Advanced Modeling

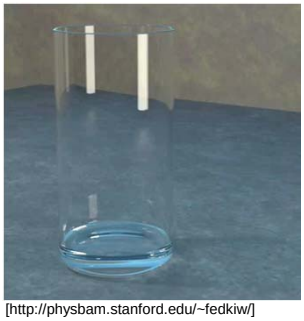
Eduard Gröller, Thomas Theußl,
Peter Rautek

Institute of Computer Graphics and Algorithms

Vienna University of Technology

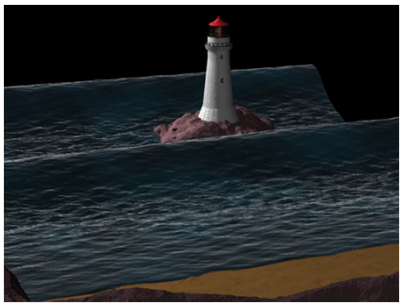
Motivation

- Real world phenomena
  - Complex geometry
  - Large deformations
  - Topological changes
  - Fuzzy objects
- Tedious or impossible to model with meshes
- Examples
  - Smoke, fire
  - Fluids
  - Fur, hair, grass



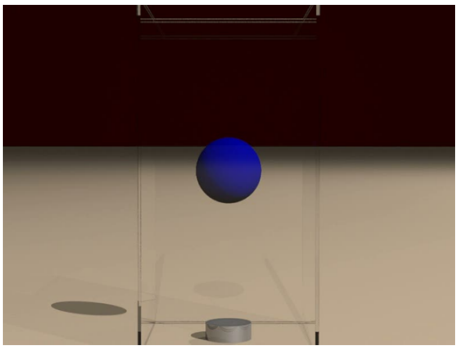
[http://physbam.stanford.edu/~fedkiw/]

Motivation



[http://physbam.stanford.edu/~fedkiw/]

Motivation



[http://physbam.stanford.edu/~fedkiw/]

Overview

- Particle systems
- Implicit modeling
  - Soft objects
  - Superquadrics
  - Level sets
- Procedural modeling
  - Sweeps
  - Cellular texure generation
  - Terrain simulation
  - Vegetation simulation
- Structure-deforming transformations

Particle Systems

- Modeling of objects changing over time
  - Flowing
  - Billowing
  - Spattering
  - Expanding
- Modeling of natural phenomena:
  - Rain, snow, clouds
  - Explosions, fireworks, smoke, fire
  - Sprays, waterfalls, lumps of grass

[Matthias Müller]

Particle Systems - History

- 1982 Star Trek II: The Wrath of Khan


"A particle system is a collection of many many minute particles that together represent a fuzzy object. Over a period of time, particles are generated into a system, move and change from within the system, and die from the system."

William T. Reeves
Particle Systems - A Technique for Modeling a Class of Fuzzy Objects
ACM Transactions on Graphics, 1983

Particle Systems

- Certain number of particles is rendered
- Particle parameters change over time:
  - Location
  - Speed
  - Appearance
- Particles die (lifetime) and are deleted

Particle Systems (2)

- Particle shapes may be spheres, boxes, or arbitrary models
- Size and shape may vary over time
- Motion may be controlled by external forces, e.g. gravity

Particle Systems (3)

- Particles interfere with other particles

Particle Systems: Bomb

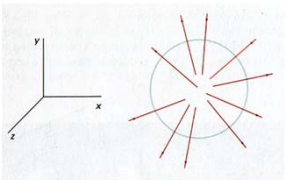


Figure 10-111
Modeling fireworks as a particle system with particles traveling radially outward from the center of the sphere.

Particle Systems: Grass Clumps

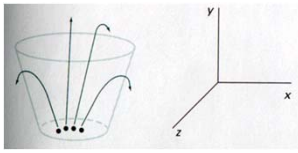
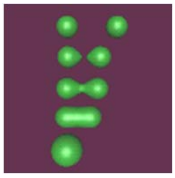


Figure 10-112
Modeling a clump of grass by firing particles upward within a tapered cylinder. The particle paths are parabolas due to the downward force of gravity.

lifetime can be encoded by color: from green to yellow

Implicit Modeling

- No fixed shape and topology
- Modeling of
  - Molecular structures
  - Water droplets
  - Melting objects
  - Muscle shapes
- Shape and topology change
  - In motion
  - In proximity to other objects

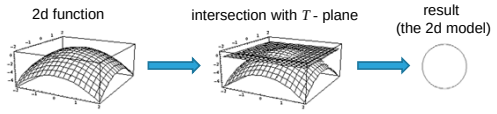


Implicit Modeling

- No seams
- Oriented surface
  (well defined inside and outside)
- Differentiable
- Closed
- Continuous

Implicit Modeling

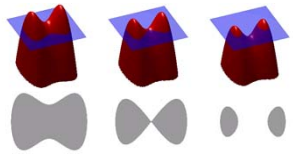
- Implicit equation e.g., $f(x,y) = -(x^2 + y^2) = T$
- Vs. explicit equation e.g., $y = kx + d$
- Function $\Re^n \rightarrow \Re$
- Right side constant (typically a threshold T)



The surface of an implicit model is defined as the set of points that fulfill the implicit equation

Implicit Modeling

- Level sets
  - $\Re^2 \rightarrow \Re$ level curve, iso contour, contour line
  - $\Re^3 \rightarrow \Re$ level surface, iso surface
  - $\Re^n \rightarrow \Re$ level hypersurface
- Changing the threshold

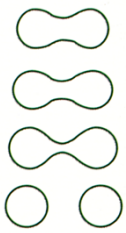


change of topology

Soft Objects: Blobs

- Volume stays constant during movement

- Molecular bonding:
  As two molecules move away from each other, the surface shapes
  - Stretch
  - Snap and finally
  - Contract into spheres



Definition of Blobby Objects

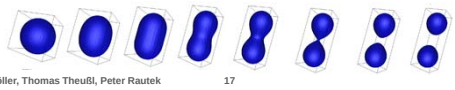
- Sum of Gaussian density functions centered at the k control points $X_k = (x_k, y_k, z_k)$

$$f(x,y,z) = \sum_k b_k e^{-a_k r_k^2} - T = 0$$

where

$$r_k^2 = (x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2$$

T is a specified threshold, and a_k and b_k adjust the blobbiness of control point k



Definition of Blobby Objects

- Metaball model uses density functions, which drop off to *0* at a finite interval
- Soft object model uses same approach with a different density-distribution characteristic

Superquadrics

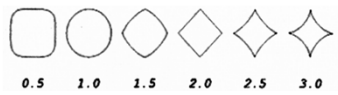
- Generalization of quadric representation
- Additional parameters
- Increased flexibility for adjusting object shapes
- One additional parameter for curves and two parameters for surfaces

Superellipse

- Exponent of x and y terms of a standard ellipse are allowed to be variable:

$$\left(\frac{x}{r_x}\right)^{2/s} + \left(\frac{y}{r_y}\right)^{2/s} = 1$$

- Influence of s:

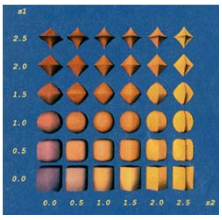


Superellipsoid

- Exponent of x, y and z terms of a standard ellipsoid are allowed to be variable:

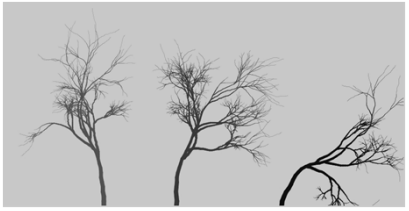
$$\left[\left(\frac{x}{r_x}\right)^{2/s_2} + \left(\frac{y}{r_y}\right)^{2/s_2}\right]^{s_2/s_1} + \left(\frac{z}{r_z}\right)^{2/s_1} = 1$$

- Influence of s_1 *and* s_2:



Procedural Modeling

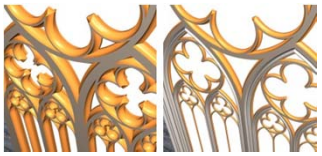
- High geometric complexity
- Complex model does not exist as geometry
  - Set of production rules



Demo Procedural Modeling

Motivation

- One window in highest resolution ~7 million triangles
- Modeled with 126 KB (18 KB zipped) of code
- Changing parameters yields very different models

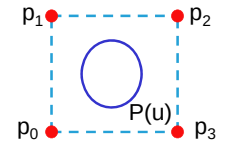


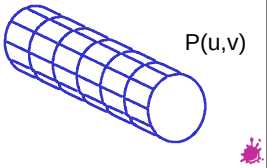
Sweeps

- Modeling of objects with symmetries:
  - Translational
  - Rotational
- Represented by
  - 2D shape
  - Sweep-path

Translational Sweeps

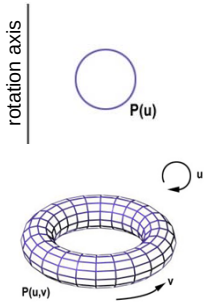
- Control points of spline curve P(u)
- Generates the solid, whose surface is described by point function P(u,v)





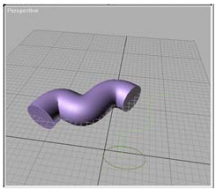
Rotational Sweeps

- Spline curve P(u)
- Rotated about given rotation axis
- Sampled at given angles yields the surface P(u,v)



General Sweeps

- Spline curve P(u)
- Moved along a sweep path (e.g., spline)
- Animated sweep path



[Kinetix 3D Studio MAX]

Sweeps - Pros and Cons

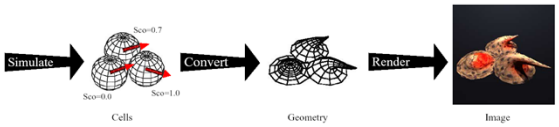
- Advantages:
  - Generates shapes that are hard to do otherwise
- Disadvantages:
  - Hard to render
  - Difficult modeling

Example

Cellular Texture Generation

- A cellular particle system, that changes geometry of surface
  - cell state
  - cell programs
  - extracellular environments

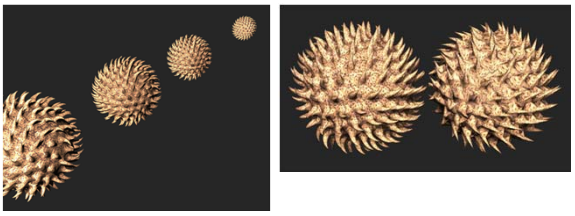
Cellular Texture Generation



- Cell state: position, orientation, shape, chemical concentrations (reaction-diffusion)
- Cell programs:
  - Go to surface, die if too far from surface, align, adhere to other cells, divide until surface is covered, ...
  - Differential equations
- Extra cellular environment: neighbor orientation, concentration, ...

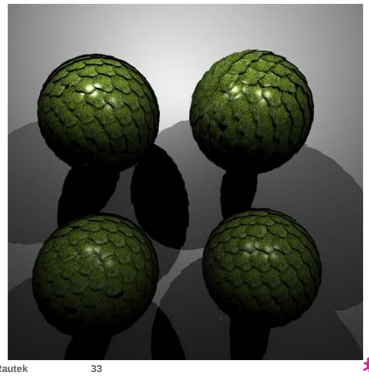
Cellular Texture Generation 2

- Levels of Detail (LOD): Use fewer polygons for further distances



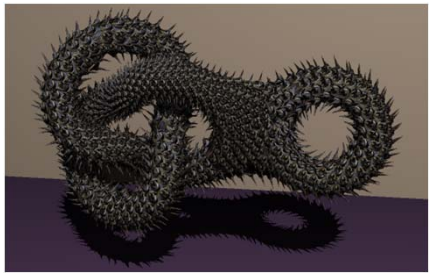
Cellular Texture Generation 3

- Cell: group of polygons with texture and transparency maps



Cellular Texture Generation - Examples

- Handling of unusual topologies
- No problem with parameterization



Cellular Texture Generation - Examples

- Reaction-diffusion determine pattern of bumps and thorns

Cellular Texture Generation - Examples

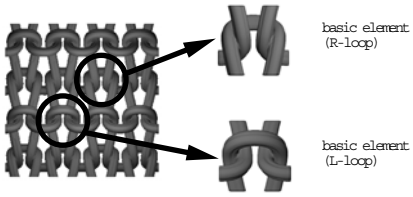
- Cells (fur) oriented like their neighbors

Cellular Texture Generation - Examples

- Cells (fur) similarly oriented

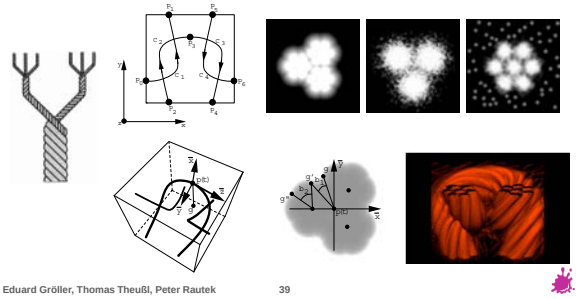
Modeling and Visualization of Knitwear

- Knitwear: simulation of thin 3D structure with instanced volume elements



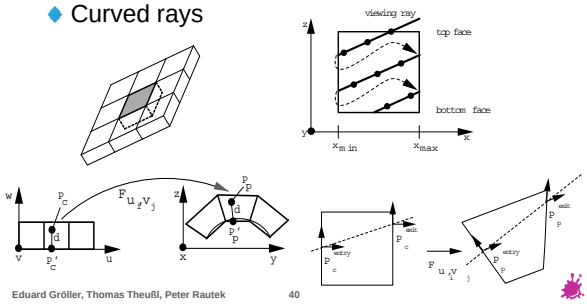
Visualization of Knitwear

- Volume element: 2D cross-section swept + rotated along parametric curve

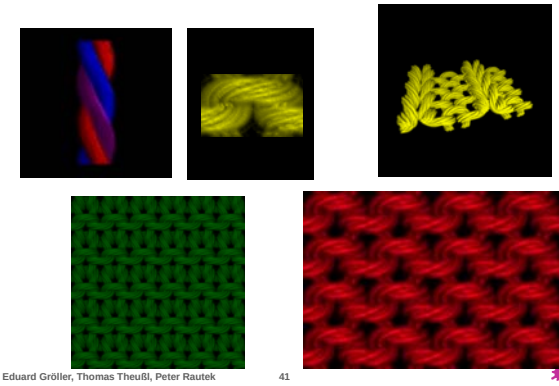


Visualization of Knitwear

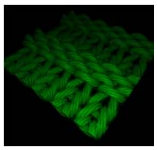
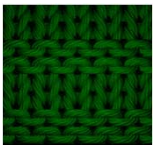
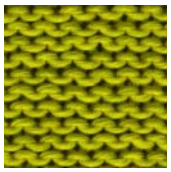
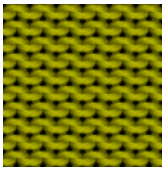
- Rendering with raycasting
  - Surface tiled with volumetric elements
  - Curved rays



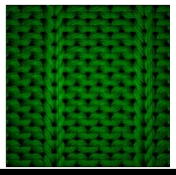
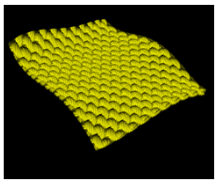
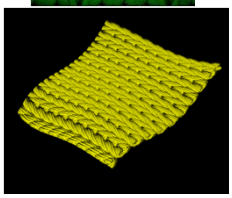
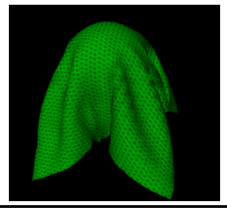
Knitwear - Examples



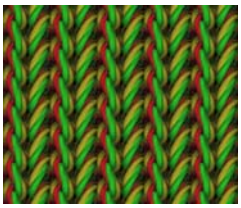
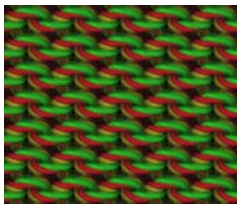
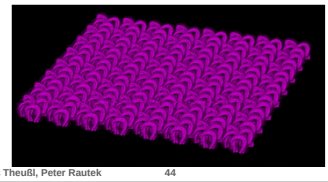
Knitwear - Examples

Knitwear - Examples

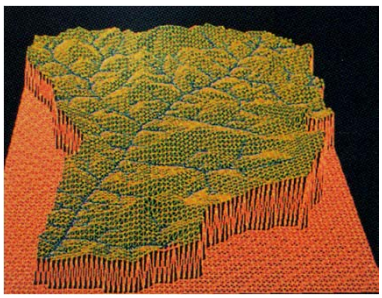
Knitwear - Examples

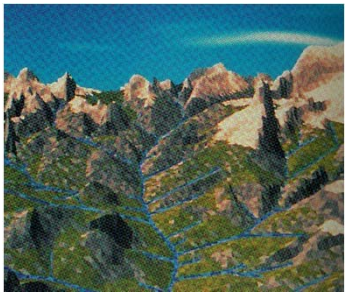
Terrain Simulation

- Fractals
- Geographical Data
- Simulations
- Hybrids

Terrain Simulation



Terrain Simulation



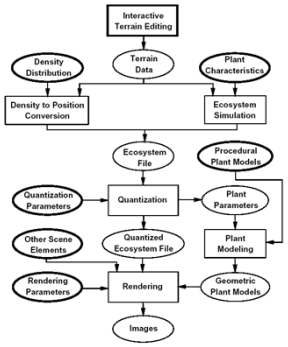
Terrain Simulation

Terrain Simulation

Terrain Simulation

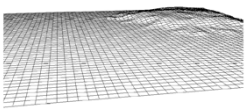
Realistic modeling and rendering of plant

Scene Synthesis System

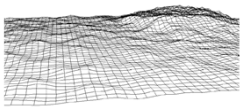


Terrain

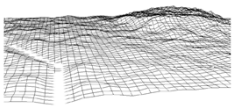
Height map



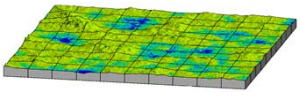
Hills through noise synthesis



Stream through masking



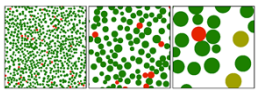
Water concentration (blue=high, yellow=low)



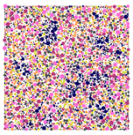
Specification of Plant Populations

- Space-occupancy
  - Explicit specification (counting plants, painting)
  - Procedural generation (cellular automata, reaction-diffusion)
- Individual based
  - Explicit specification (survey, interactive specification)
  - Procedural generation (point pattern generation model)

Self-thinning:
Green: not dominated
Red: dominated
Yellow: old



Distribution of eight species



Realistic Modeling and Rendering of Plants

- Complex models necessary for realistic appearance
  - Plant distribution by ecosystem simulation and/or manual setting
  - Reduce geometric complexity by approximate instancing (similar plants, groups of plants or plant organs)
  - Parametrized models of individual plants

Plant Ecosystems - Examples

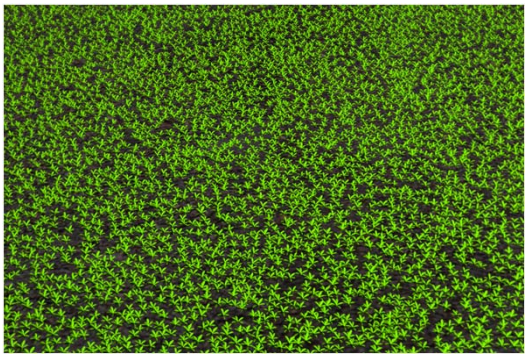
Plant Ecosystems - Examples

Plant Ecosystems - Examples

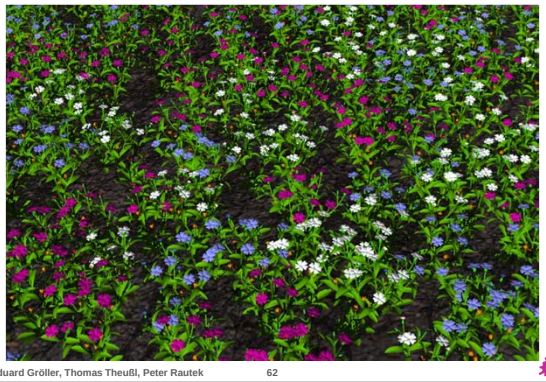
Plant Ecosystems - Examples

Plant Ecosystems - Examples

Plant Ecosystems – Self Thinning



Plant Ecosystems – Self Thinning



Plant Ecosystems - Examples

Plant Ecosystems - Examples

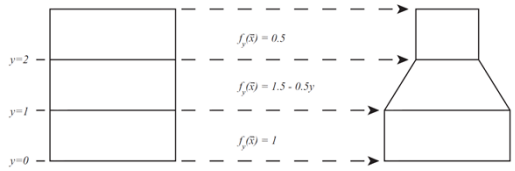
Structure-Deforming Transformations

- Non-linear transformations
  - Tapering: non-linear scaling
  - Twist: non-linear rotation
  - Bend: also non-linear rotation

Tapering

- Scale factor is a function:

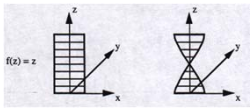
$$\vec{x}' = \begin{pmatrix} f_x(\vec{x}) & 0 & 0 \\ 0 & f_y(\vec{x}) & 0 \\ 0 & 0 & f_z(\vec{x}) \end{pmatrix} \cdot \vec{x}$$



Twist

- Angle of rotation is a function
  e.g., for rotation about z-axis

$$\vec{x}' = \begin{pmatrix} \cos f(\vec{x}) & -\sin f(\vec{x}) & 0 \\ \sin f(\vec{x}) & \cos f(\vec{x}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \vec{x}$$

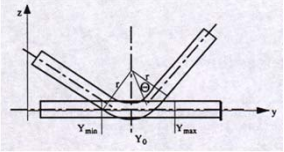


Bend

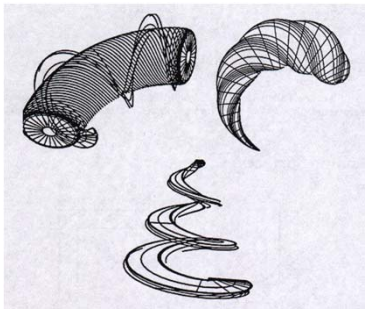
- Also non-linear rotation

$$Y = \begin{cases} -\sin(\Theta)(z-r)+y_0 & ,y_{min}<y<y_{max} \\ -\sin(\Theta)(z-r)+y_0+\cos(\Theta)(y-y_{min}), & y<y_{min} \\ -\sin(\Theta)(z-r)+y_0+\cos(\Theta)(y-y_{max}), & y>y_{max} \end{cases}$$

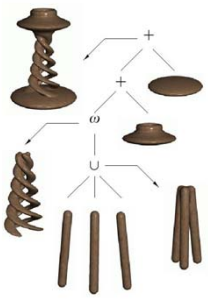
$$Z = \begin{cases} \cos(\Theta)(z-r)+y_0+r & ,y_{min}<y<y_{max} \\ \cos(\Theta)(z-r)+y_0+r+\sin(\Theta)(y-y_{min}), & y<y_{min} \\ \cos(\Theta)(z-r)+y_0+r+\sin(\Theta)(y-y_{max}), & y>y_{max} \end{cases}$$



Example 1

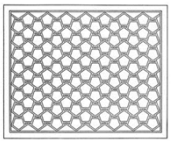


Example 2



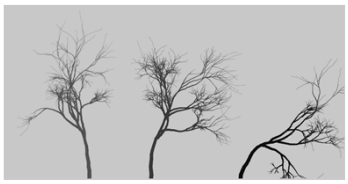
Other Topics not Covered Here

- Shape grammars
- Procedural architecture
- Fractals (see Fraktale VO WS)



Thank you for your attention

Questions?



References

P. Bourke. Implicit surfaces. `http://paulbourke.net/miscellaneous/implicitsurf/`, 1997.

S. Havemann. *Generative Mesh Modeling*. PhD thesis, TU Braunschweig, 2005.

D. Hearn and M. P. Baker. *Computer graphics (2nd ed.): C version*. Prentice-Hall, Inc., 1997.

M. Lentine, W. Zheng, and R. Fedkiw. A novel algorithm for incompressible flow using only a coarse grid projection. *ACM Transactions on Graphics*, 29(4), 2010.

F. Losasso, J. O. Talton, N. Kwatra, and R. Fedkiw. Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):797–804, 2008.

B. Wyvill, A. Guy, and E. Galin. The blobtree. warping, blending and boolean operations in an implicit surface modelling system. Technical report.