 0.05), setRange(1.4, 1.25), setScale1(0.06));
14
15  Axiom ==> shaft [-CHILD-> Scale(1.5) [Ring1]]
16                  [-CHILD-> Ring2]
17                  [-CHILD-> Ring3]
18                  [-CHILD-> Ring4];
```
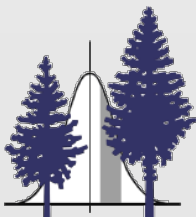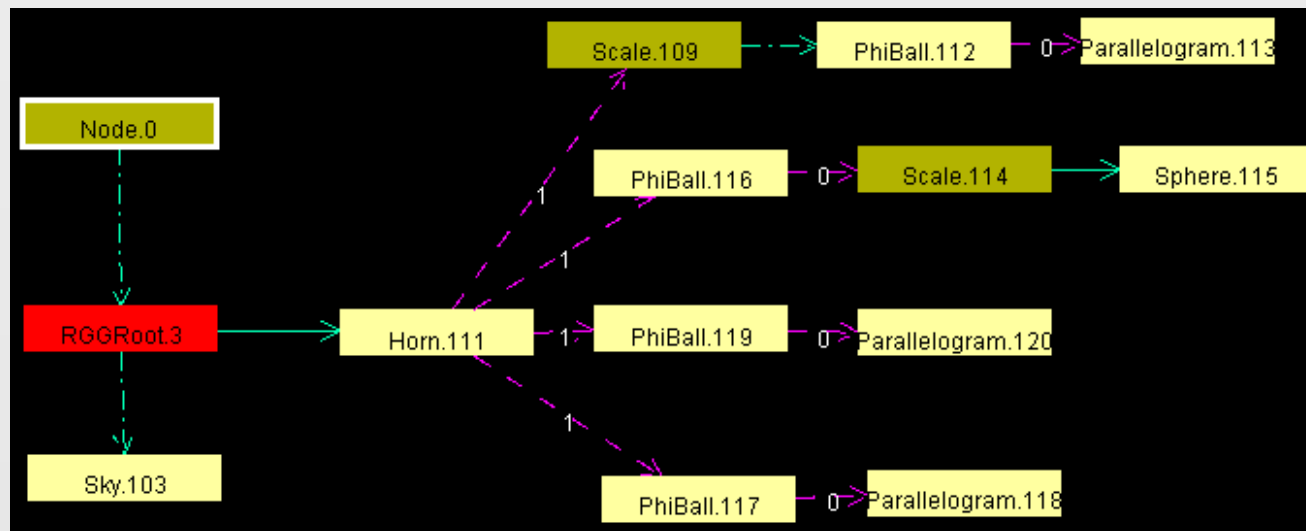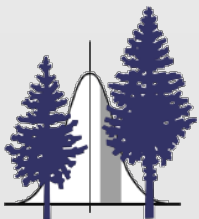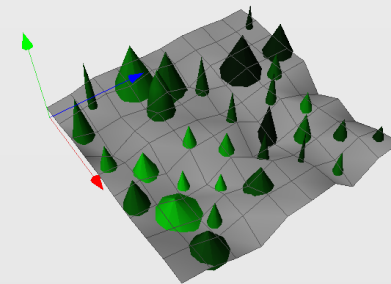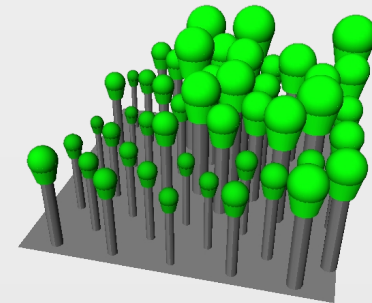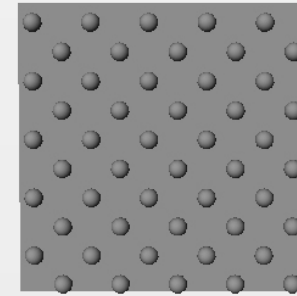
Example source code

```
1  const ShaderMat petalMat1 = shader("Lambert");
2
3  Parallelogram petal1 () {
4      TMatrix4d m = new TMatrix4d();
5      m.rotZ(Math.PI/2);
6      return leaf(5f, 2f).(setMaterial(petalMat), setTransform(m));
7  }
8
9  Ring1 ==> PhiBall(75).(setRadius(1.5, 1.5, 0.6), setFan(1.5, 1.35),
10         setScale1(0.8)) -MULTIPLY-> petal1;
11
12 shaft ==> Horn(15).(setMaterial(shaftMat), setLength(0.375), setRotZ1(0.05),
13         setRotX(-0.05, 0.05), setRange(1.4, 1.25), setScale1(0.06));
14
15 Axiom ==> shaft [-CHILD-> Scale(1.5) [Ring1]]
16                 [-CHILD-> Ring2]
17                 [-CHILD-> Ring3]
18                 [-CHILD-> Ring4];
```
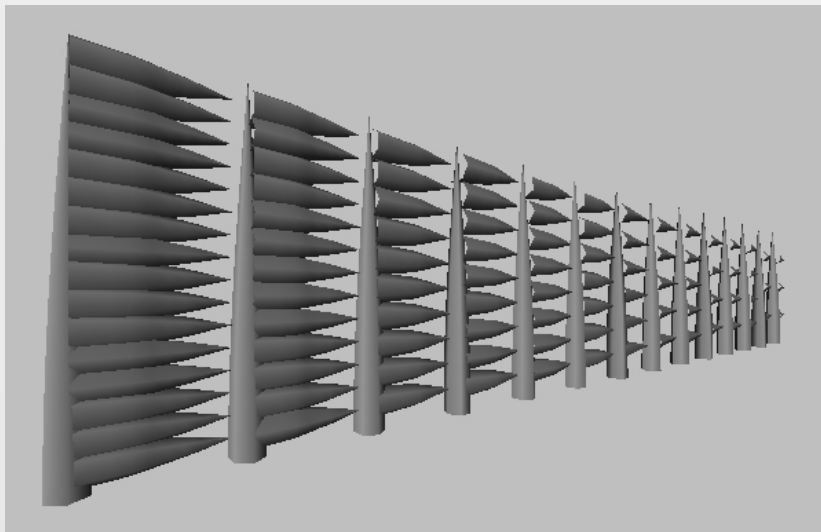
Arrange block

- arrange great amount of instances on a terrain

- distribution classes:

    - geometric arrangement
    - probability arrangement
    - halftoning
    - additional operations

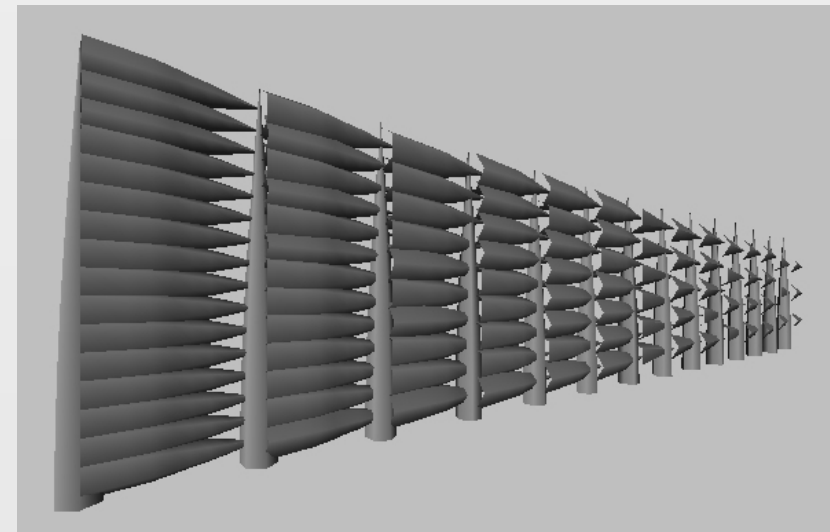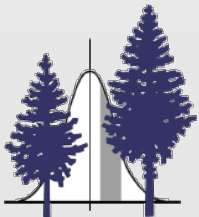- user defined location parameter field

Level of detail

- dependence on the distance of the object to the camera the model is represented in different resolutions

- technique based on Lintermann "Drop and Resize"
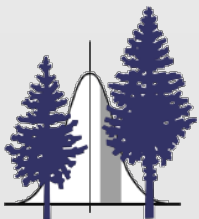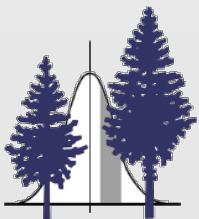

scale factor = zero

scale factor = maximal

Conclusion

+ intuitive modelling way, which fast produce handsome results

+ easy method to control geometrical and structural complexity

+ great diversity of objects can be modelled

Conclusion

+ intuitive modelling way, which fast produce handsome results

+ easy method to control geometrical and structural complexity

+ great diversity of objects can be modelled

- biological not funded; appearance depends only on the modeller

- dates taken from nature are heavily too transmit

- limited for simulation of growing processes

- performance problems with bigger models

Literature

- Home page of the interactive modelling platform GroIMP; BTU Cottbus: Ole Kniemeyer and Winfried Kurth, www.grogra.de

- O. Deussen; Computergenerierte Pflanzen, Technik und Design digitaler Pflanzenwelten; Springer, Berlin, 2003

- M. Henke; Entwurf und Implementation eines Baukastens zur 3D-Pflanzenmodellierung in GroIMP; Diploma thesis, BTU Cottbus, 2007