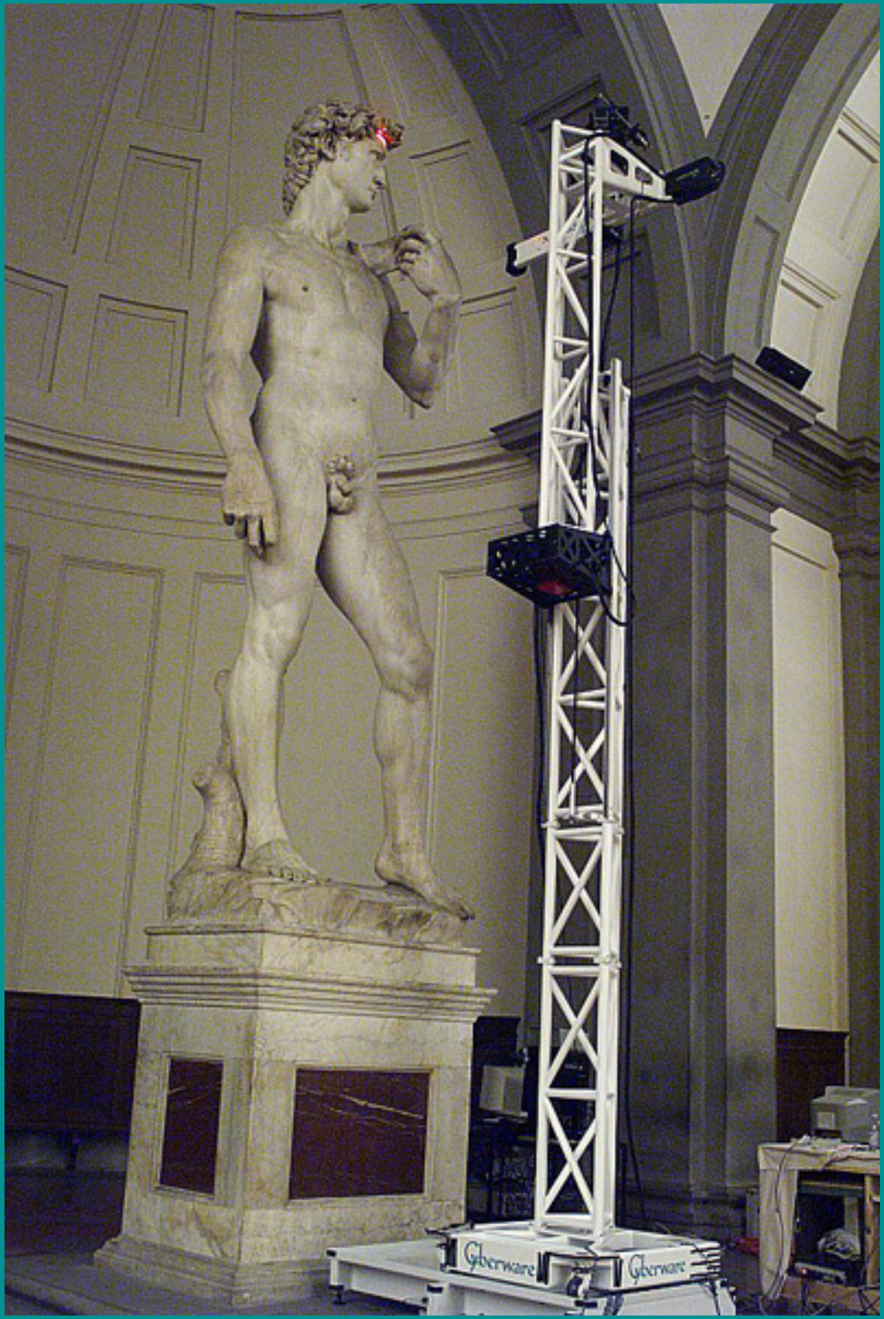
Modeling Approaches from Dr. Penny Rheigans graphics lecture

- 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

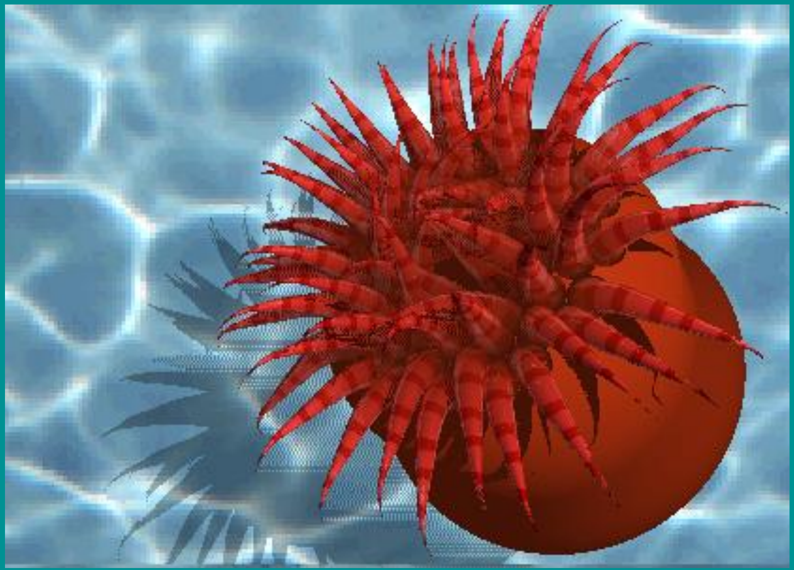
- Describe visual attributes through some function, usually defined over space
  - Shape
  - Density
  - Color
  - Texture

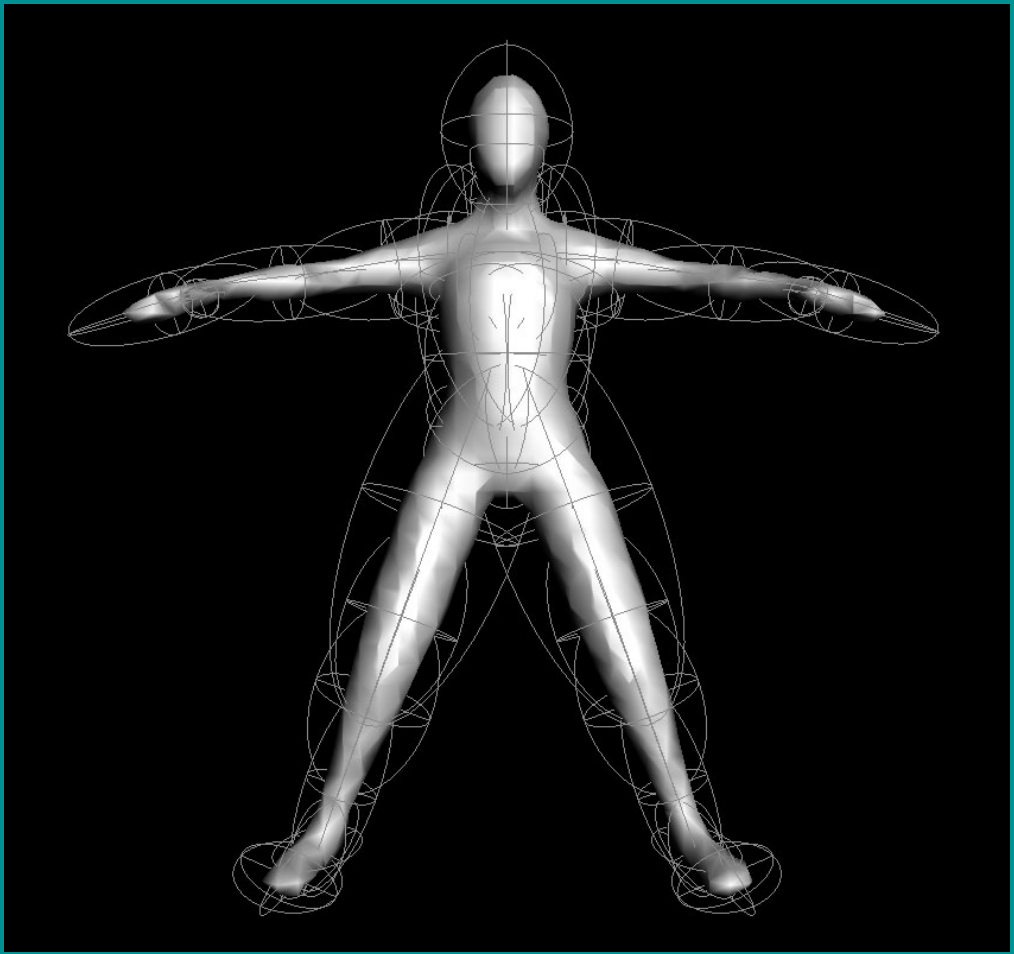
Procedural Approaches

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

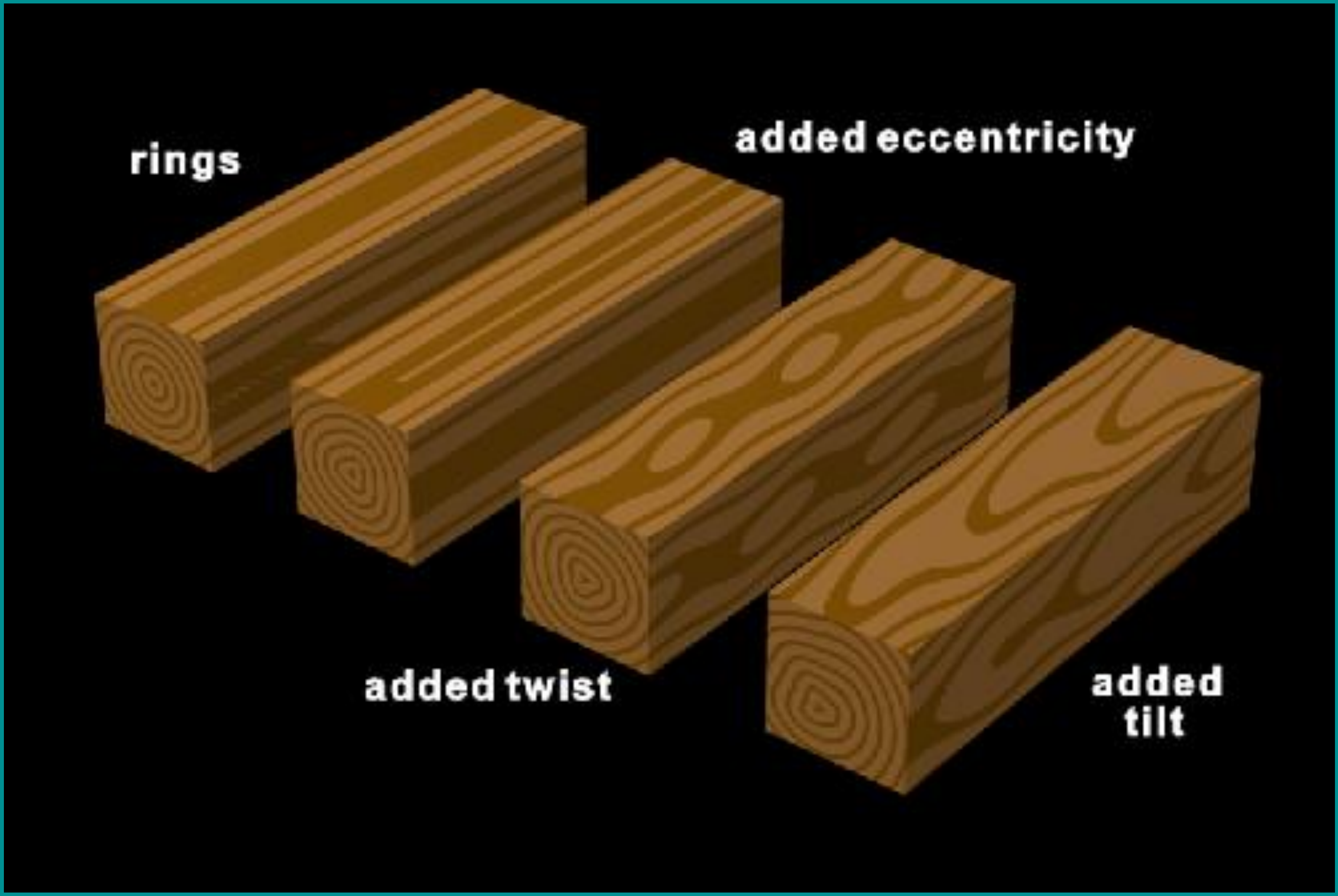
Implicit Functions

- Model as sum of implicit functions
- Surface at threshold





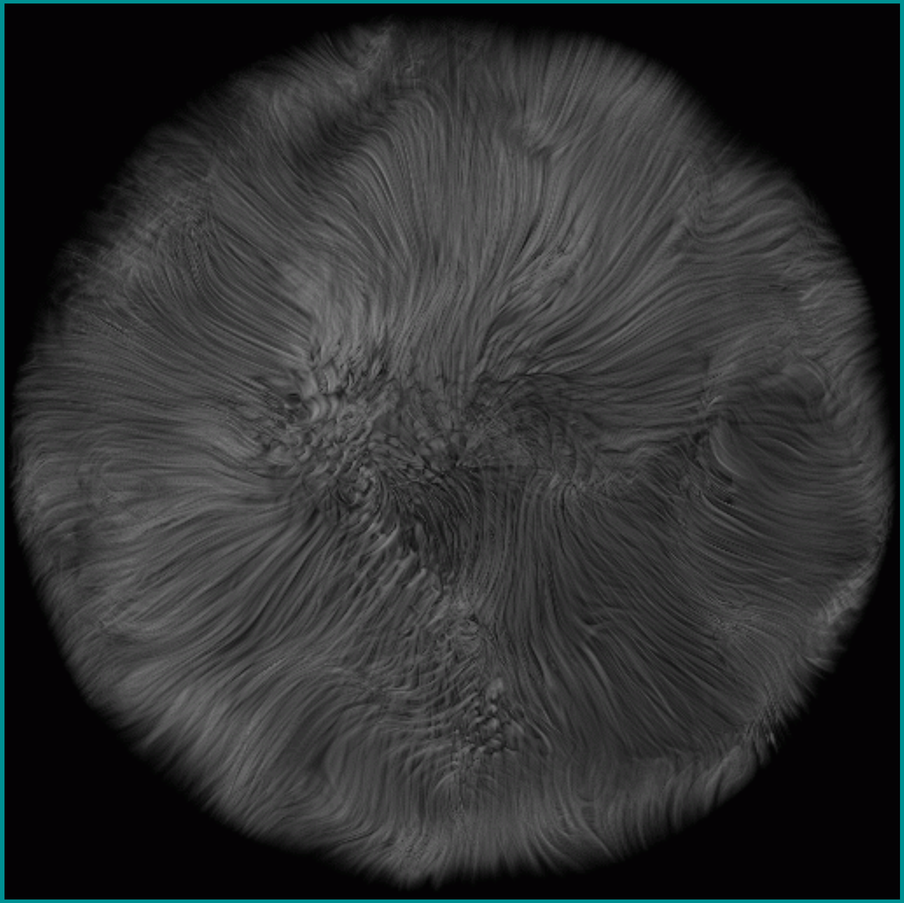
3D Solid Texture

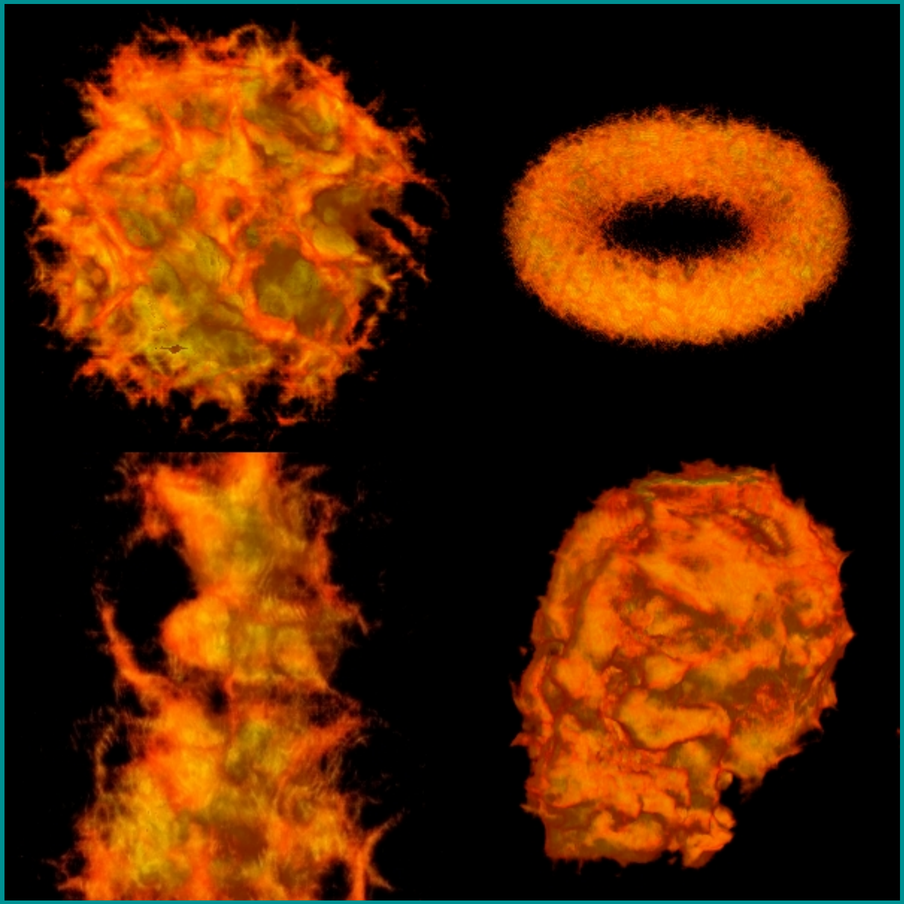


3D Solid Texture

Hypertexture

- Add noise or turbulence to functions





Hypertexture

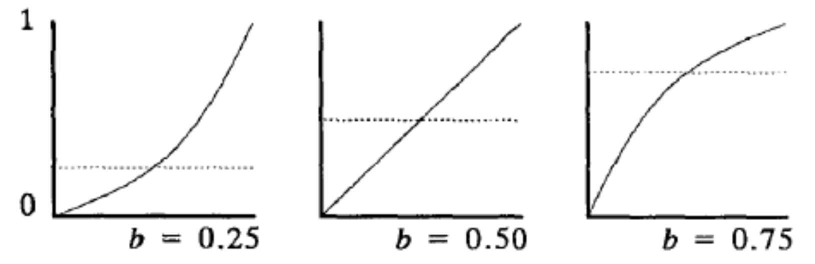
- Ken Perlin and Eric Hoffert, Hypertexture, SIGGRAPH ‘89.
- Extend 3D procedural noise textures to include opacity component to create volume models
  - object density function D(x)
  - Density modulation function (DMF) f_i

Boolean Operations

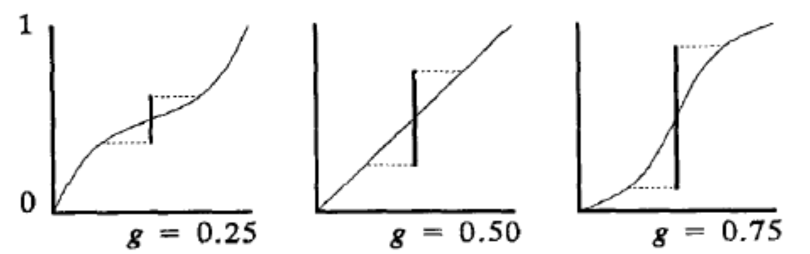
- Intersection
- Complement
- Difference
- Union

Base DMFs

- Bias
- Gain
- Noise
- Turbulence
- Arithmetic functions



Different *bias* curves



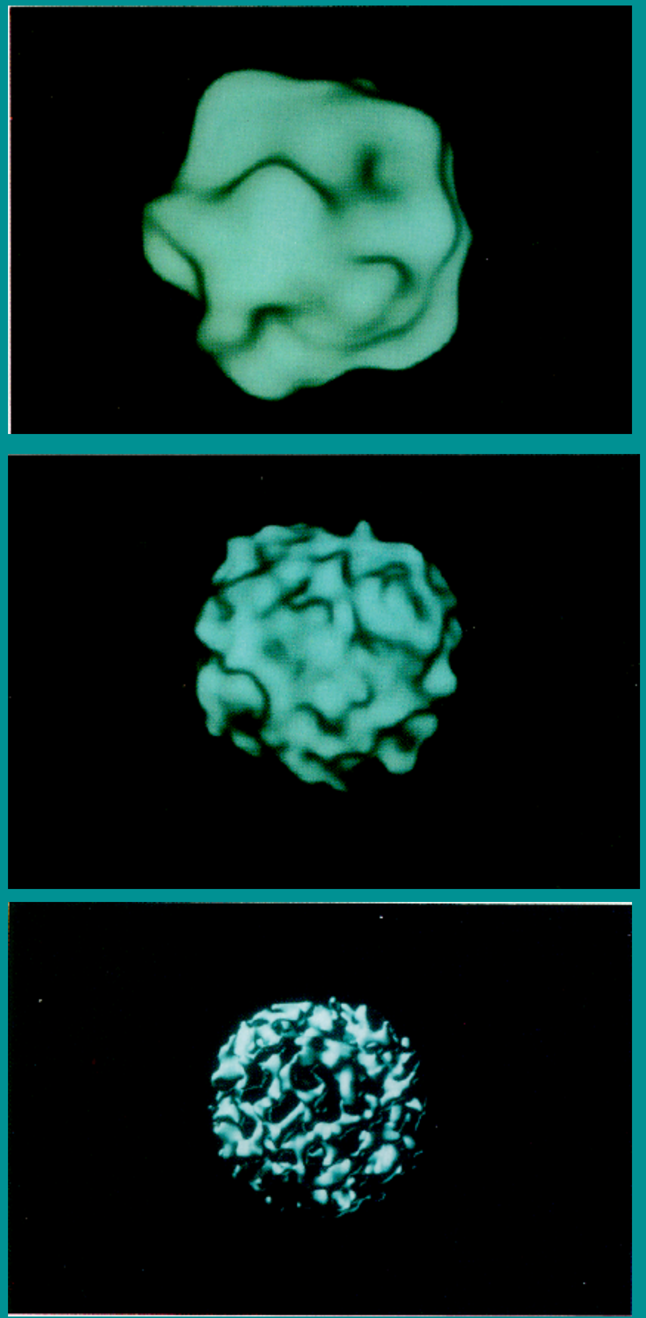
Different *gain* curves

Basic Noise

- Basic noisy sphere

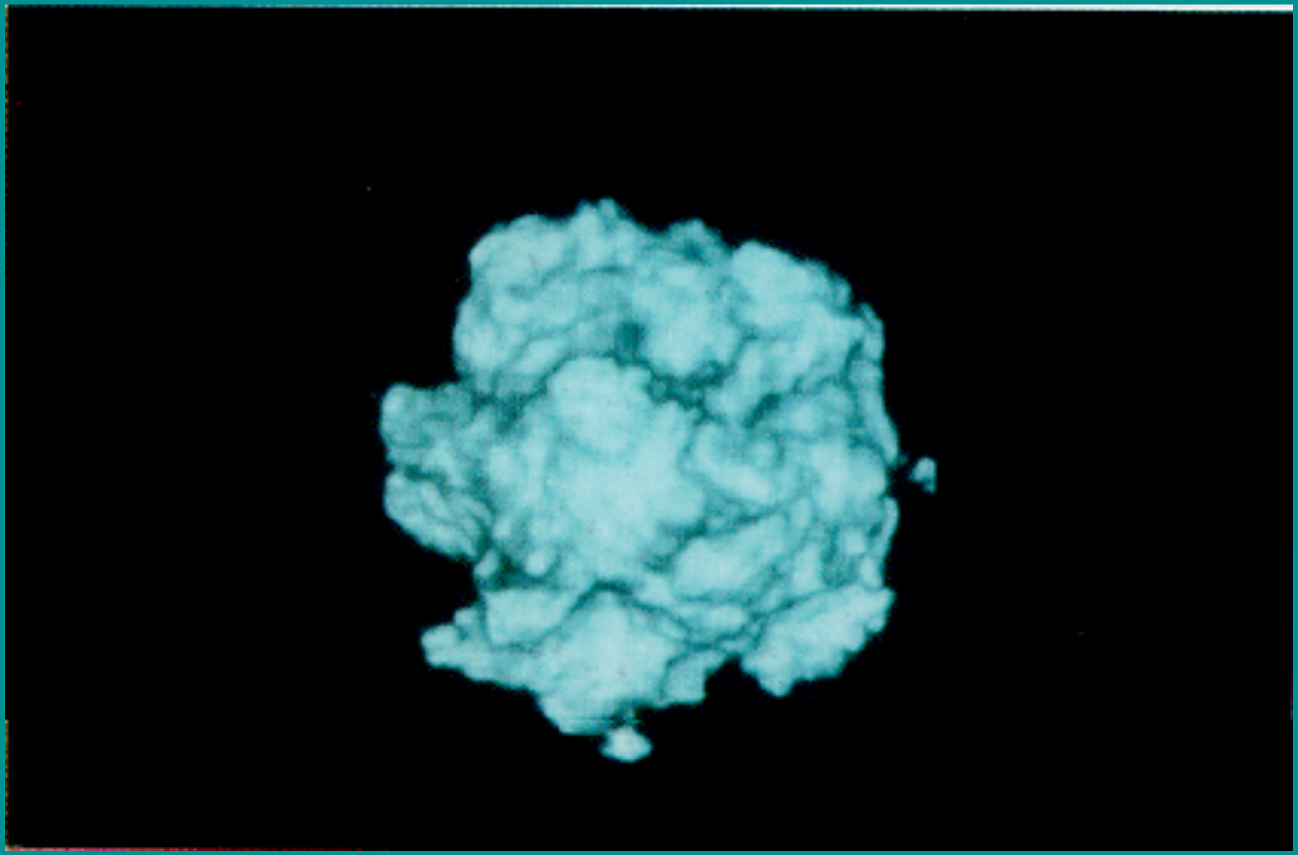
$$D(\mathbf{x}) = sphere(\mathbf{x}\,(1 + \frac{1}{f} noise(f\mathbf{x})))$$

- Vary
  - Frequency
  - Amplitude



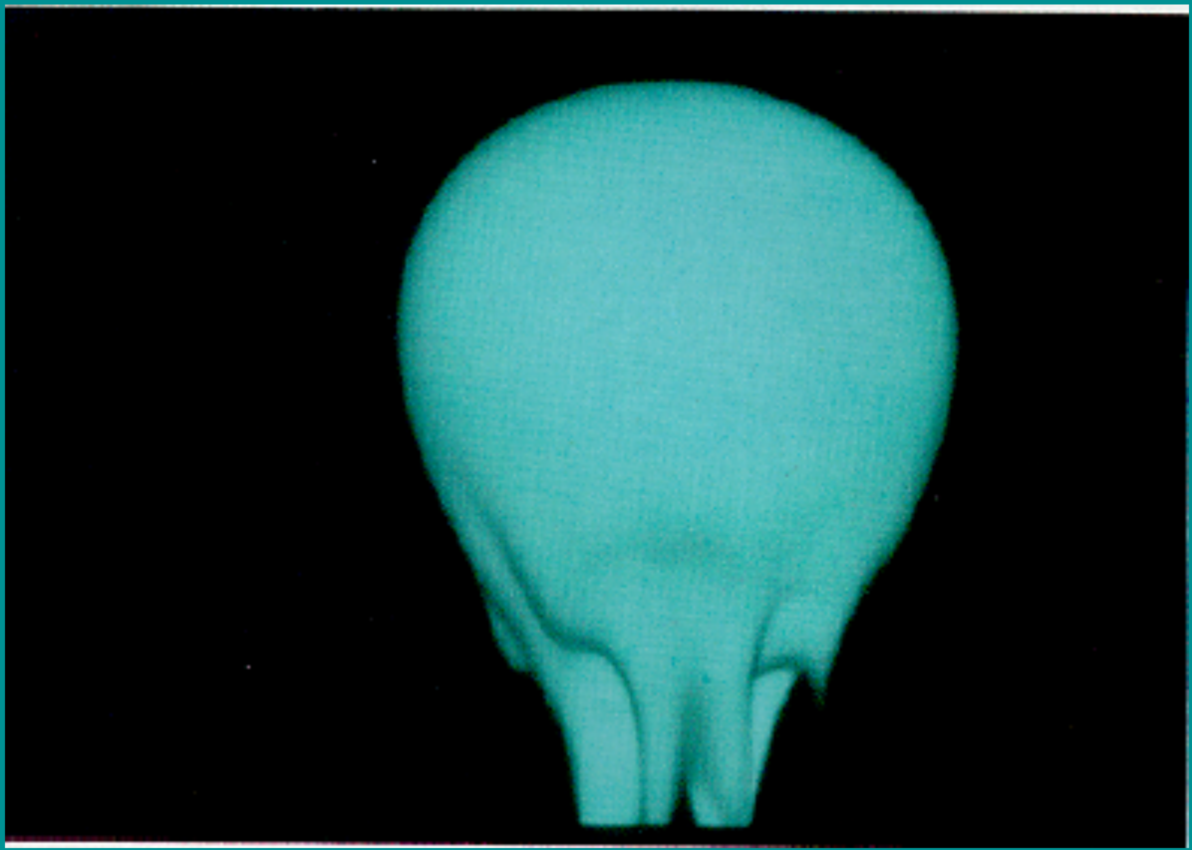
Turblence

- Mix different frequencies of noise



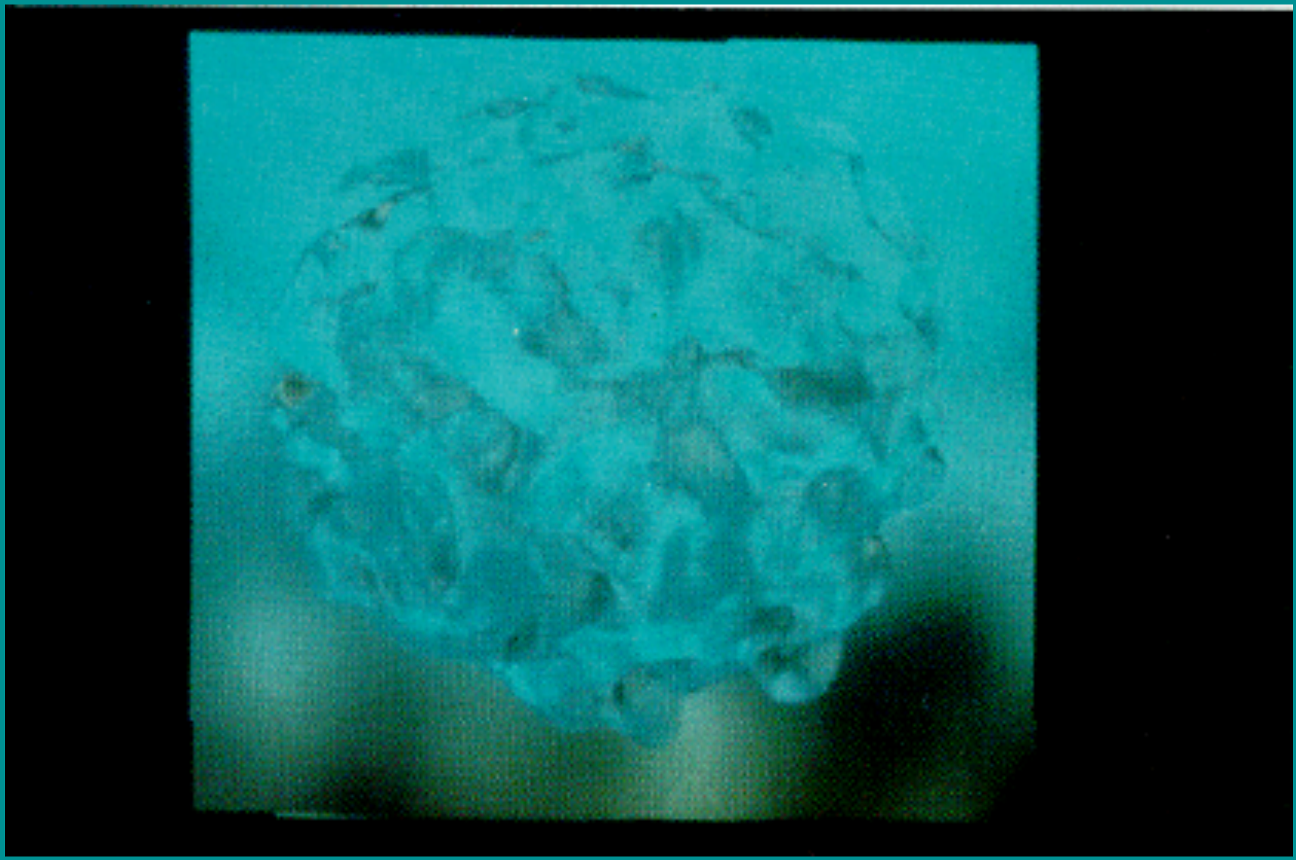
Shaped Noise

- Vary only single component



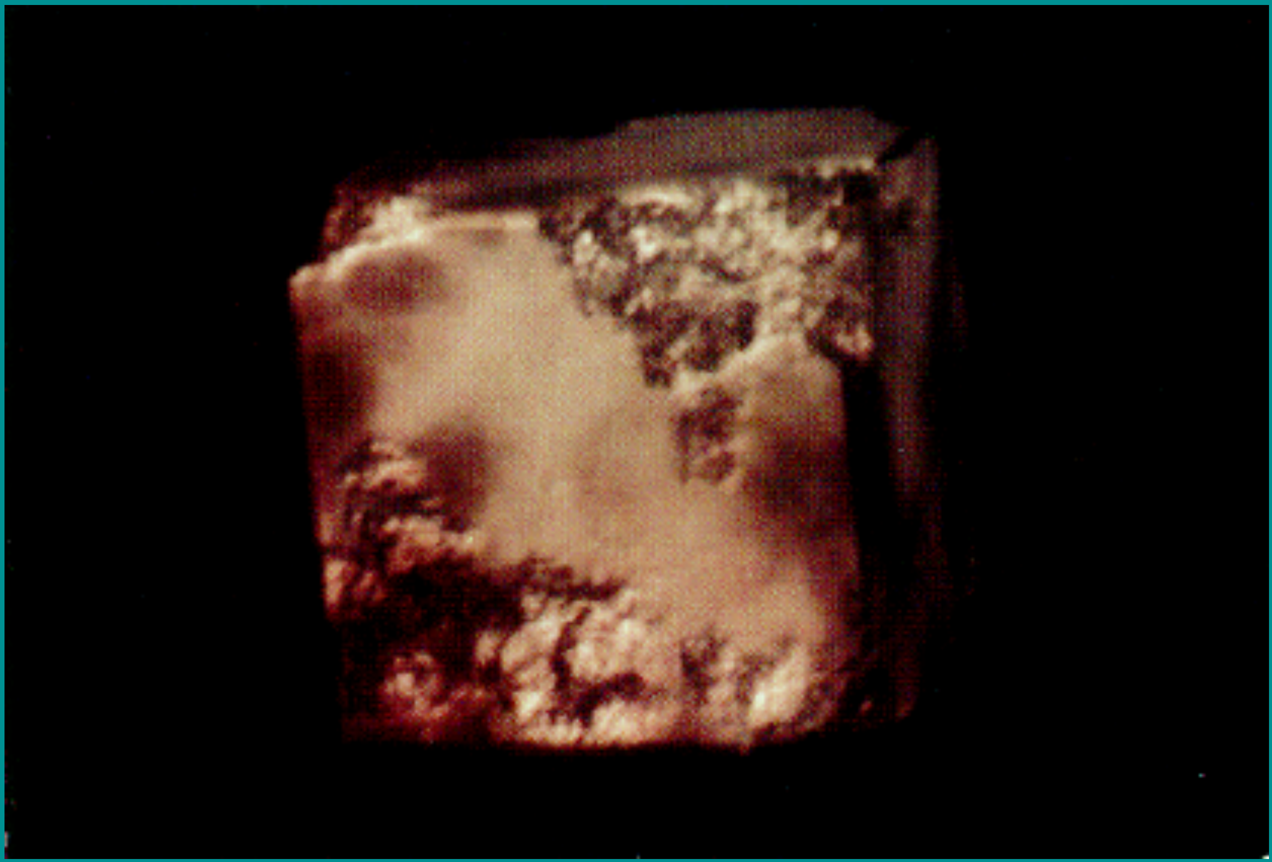
Transparency

- Refractive Hypertexture



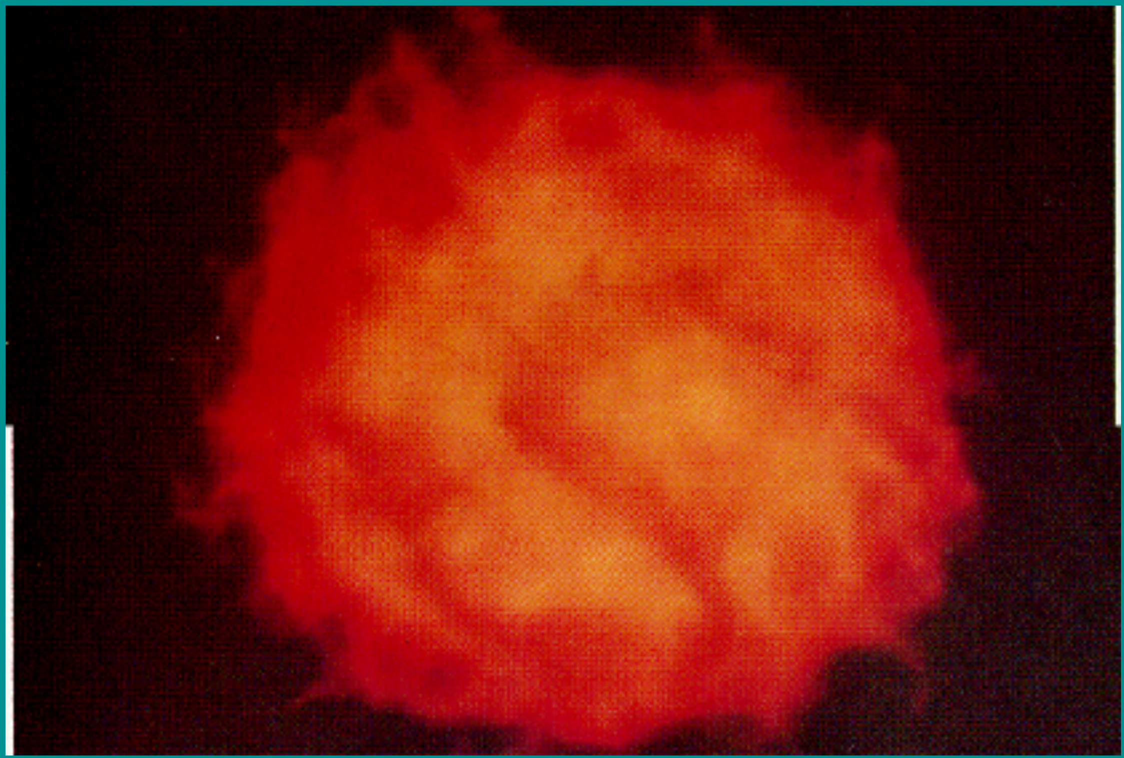
Erosion

- Boolean intersection of fractal sphere with cube



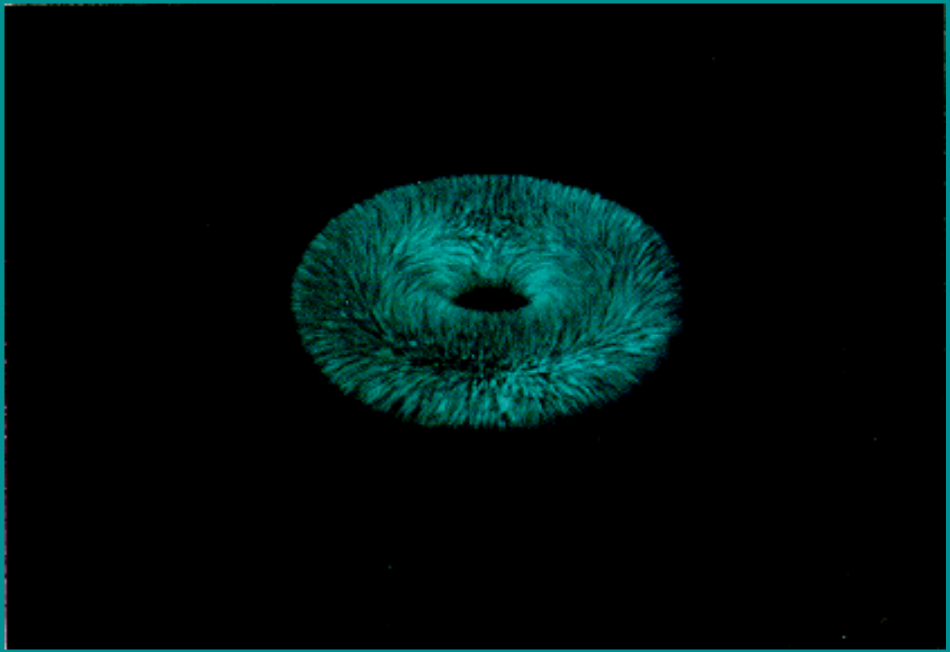
Fire

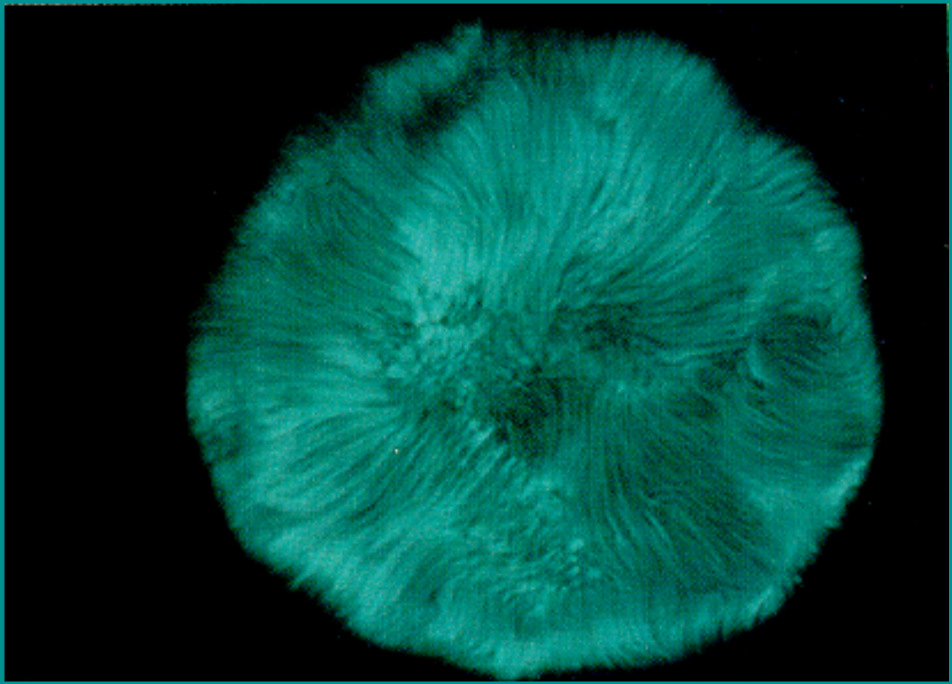
- Density func: $D(\mathbf{x}) = sphere(\mathbf{x}\,(1 + turbulence(\mathbf{x})))$
- Colormap

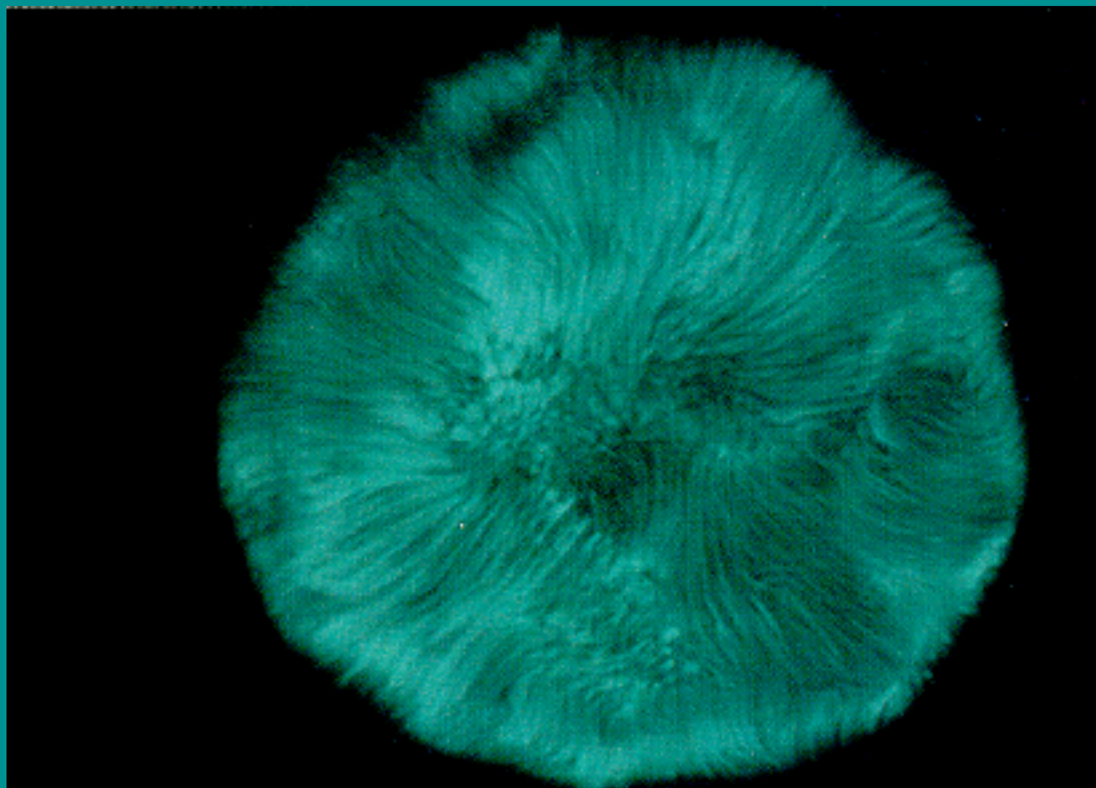


Fur

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

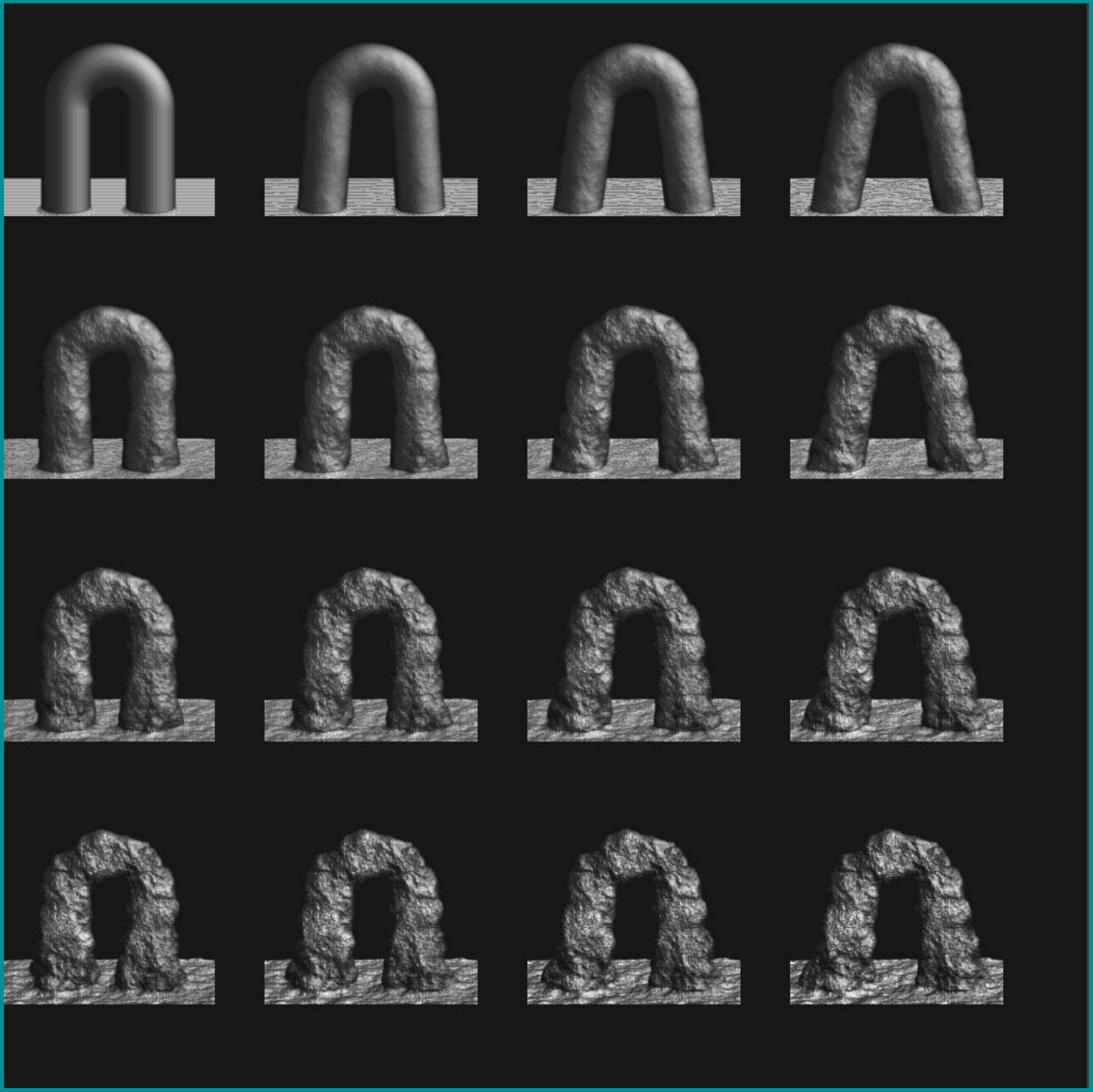


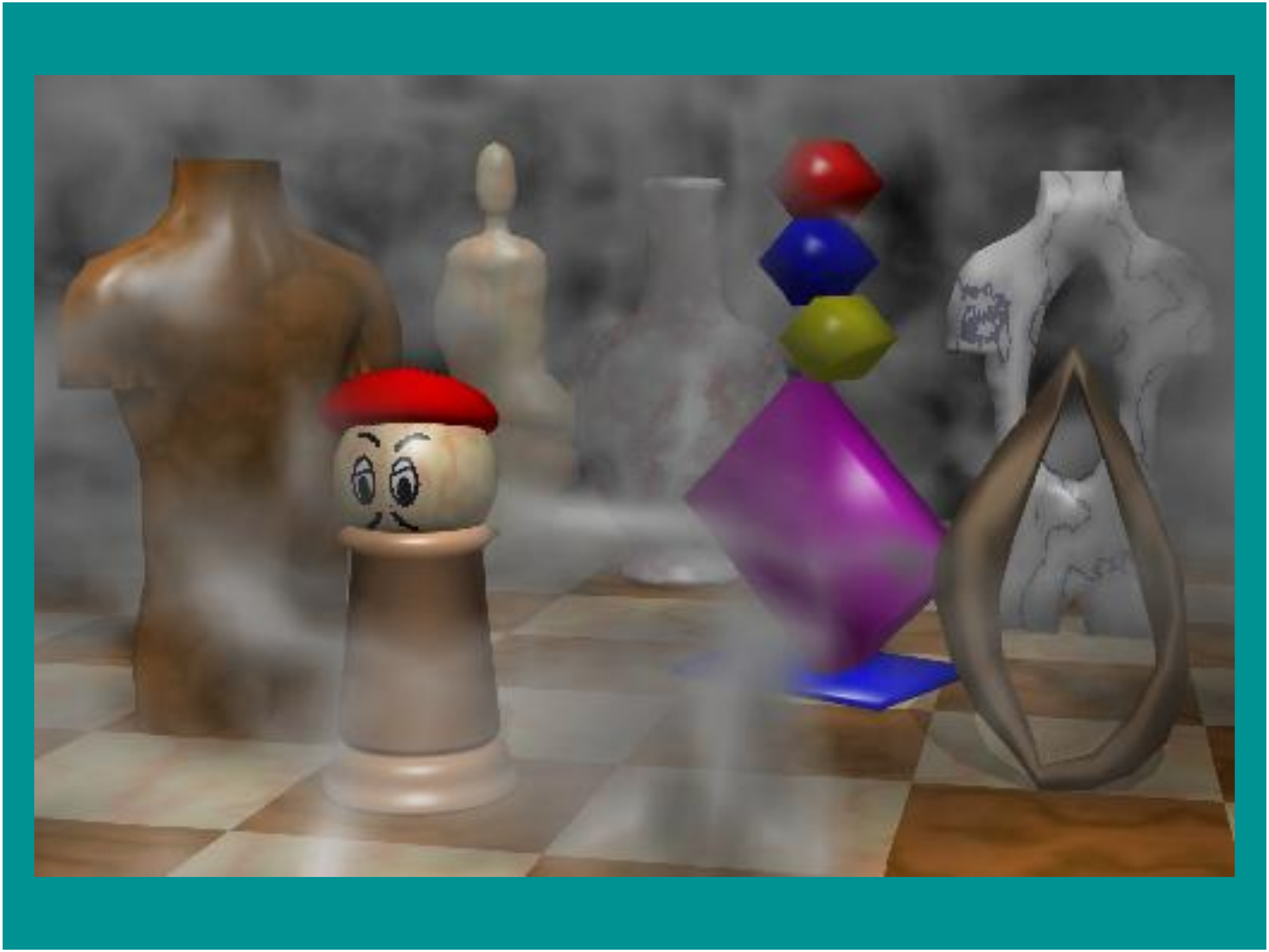


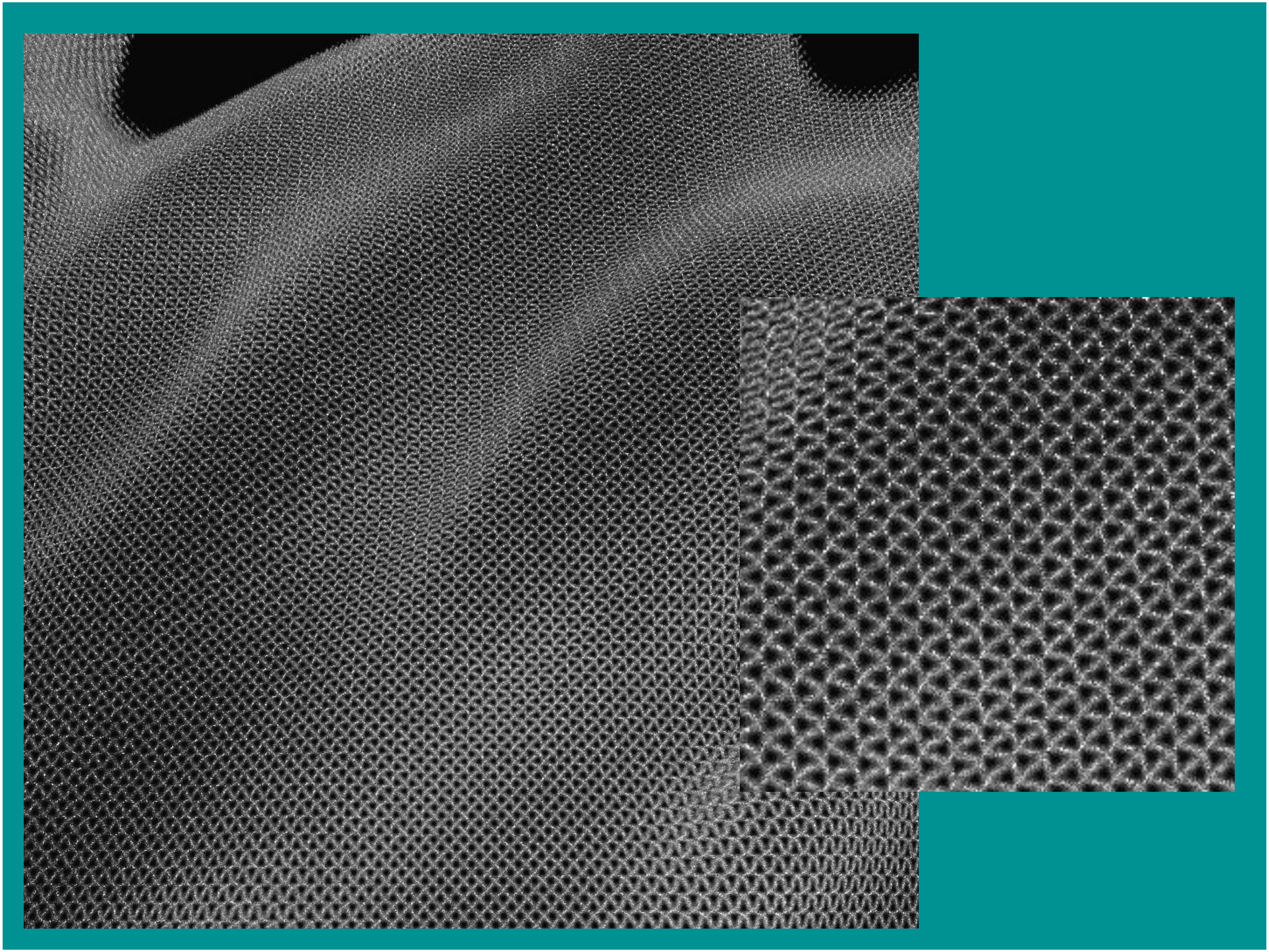


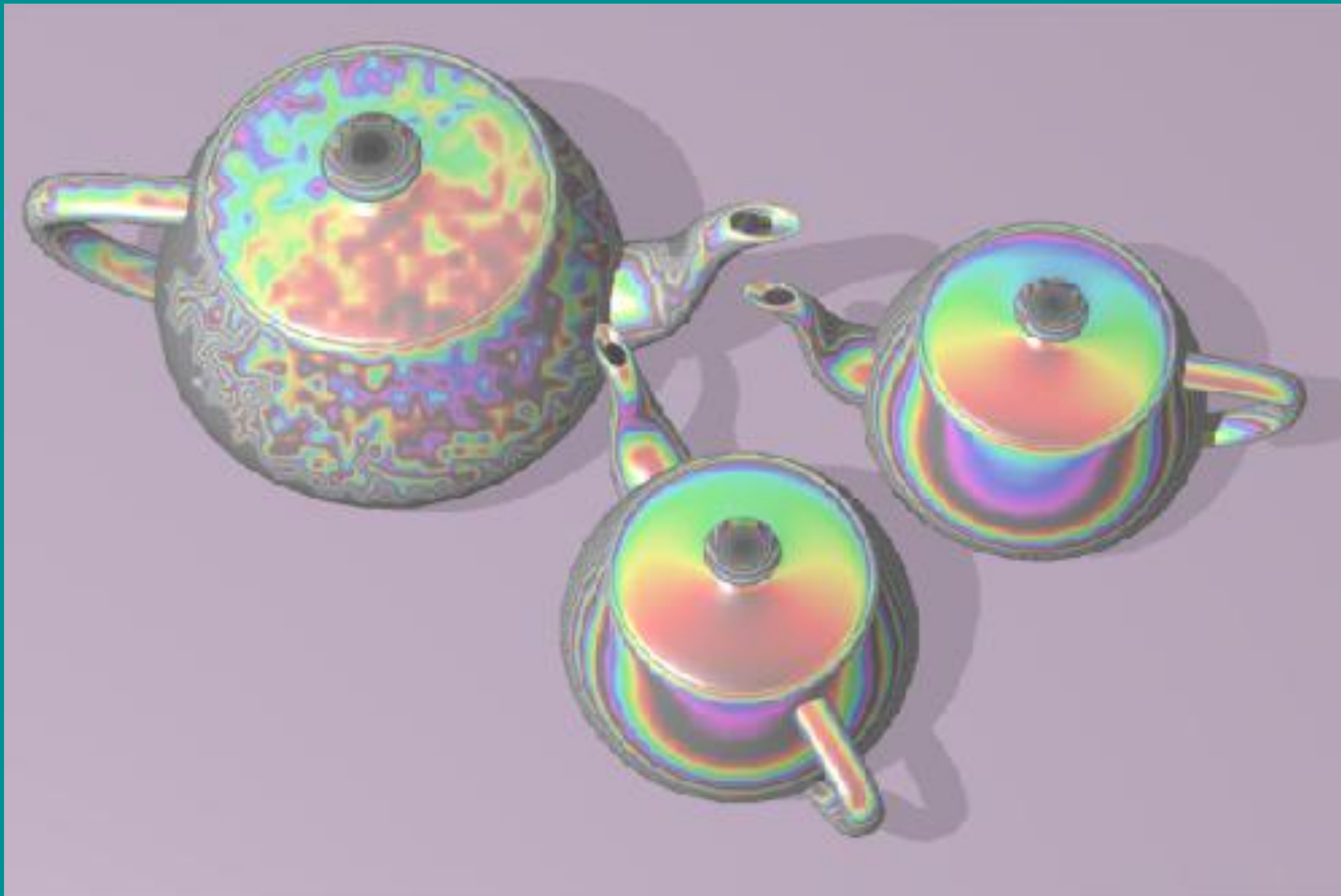
Noisy Things

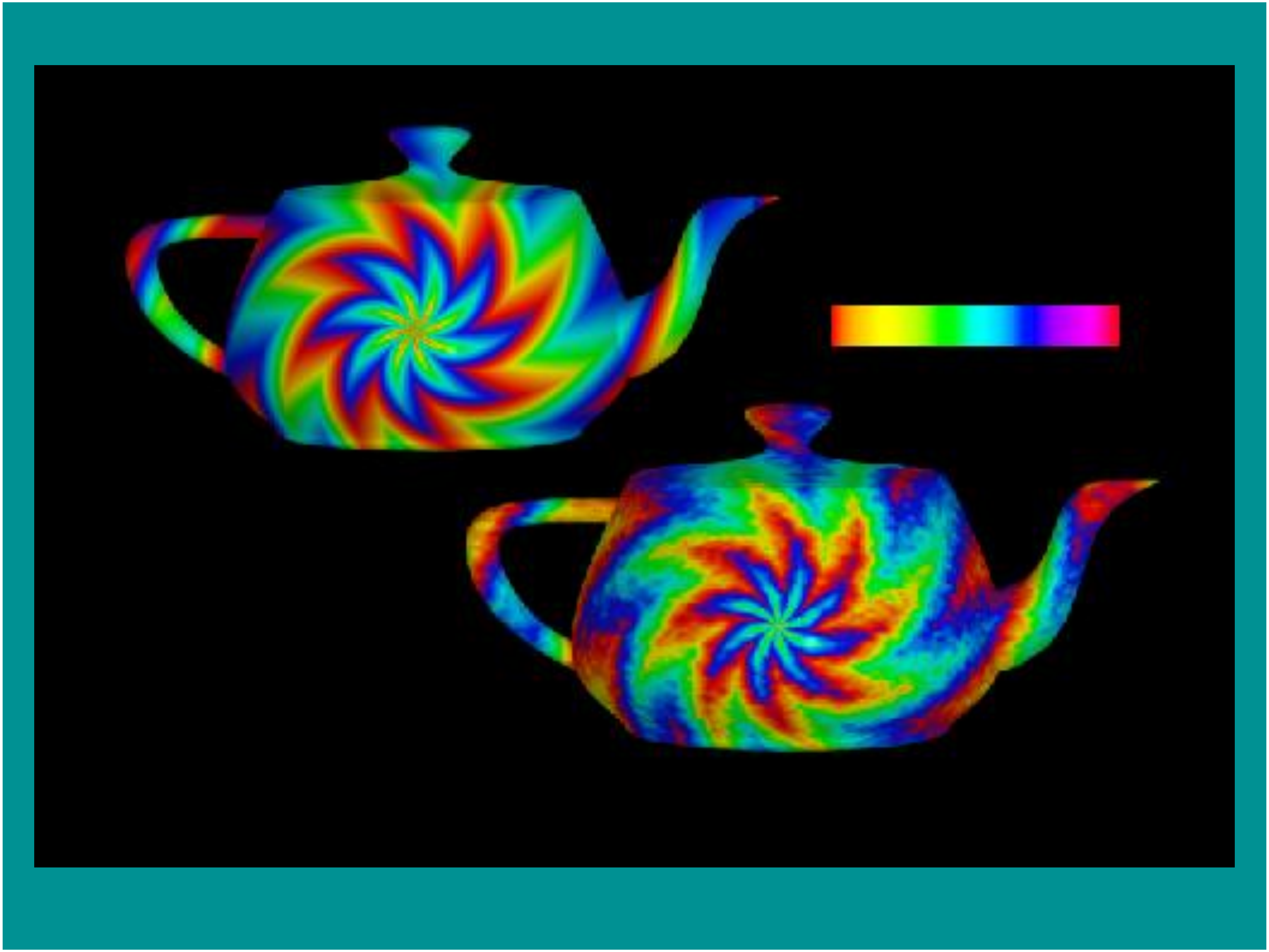
- Color
- Specularity
- Opacity/Density
- Normals
- Displacements
- Index of Refraction
- Procedural Texture Parameters

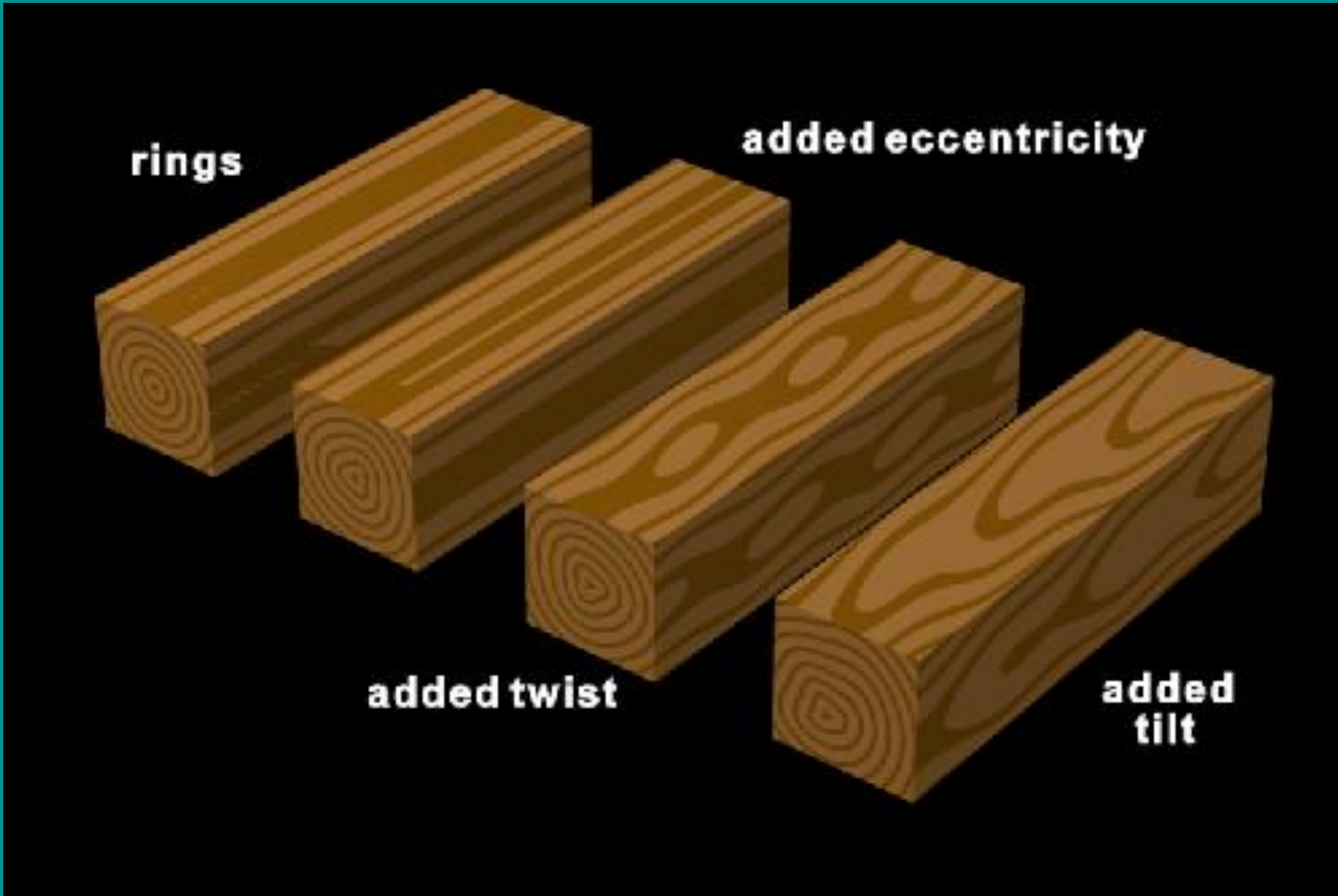











rings
added eccentricity
added twist
added
tilt

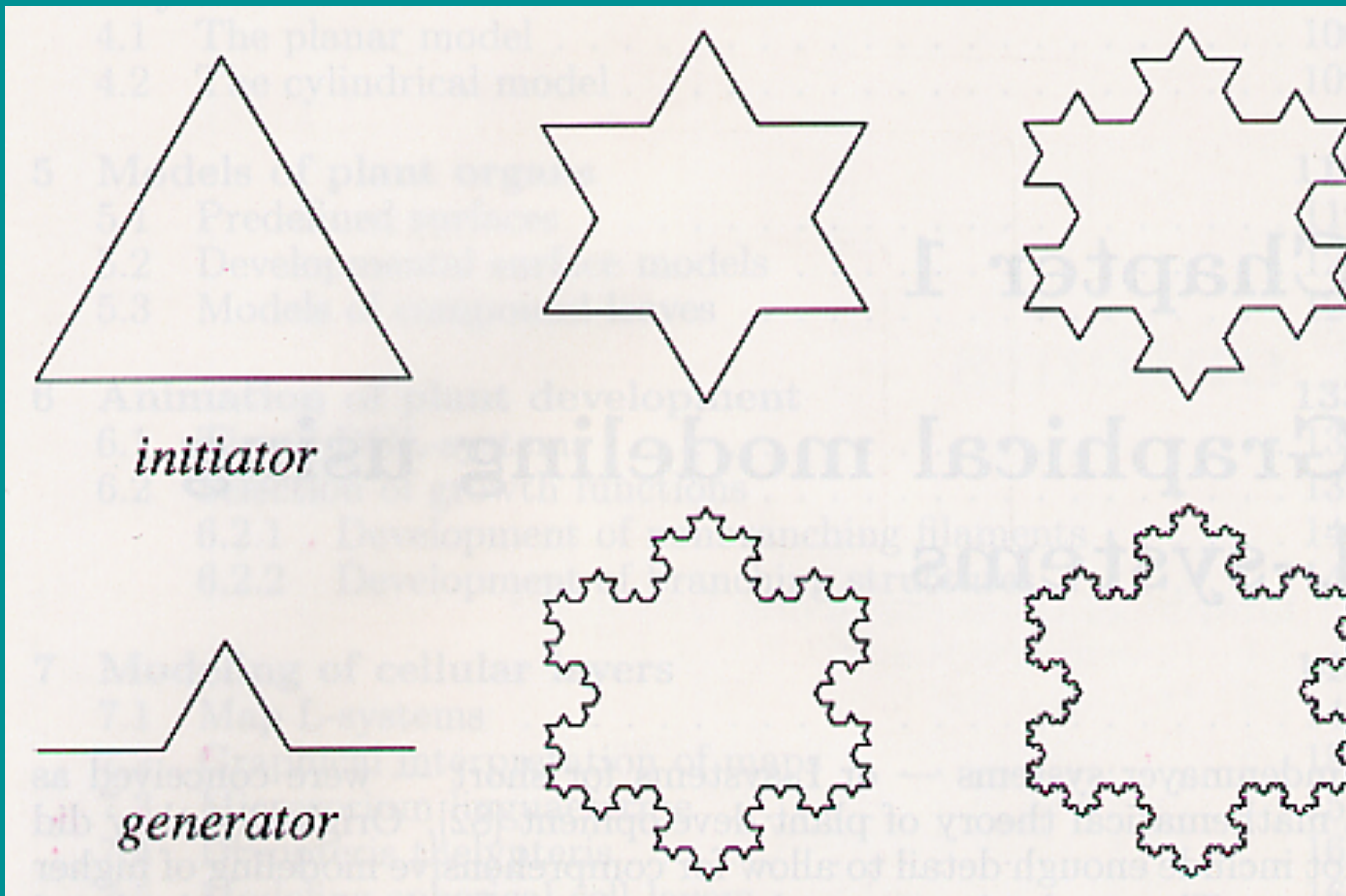
Fractals

- 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

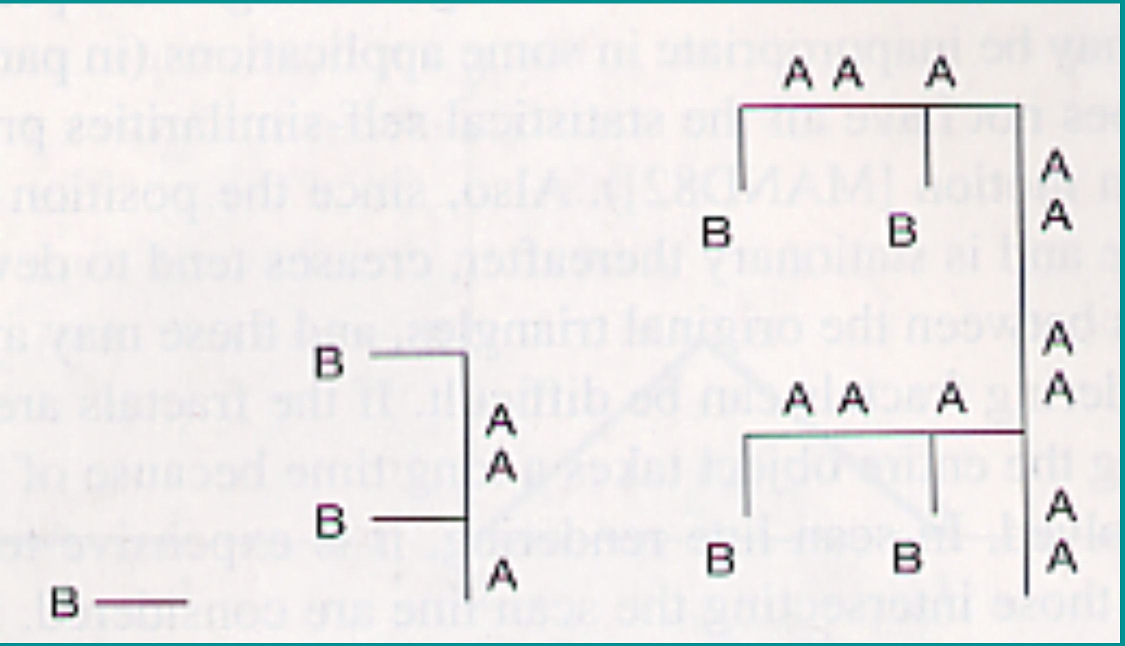
- A CFG $G=(V,T,S,P)$ where
  - V is a set of non-terminals
  - T is a set of terminals
  - S is the start symbol
  - P is a set of productions (rules) of the form:
    - $A \rightarrow x$, where $A \in V$, $x \in (V \cup T)^*$

Applying Grammar Rules

Rules

- B → A[B]AA[B]
- A → AA

- Branches to left



Strings

1: B

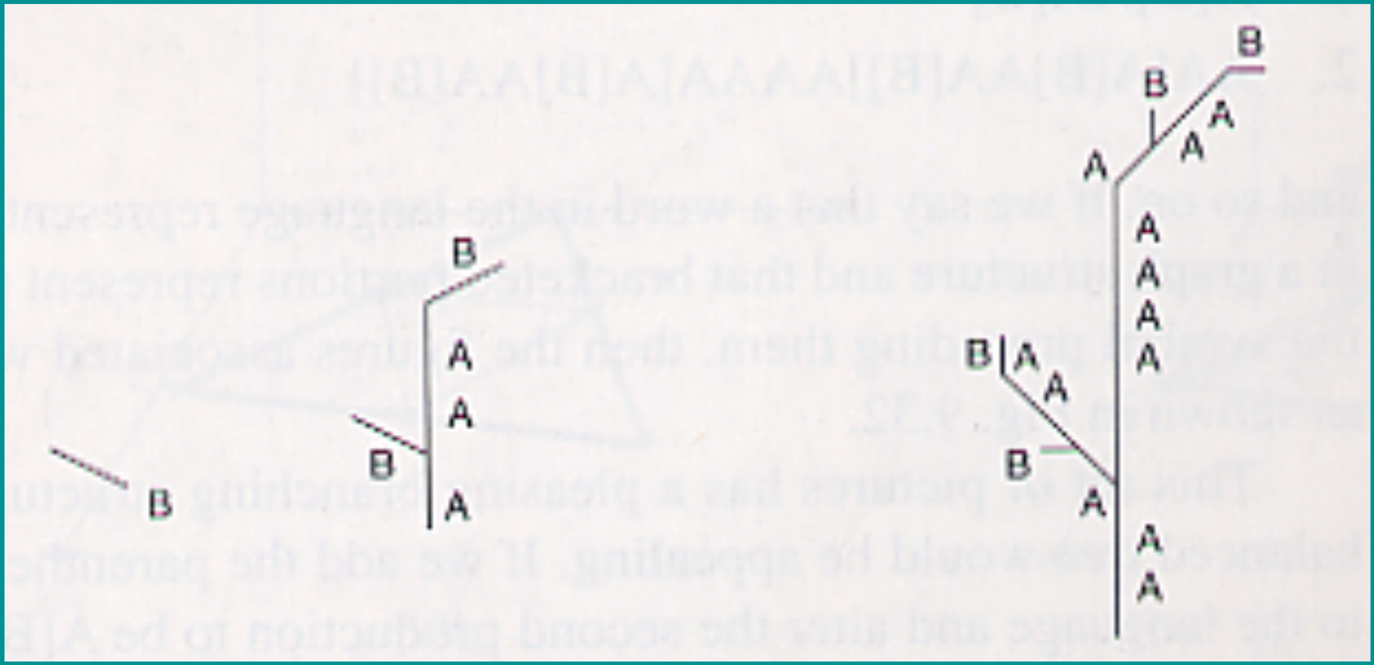
2: A[B]AA[B]

3: AA[A[BAA[B]]AAAA[A[B]AA[B]]

Applying Grammar Rules

Rules

- B → A[B]AA(B)
- A → AA



Branch to

- Left []
- Right ()

Strings

1: B

2: A[B]AA(B)

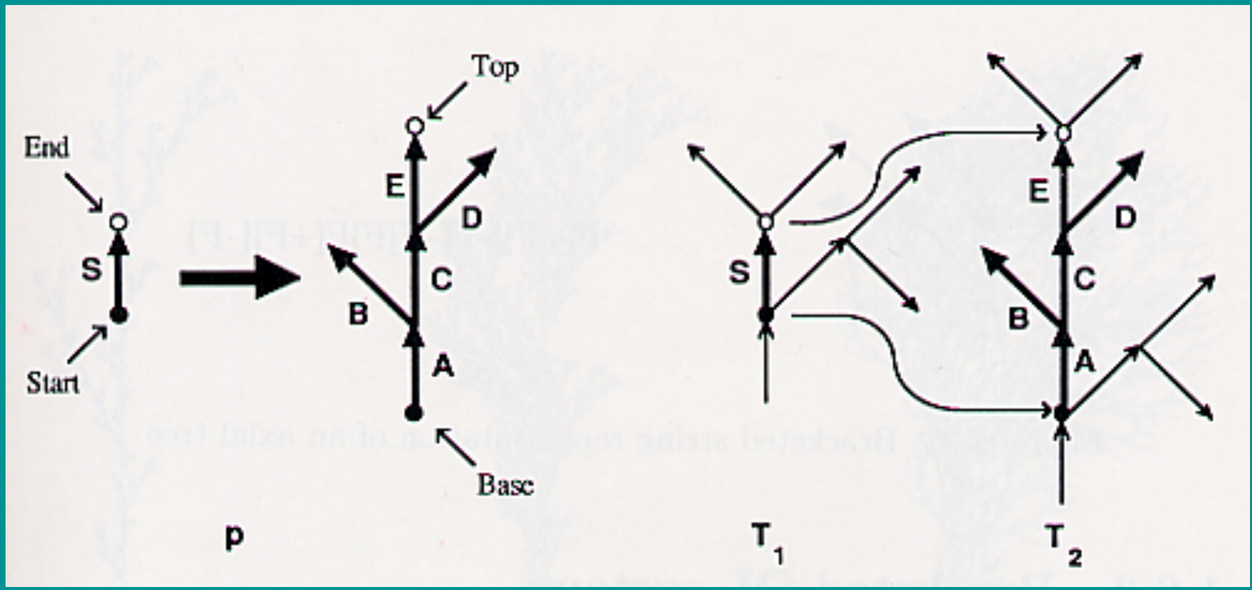
3: AA[A[BAA(B)]AAAA(A[B]AA(B))

L-system Productions

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

+ left angle

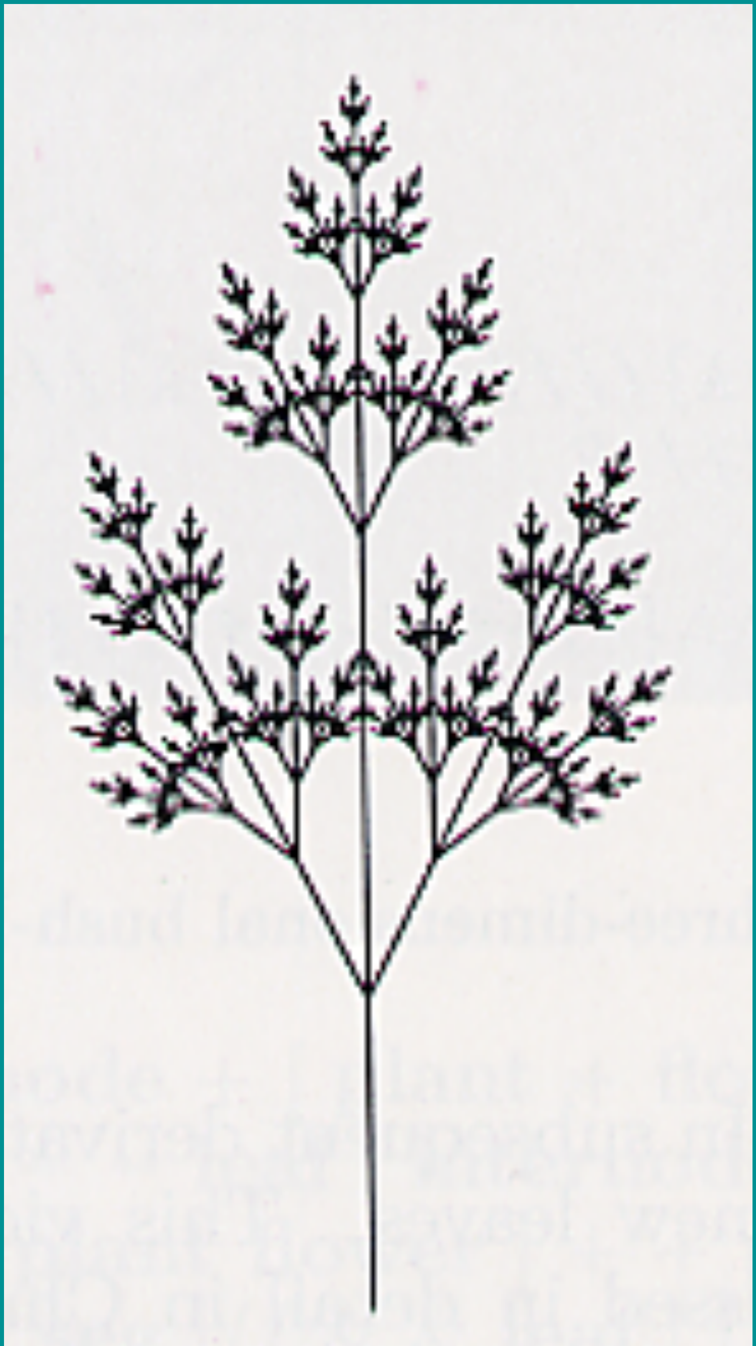
- right angle



- N = 7, a = 25.7°
- S = X
- Rules:

  X→ F[+X][-X]FX

  F → FF



- N = 5, a = 22.5°
- S = X
- Rules:

  $X \rightarrow F-[[X]+X]+F[+FX]-X$

  $F \rightarrow FF$

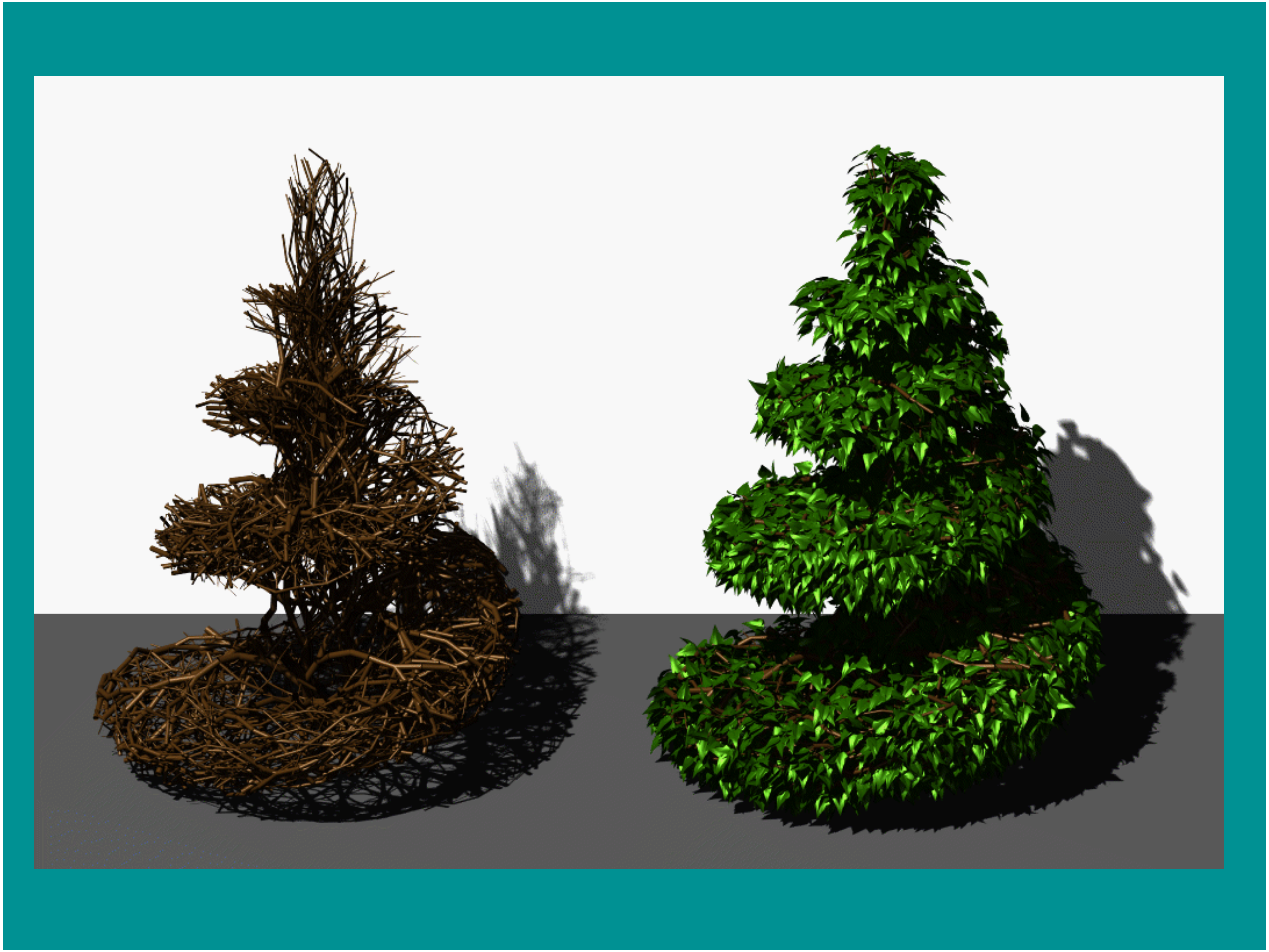
- $N = 4$, $a = 22.5°$
- $S = F$
- Rules:

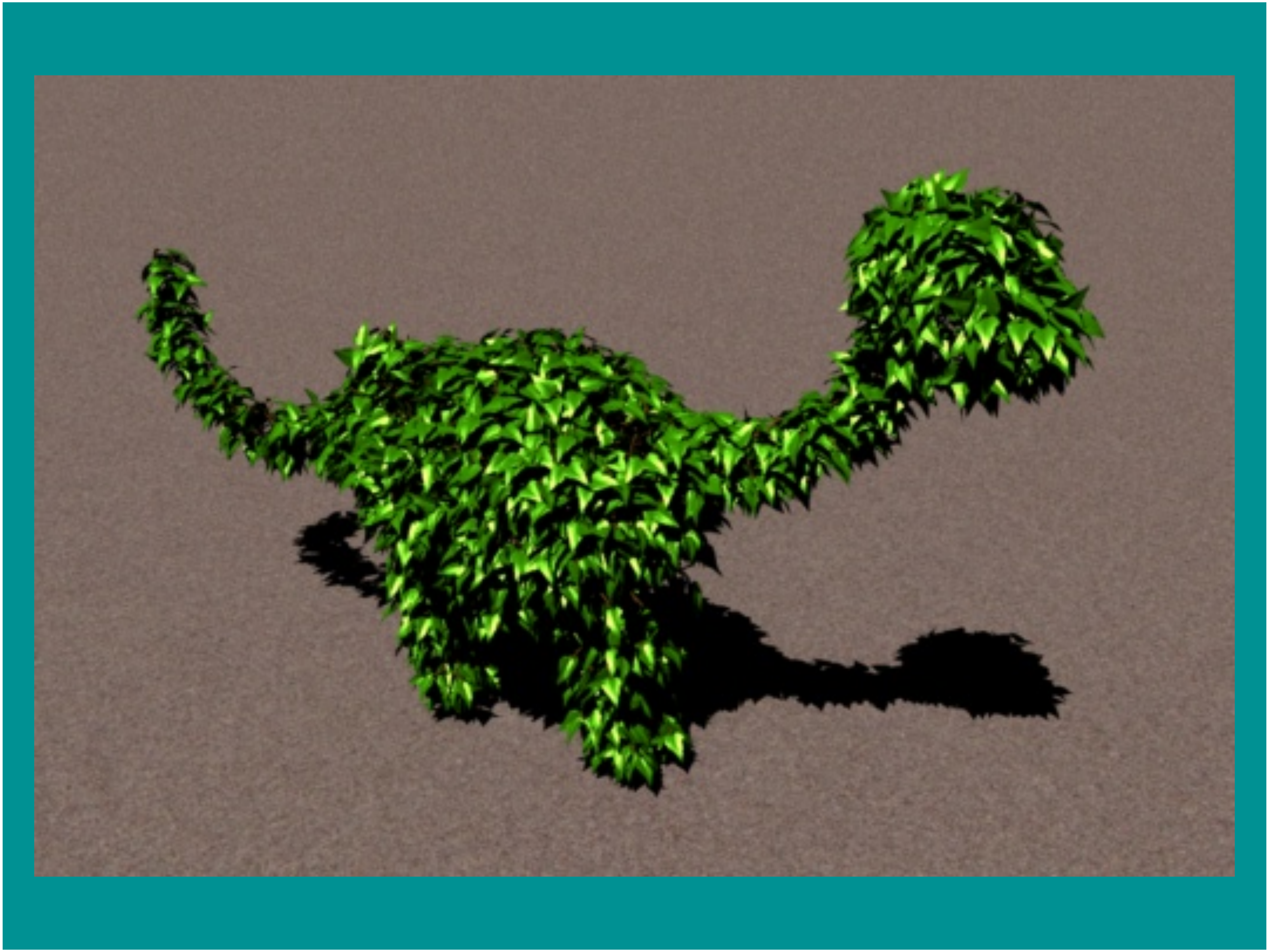
  $F \rightarrow FF-[F+F+F]+[+F-F-F]$

Additions

- 3D structure
- Randomness
- Leaves
- Flowers

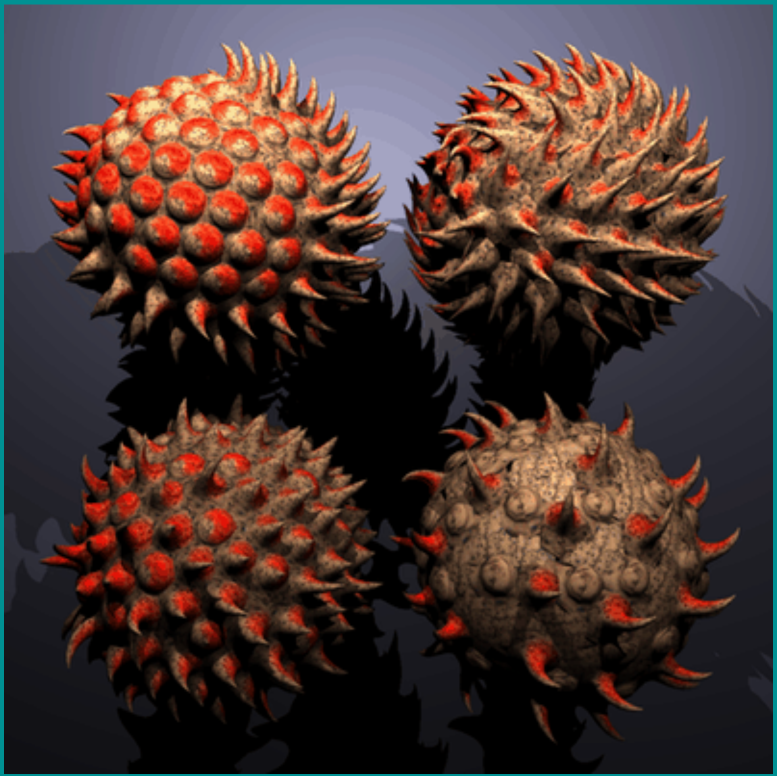

tenui



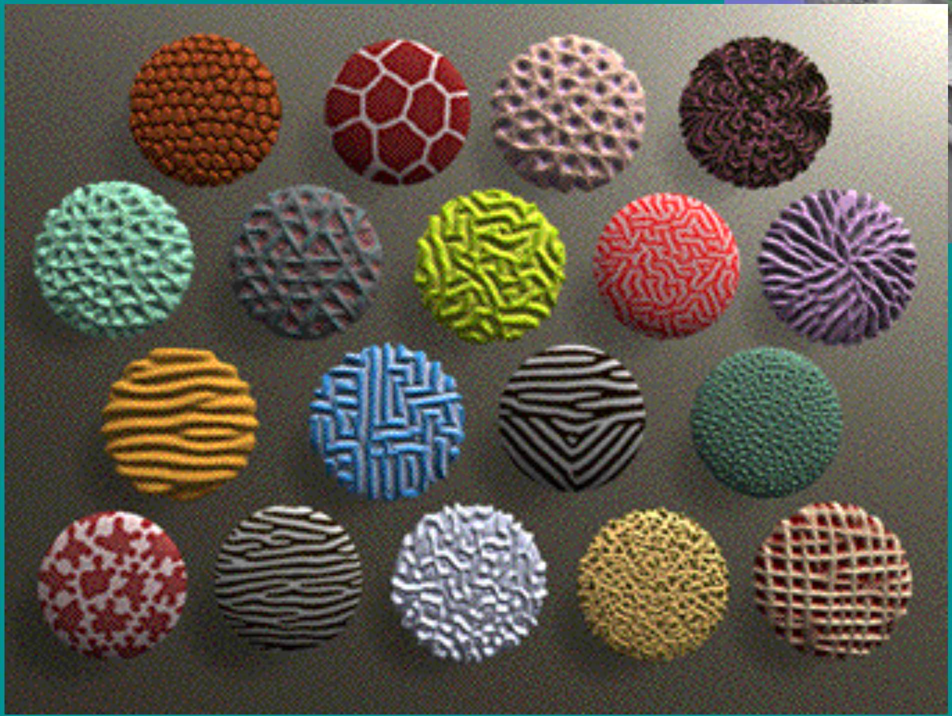


Biological Simulations

- Simulate developmental biological process

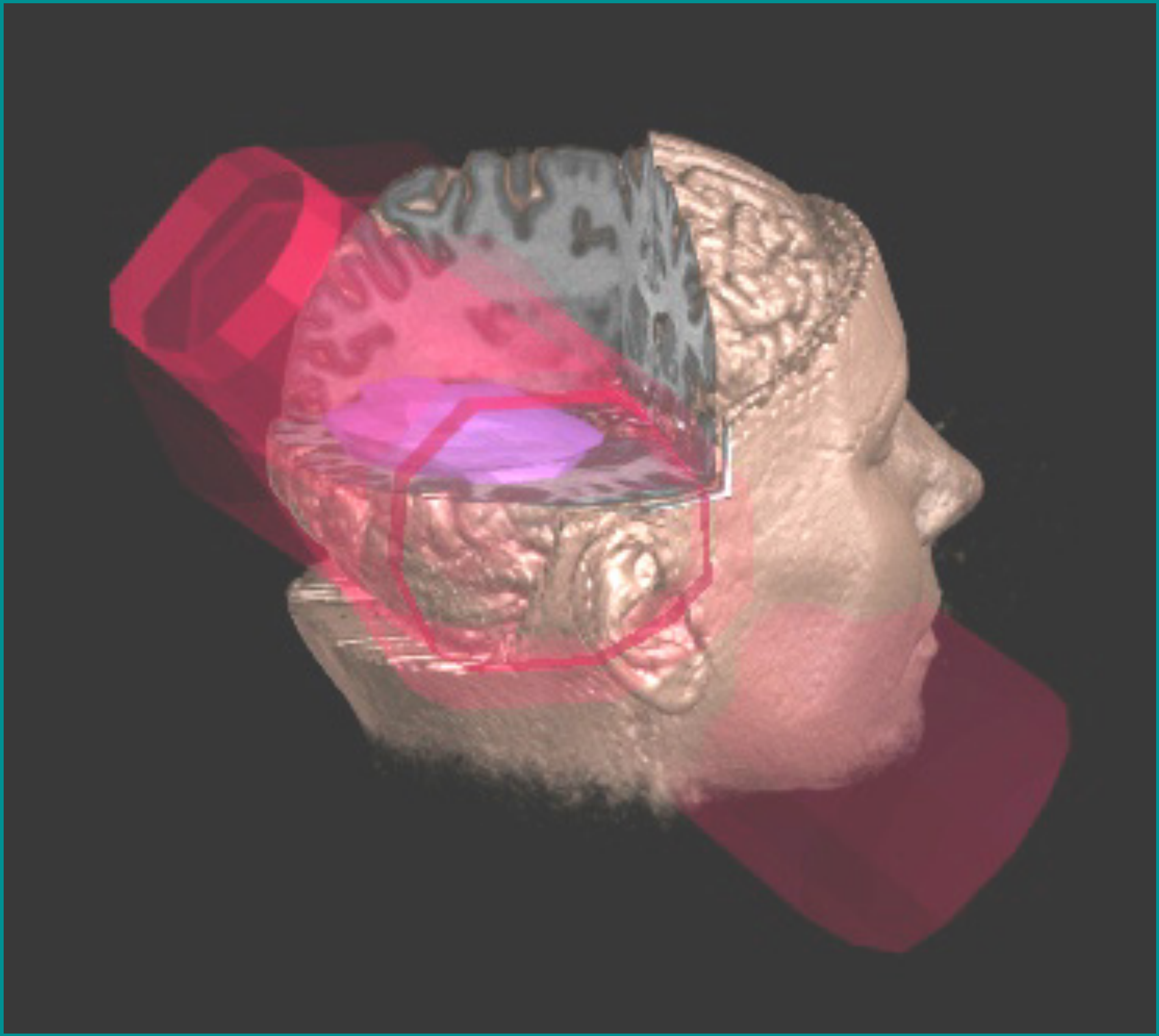


Reaction Diffusion

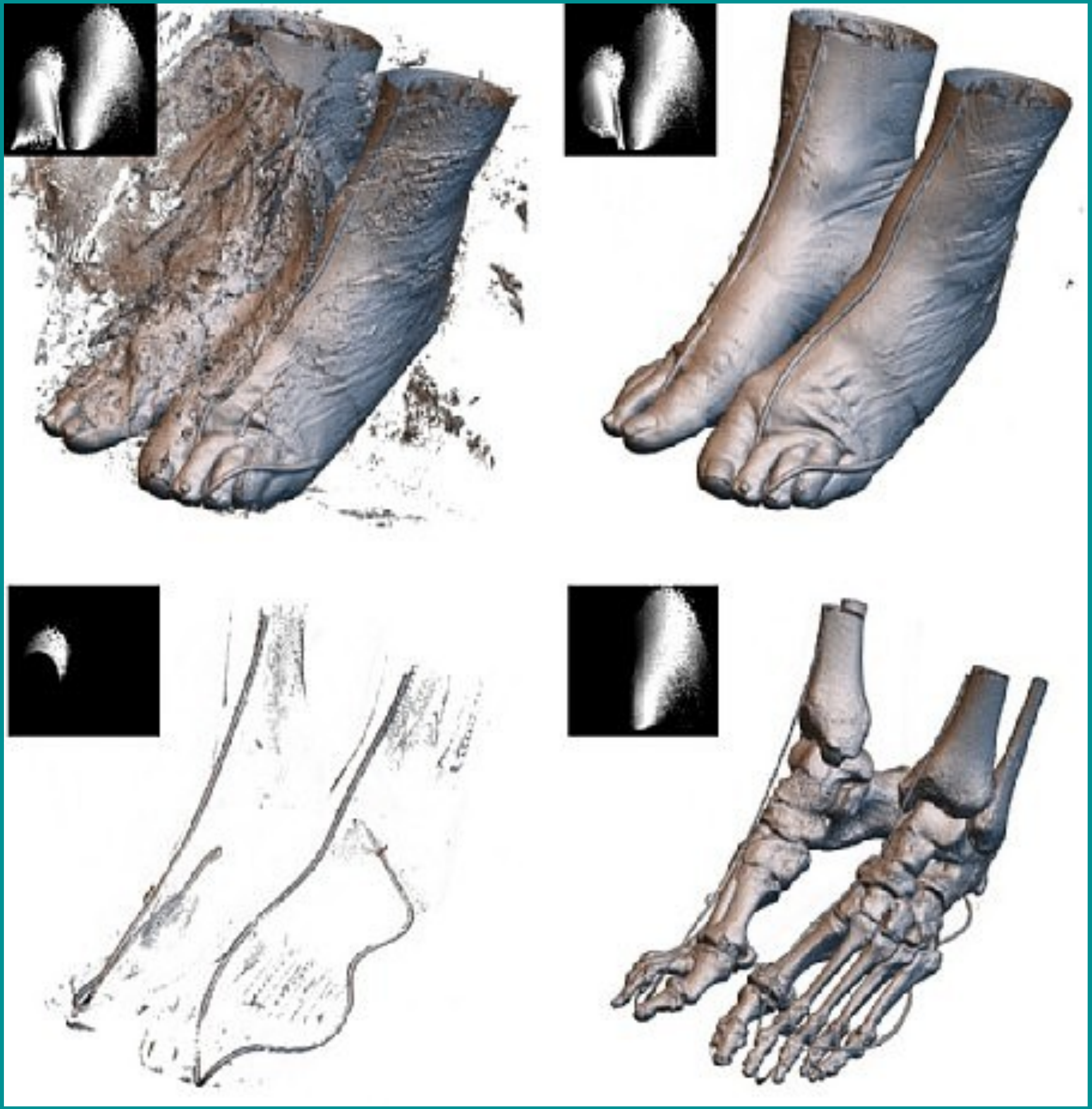


Visualization

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



Marc Levoy



Gordon Kindlemann

Digital Human Project -- NLM

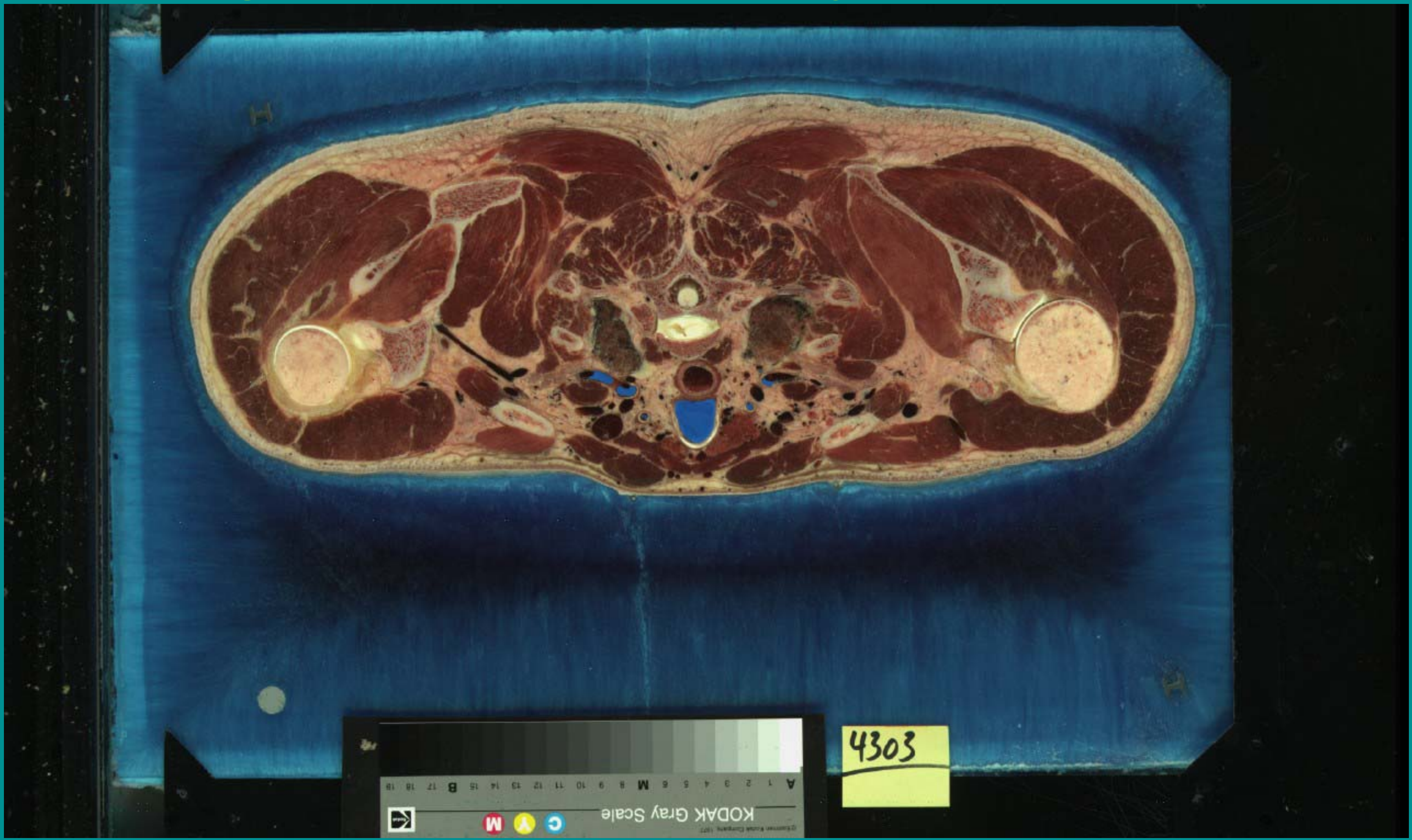


Image-based Rendering

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