

CMSC 435

Introductory Computer Graphics

Penny Rheingans
UMBC

Announcements

- Proj4
 - Due Nov 12
 - Questions?

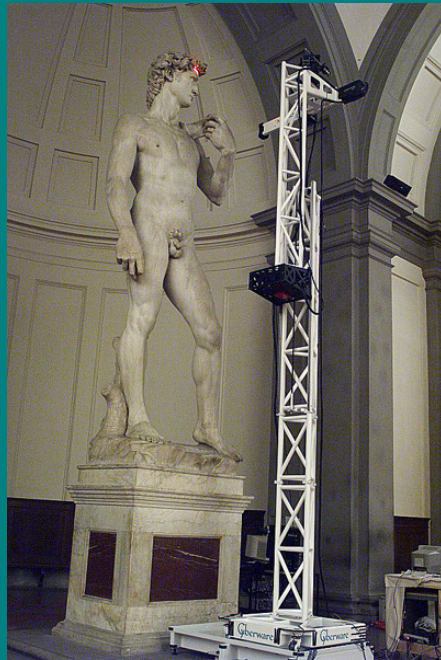
Modeling Approaches

- Manual primitive creation
- Scanning from physical object
- Procedurally
- From data (visualization)
- Through image capture (image-based rendering)

Scanning



- Digital Michaelangelo Project at Stanford



Procedural Modeling

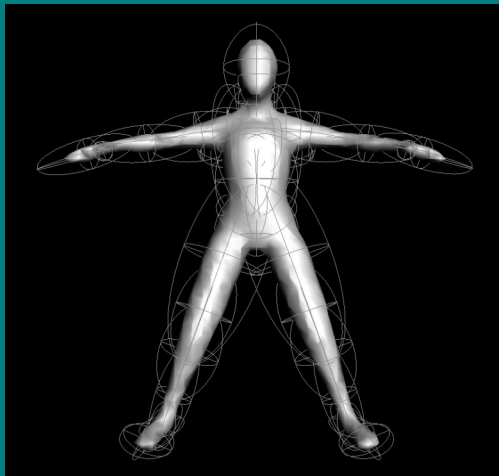
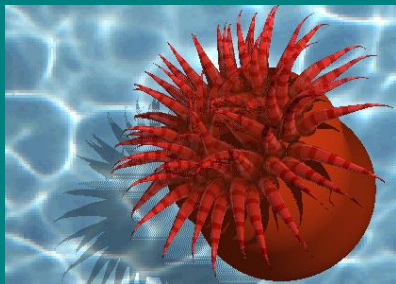
- Describe visual attributes through some function, usually defined over space
 - Shape
 - Density
 - Color
 - Texture

Procedural Approaches

- Implicit Functions
- Noise/hypertexture
- Fractals
- Grammars
- Genetic/biological simulations

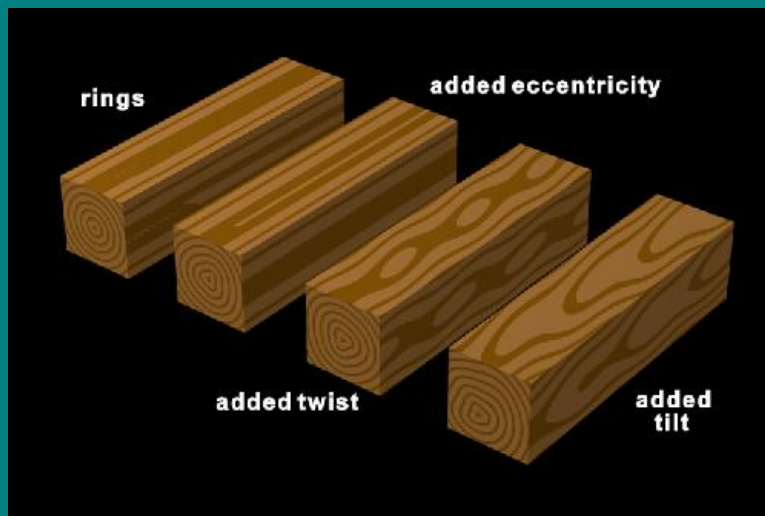
Implicit Functions

- Model as sum of implicit functions
- Surface at threshold





3D Solid Texture

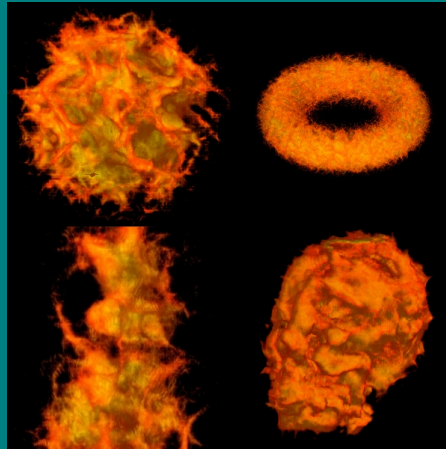
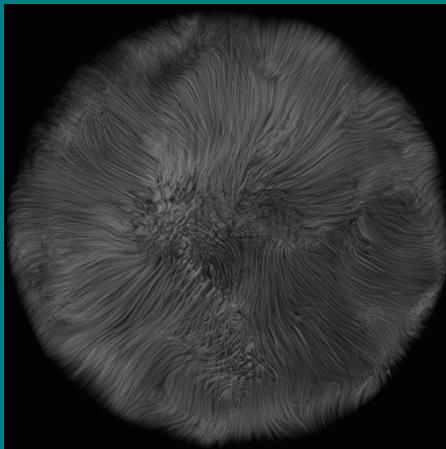


3D Solid Texture



Hypertexture

- Add noise or turbulence to functions



Hypertexture

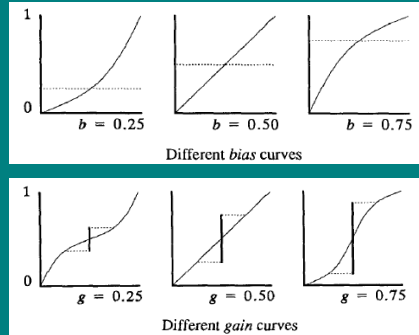
- Ken Perlin and Eric Hoffert, Hypertexture, SIGGRAPH '89.
- Extend 3D procedural noise textures to include opacity component to create volume models
 - object density function $D(x)$
 - Density modulation function (DMF) f_i

Boolean Operations

- Intersection
- Complement
- Difference
- Union

Base DMFs

- Bias
- Gain
- Noise
- Turbulence
- Arithmetic functions

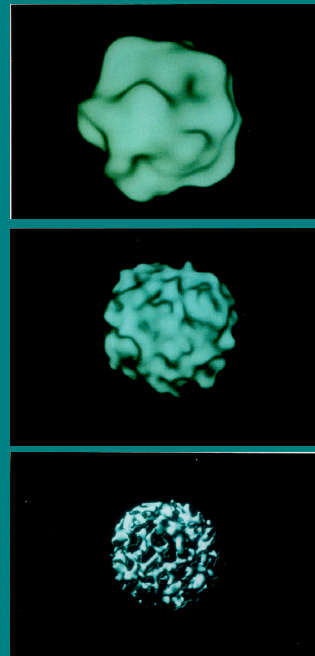


Basic Noise

- Basic noisy sphere

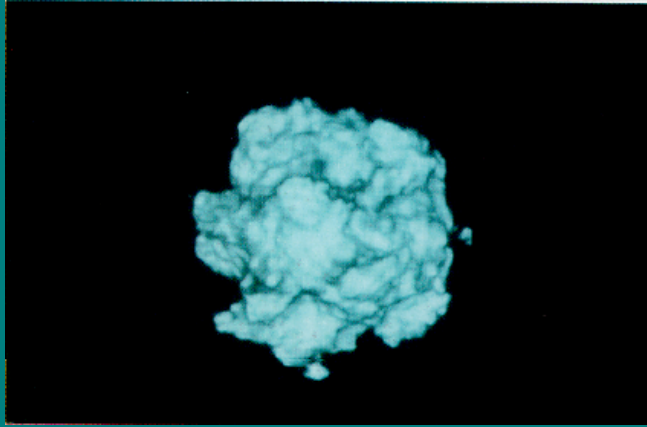
$$D(\mathbf{x}) = \text{sphere}(\mathbf{x} (1 + \frac{1}{f} \text{noise}(f\mathbf{x})))$$

- Vary
 - Frequency
 - Amplitude



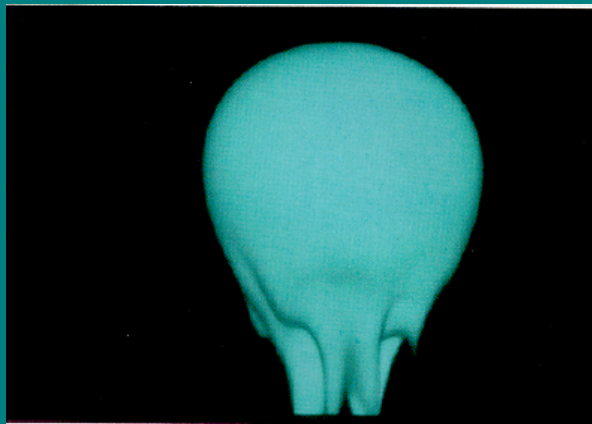
Turbulence

- Mix different frequencies of noise



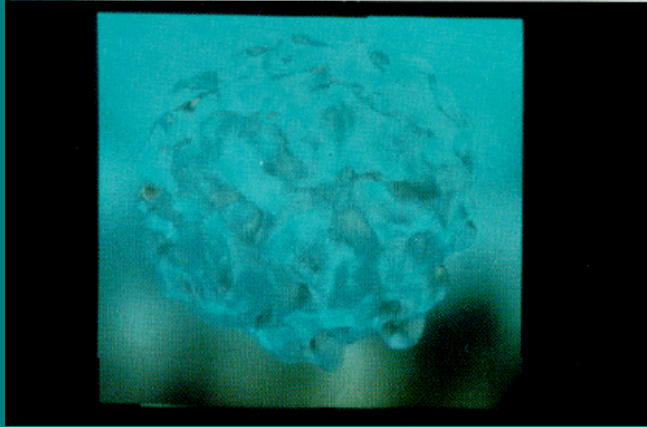
Shaped Noise

- Vary only single component



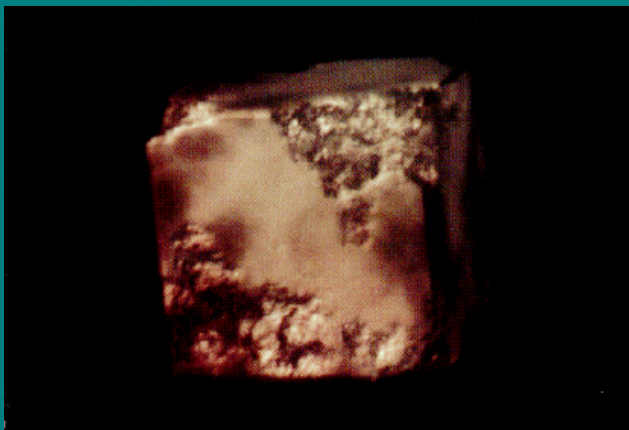
Transparency

- Refractive Hypertexture



Erosion

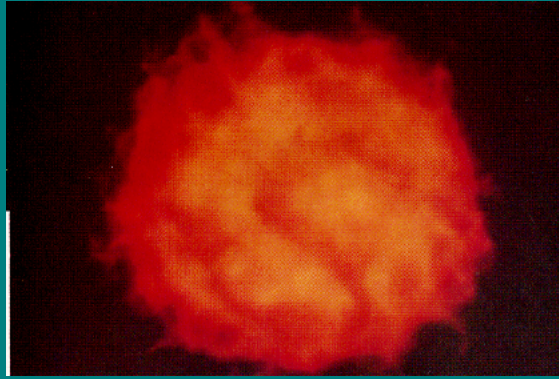
- Boolean intersection of fractal sphere with cube



Fire

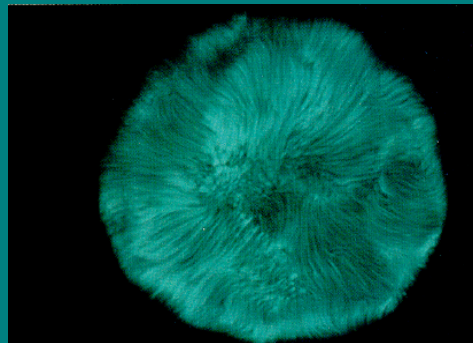
- Density func:
- Colormap

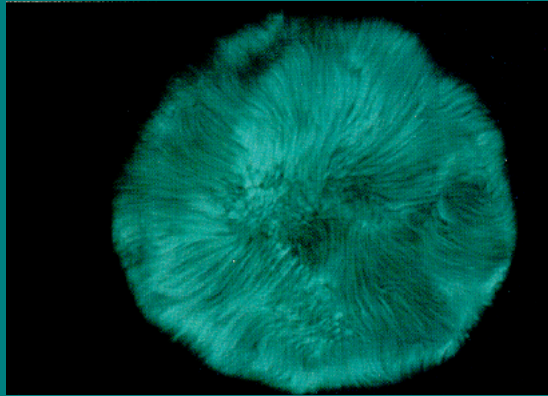
$$D(\mathbf{x}) = \text{sphere}(\mathbf{x} (1 + \text{turbulence}(\mathbf{x})))$$



Fur

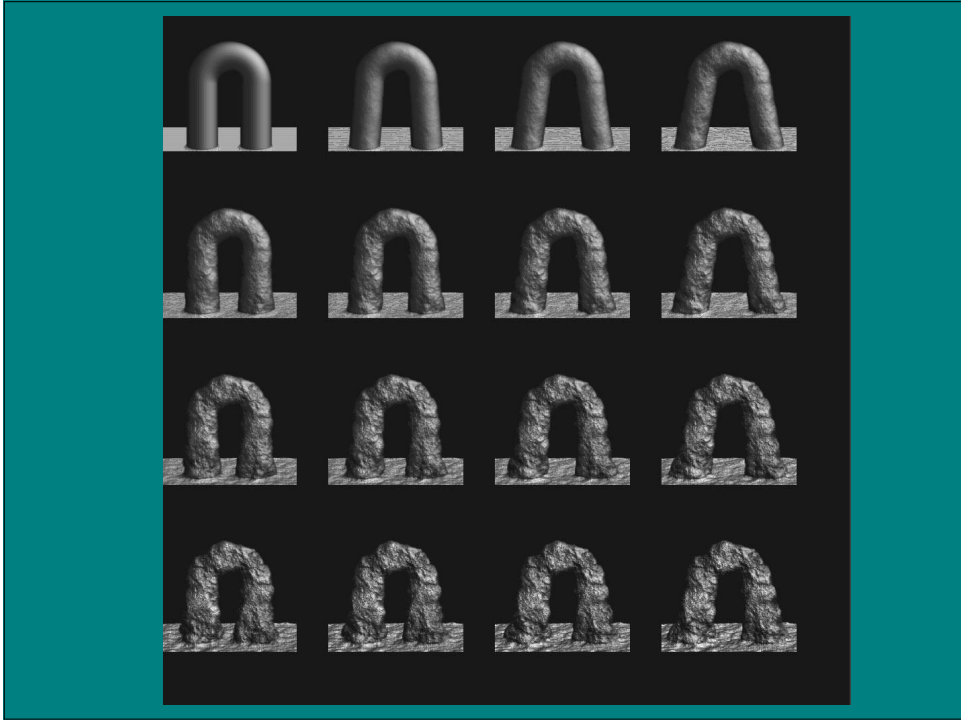
- Project points to create hairs
- Modulate density
- Control bias and gain
- Add noise in growth direction

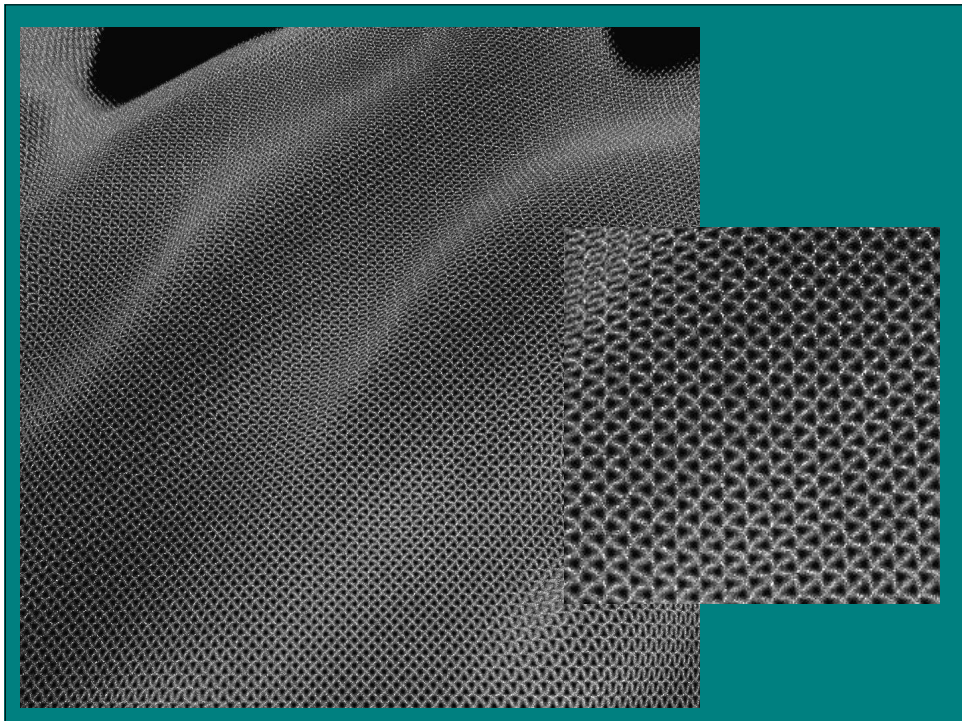
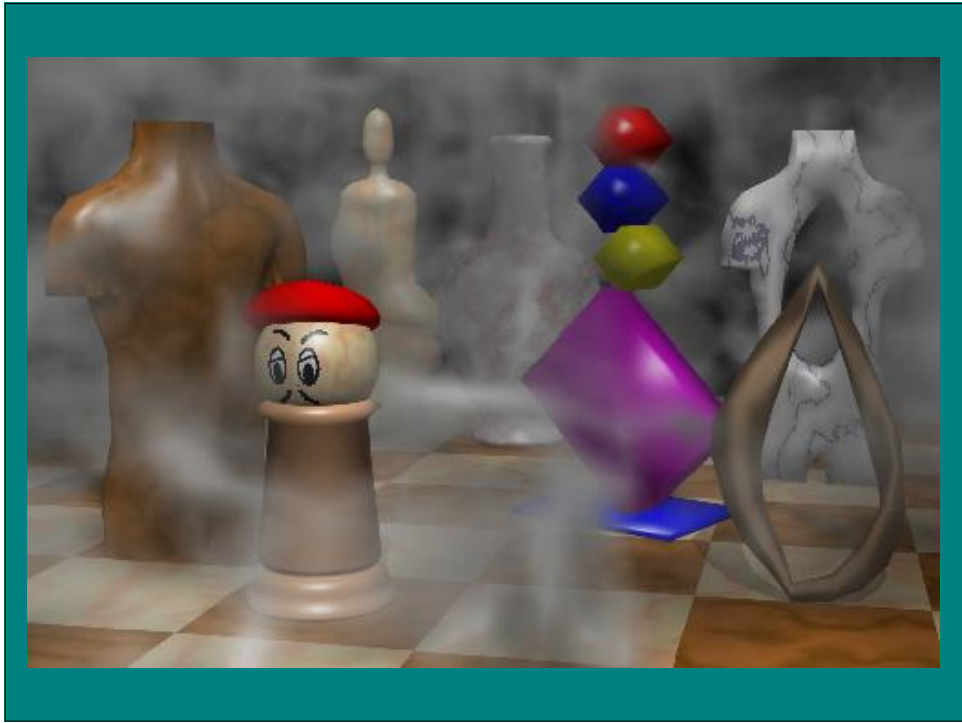


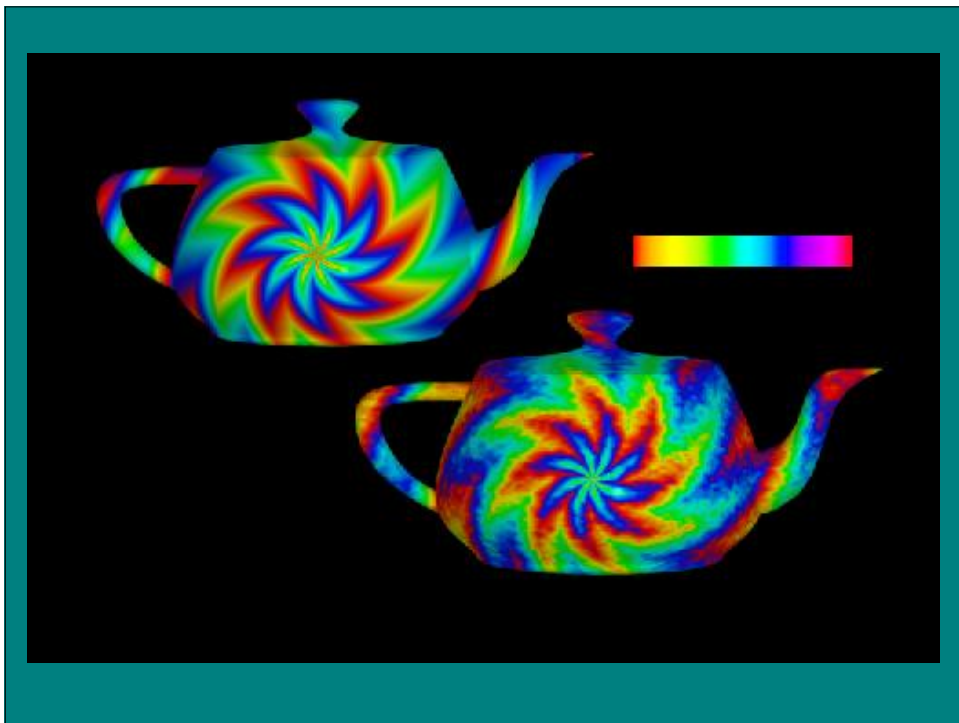


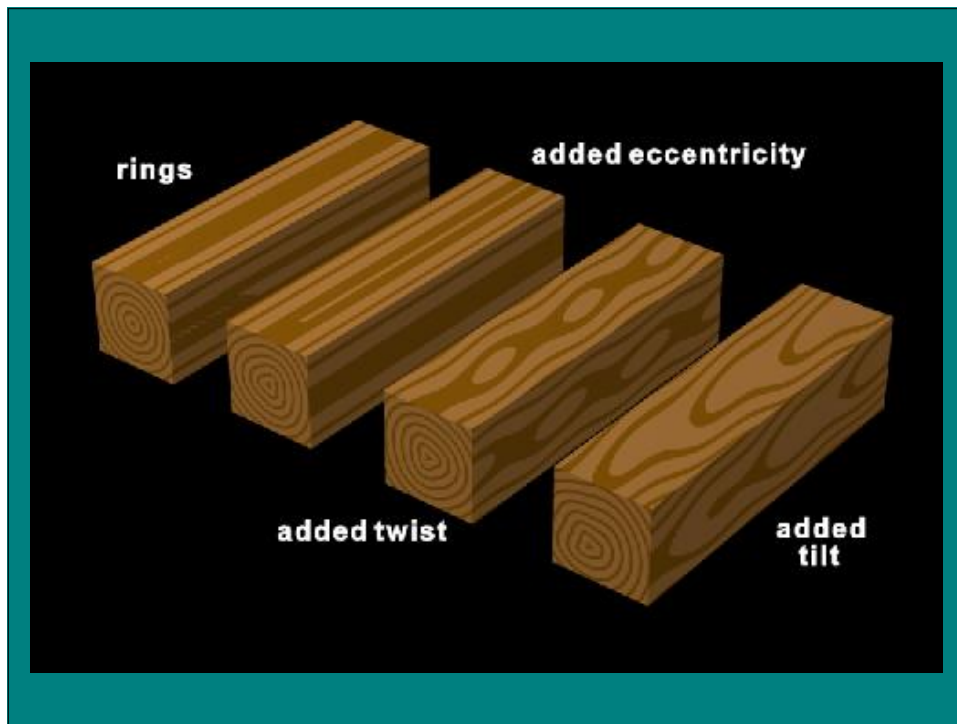
Noisy Things

- Color
- Specularity
- Opacity/Density
- Normals
- Displacements
- Index of Refraction
- Procedural Texture Parameters









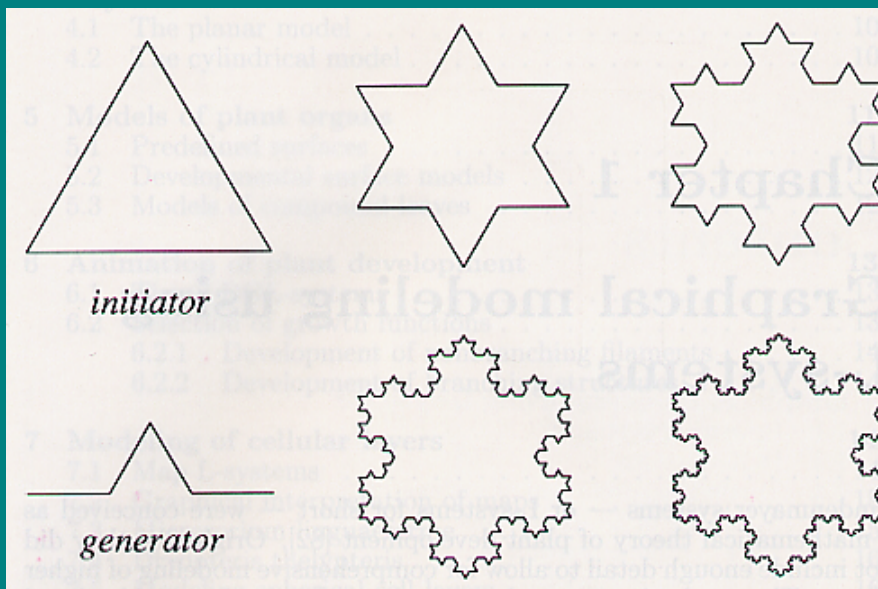
Fractals

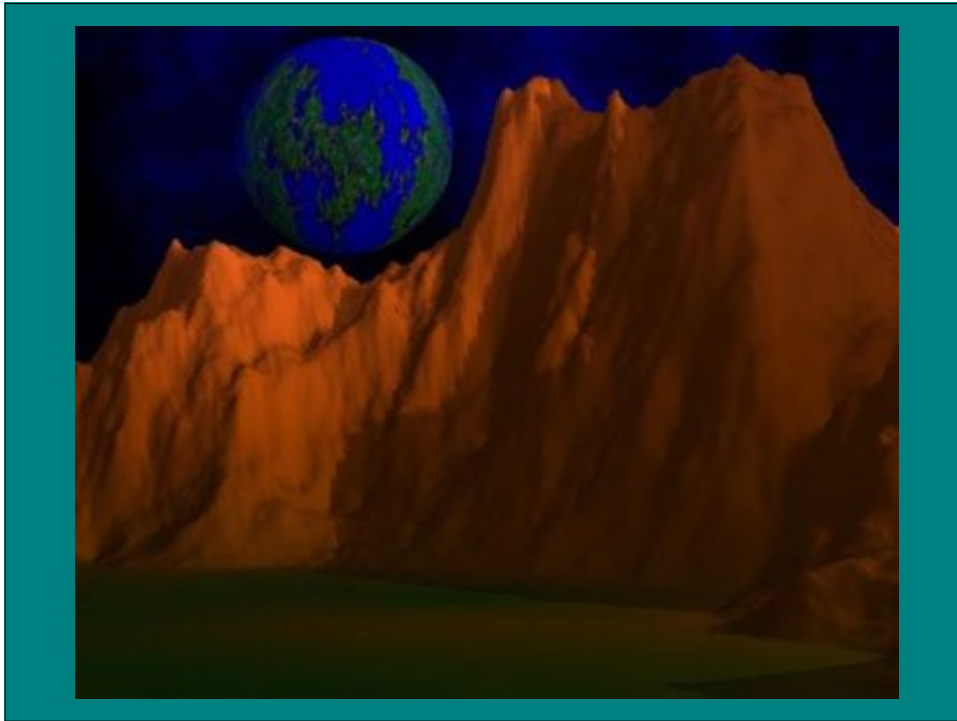
- Complex structure through self-similarity across scales
- Process: at each generation, replace each primitive with a self-similar sequence

Koch Curves

```
void Koch (float dir, float len, int n) {  
    float rads = 0.017453293;  
    if (n > 0) {  
        Koch(dir, len/3.0, n-1);  
        Dir = dir+60;  
        Koch(dir, len/3.0, n-1);  
        Dir = dir-120;  
        Koch(dir, len/3.0, n-1);  
        Dir = dir+60;  
        Koch(dir, len/3.0, n-1);  
    }  
    else LineTo(len*cos(rads*dir),len*sin(rads*dir));  
}
```

Koch Curves





Grammar-based Modeling

- Use (mostly) context-free grammars (CFG) to specify structural change over generations
- Often used to simulate a biological growth process
 - Plants
 - Seashells

Context-Free Grammar

- A CFG $G=(V,T,S,P)$ where
 - V is a set of non-terminals
 - T is a set of terminals
 - S is the start symbol
 - P is a set of productions (rules) of the form:
 - $A \rightarrow x$, where $A \in V, x \in (V \cup T)^*$

Applying Grammar Rules

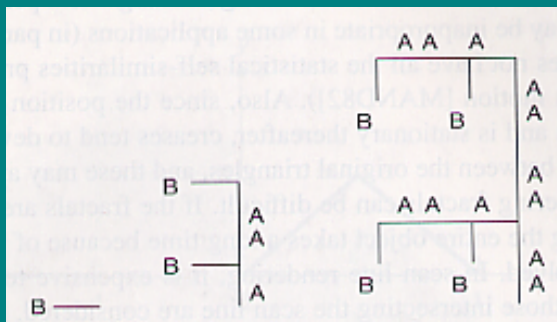
Rules

- $B \rightarrow A[B]AA[B]$
- $A \rightarrow AA$

- Branches to left

Strings

- 1: B
- 2: A[B]AA[B]
- 3: AA[A[BAA[B]]AAAA[A[B]AA[B]]]



Applying Grammar Rules

Rules

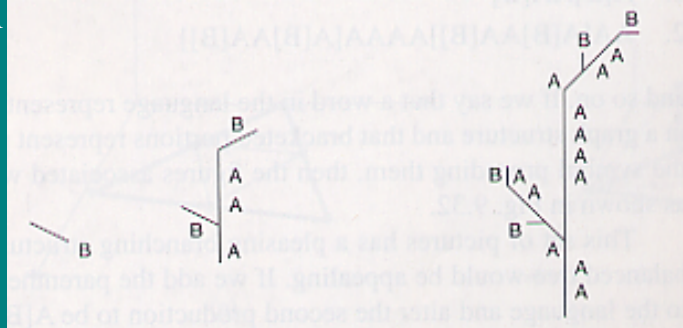
- $B \rightarrow A[B]AA(B)$
- $A \rightarrow AA$

Branch to

- Left []
- Right ()

Strings

- 1: B
- 2: A[B]AA(B)
- 3: AA[A[BAA(B)]AAAA(A[B]AA(B))

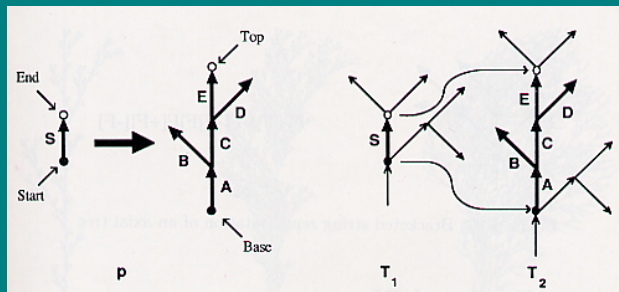


L-system Productions

$S \rightarrow A[+B]C[-D]E$

+ left angle

- right angle



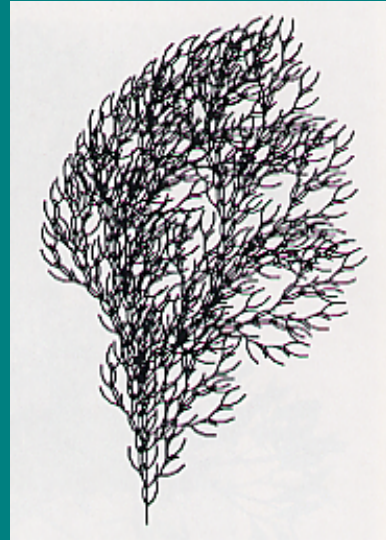
- $N = 7, a = 25.7^\circ$
- $S = X$
- Rules:
 $X \rightarrow F[+X][-X]FX$
 $F \rightarrow FF$



- $N = 5, a = 22.5^\circ$
- $S = X$
- Rules:
 $X \rightarrow F-[[X]+X]+F[+FX]-X$
 $F \rightarrow FF$



- $N = 4, a = 22.5^\circ$
- $S = F$
- Rules:
 $F \rightarrow FF-[F+F+F]+[+F-F-F]$

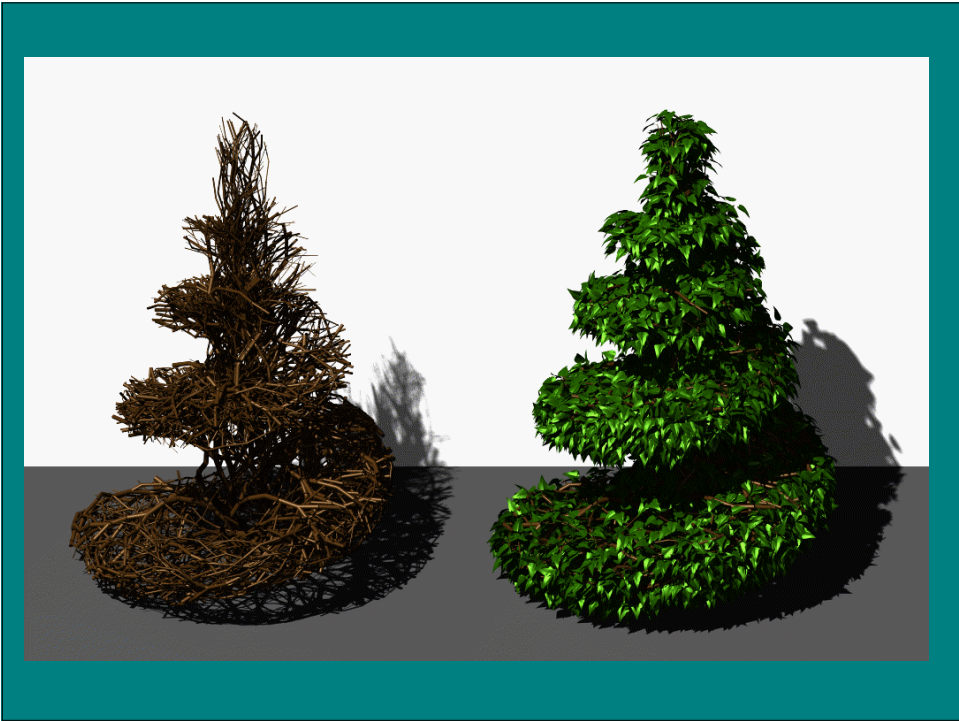
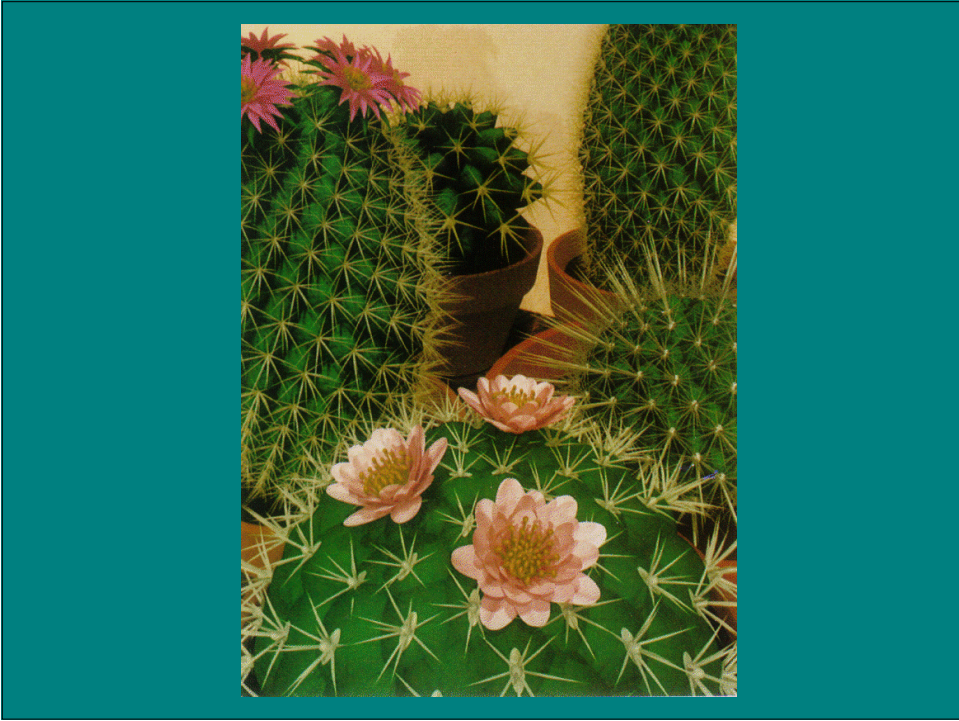


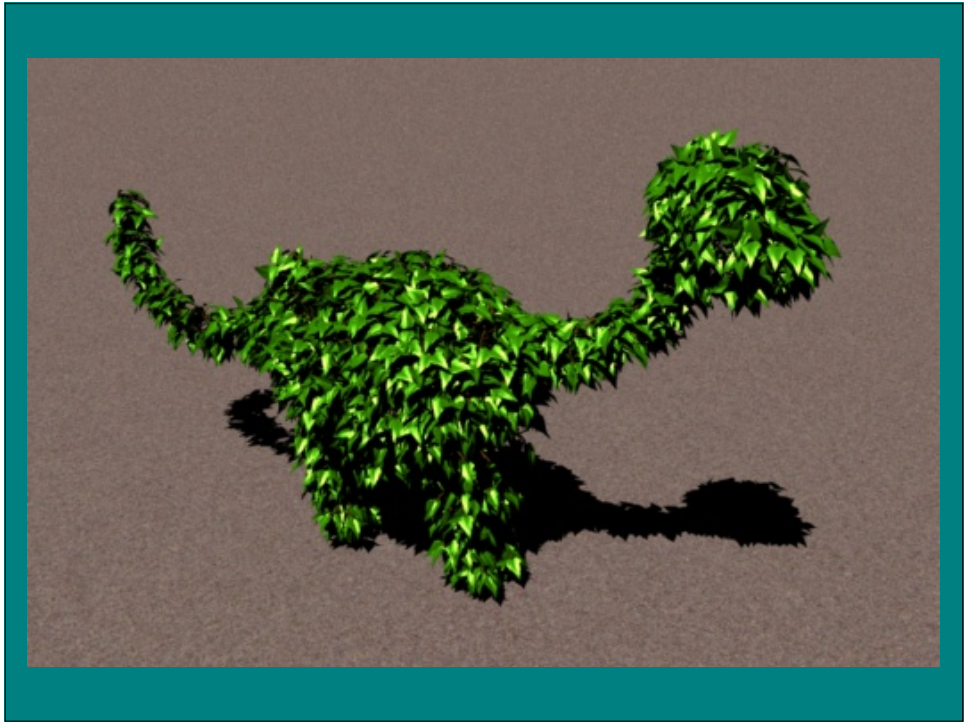
Additions

- 3D structure
- Randomness
- Leaves
- Flowers



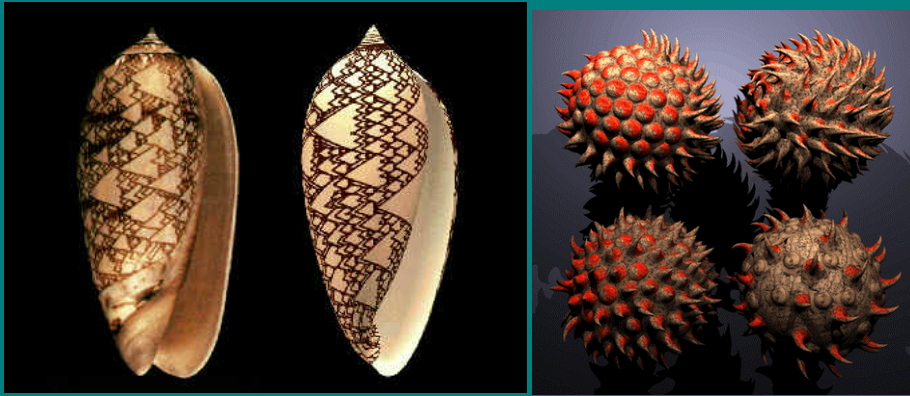






Biological Simulations

- Simulate developmental biological process

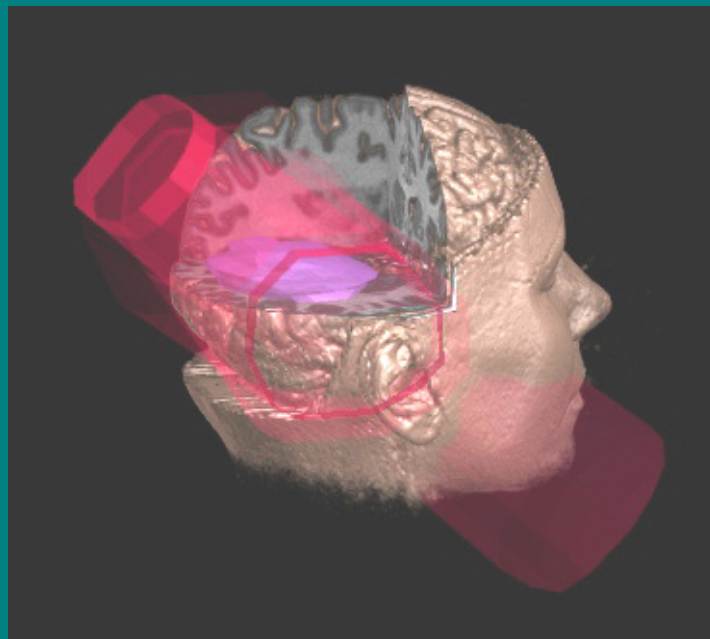


Reaction Diffusion



Visualization

- Derive model from data through abstraction process
- Examples
 - Isosurfaces of volume data
 - Fluid flow



Marc Levoy

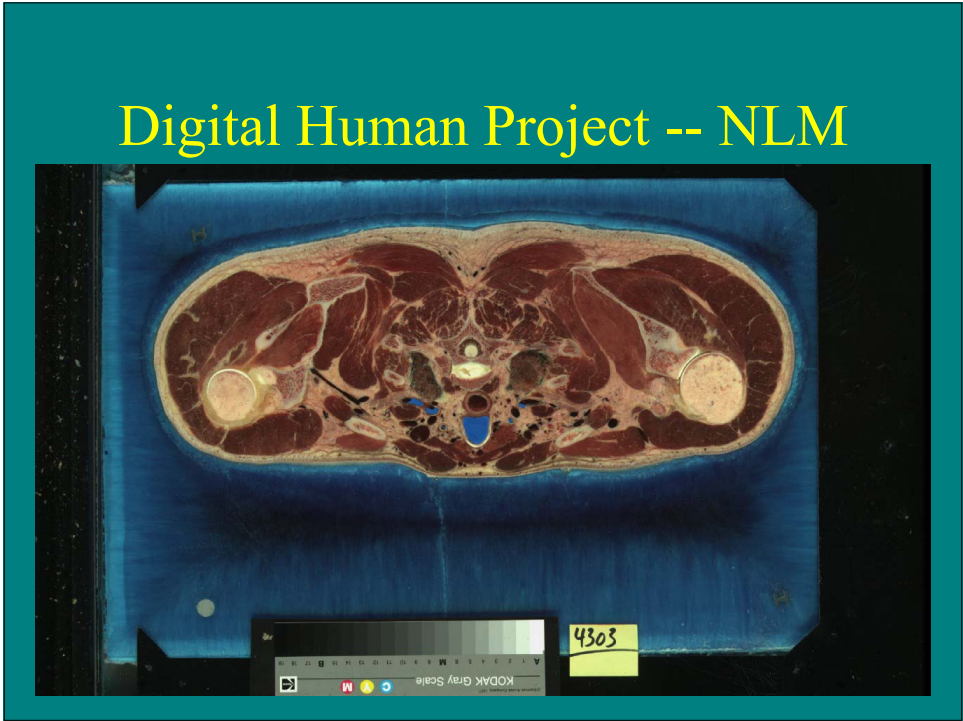


Image-based Rendering

- View Interpolation
- Plenoptic Rendering
- Lumigraph/ Light Fields
- Layered Depth Images
- Synthetic/Real Objects
- Points as Primitives

