CS 428: Fall 2009

Introduction to Computer Graphics

Procedural modeling

Procedural modeling

- Towards realism
  - Complexity = work (e.g. 2D/3D content creation)
  - Idea: put burden of *work* on computer for modeling relevant but nonspecific *detail*
  - Small specification → large range of detail/structure **amplification**
  - Examples
    - Mountains, trees, rivers, lightning, clouds, fire

Mountains

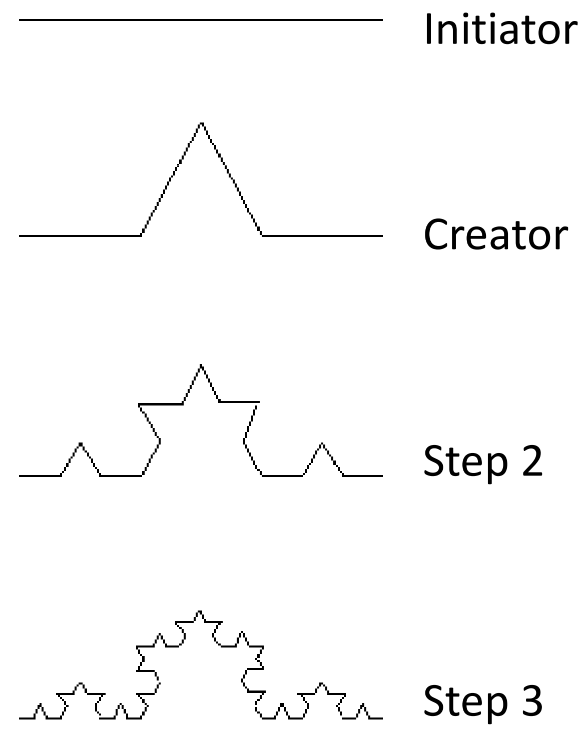
Clouds

Fractals

- Common approach in CG
- A *language* for complexity of form
- An engineering approach
  - Modeling by structural similarity
  - not based on reality
- Definition
  - A geometrically complex object constructed via repetition over a range of scales (sizes):
    leaves → trees → forests

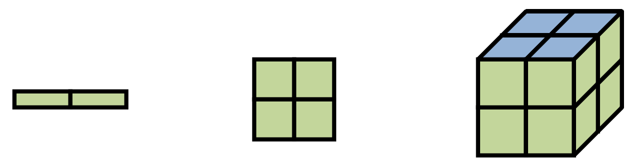
Fractal self similarity

- Fractals have some geometrical scale invariance
- Example: Koch-curve
  - Each of the 4 line segments in the k-th step is a minified version of the entire curve in previous step by factor 1/3
  - Cropped detail of the original curve can not be distinguished from the original



Fractal dimension

- For objects of dimensions 1, 2 and 3
- Subdivide into N equally sized parts
  - Line $r = \frac{1}{N}$
  - Square $r = \frac{1}{\sqrt{N}}$
  - Cube $r = \frac{1}{\sqrt[3]{N}}$



$$N = r^{-D} \quad \Leftrightarrow \quad D = \frac{\log N}{\log 1/r}$$

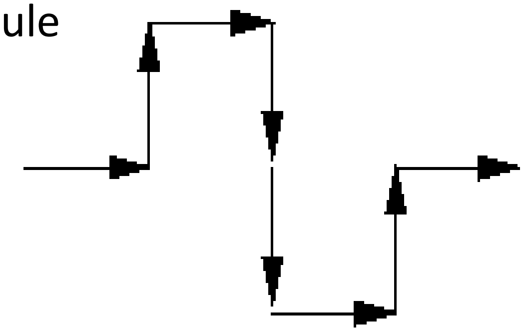
- A segment of the Koch-curve is made up from N=4 parts, each scaled by r=1/3

$$D = \frac{\log 4}{\log 3} = 1.2619$$

Fractal dimension

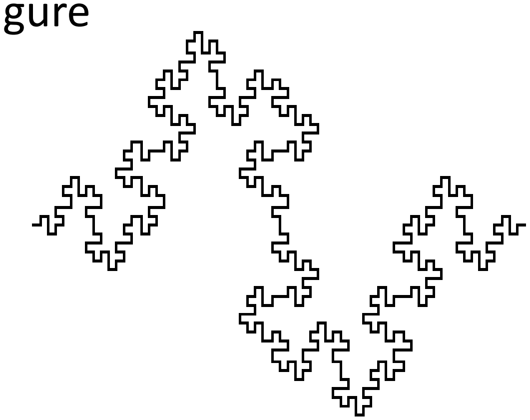
- Relation between number of parts N, and the associated scale factor r $D = \log N / \log 1/r$
- D is the fractal dimension or the self-similarity dimension of the structure (*roughness*)

Rule

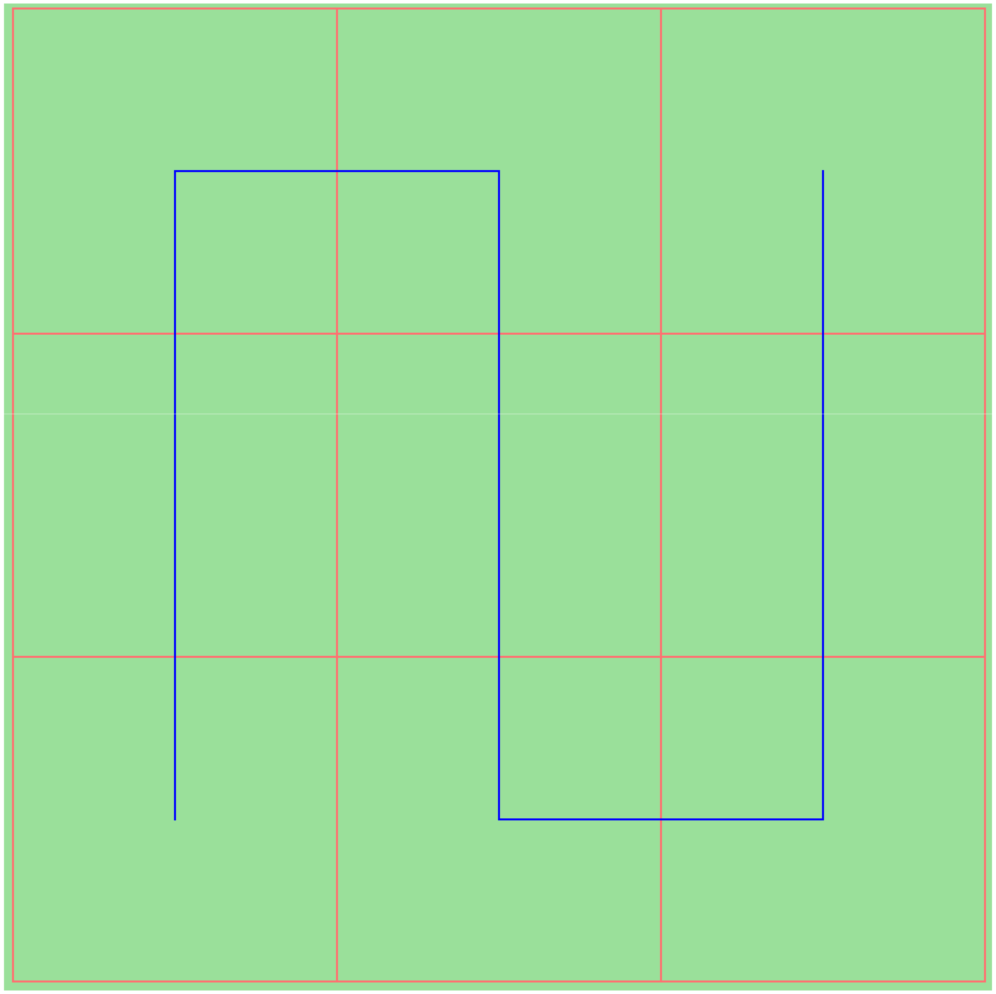


N=8, r=1/4, D=log(8)/log(4)=1.5

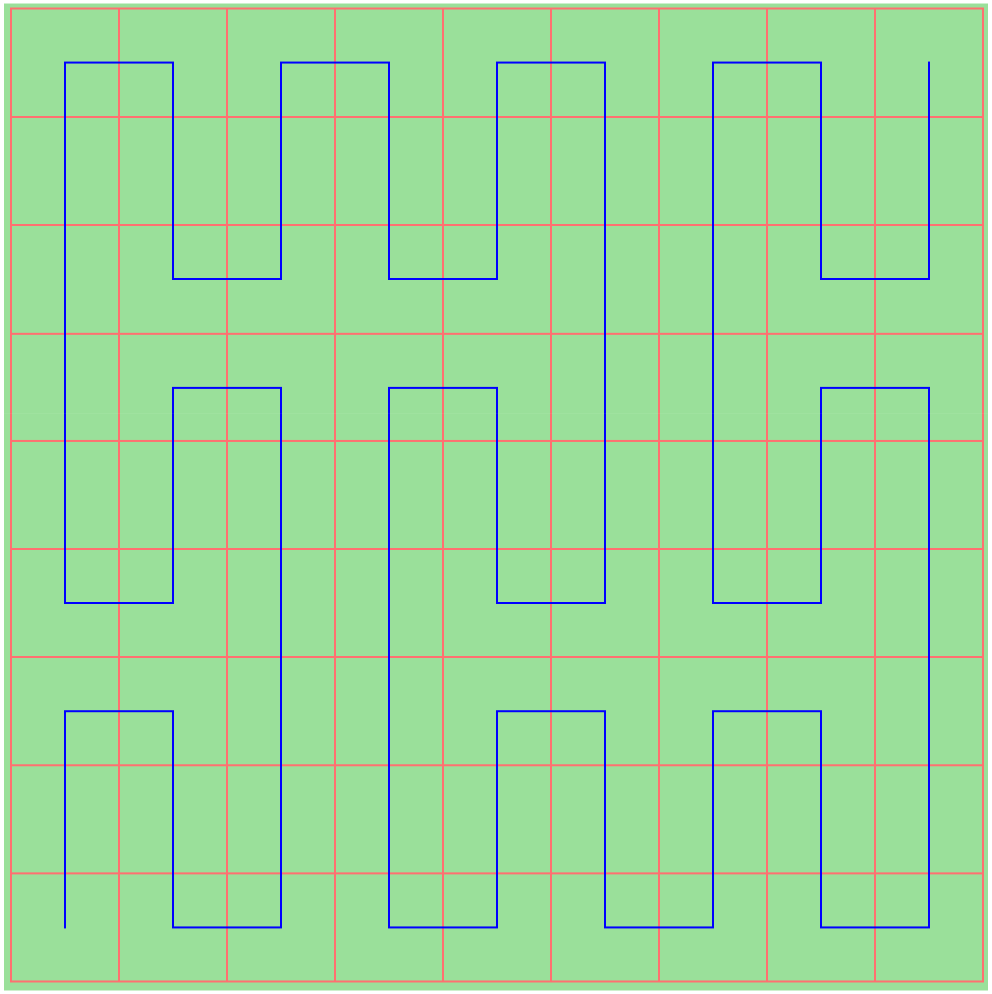
Figure



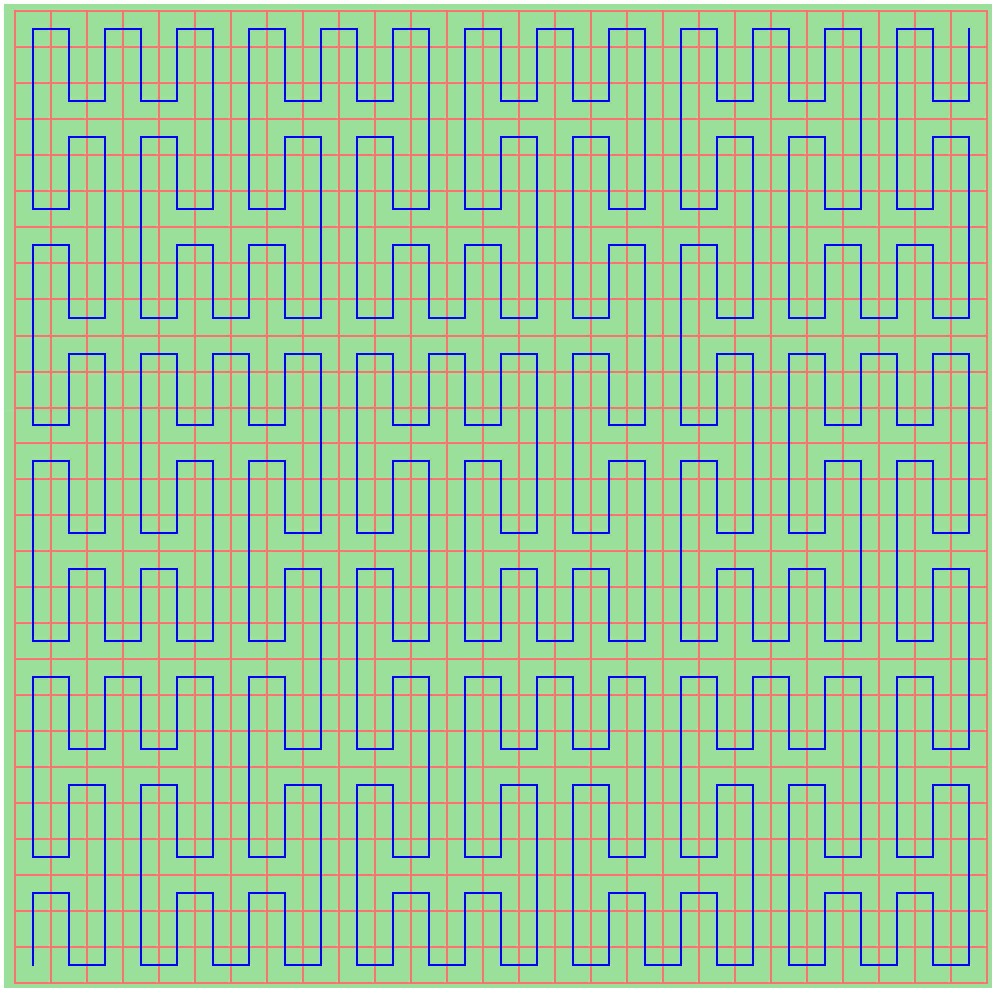
Peano space filling curves



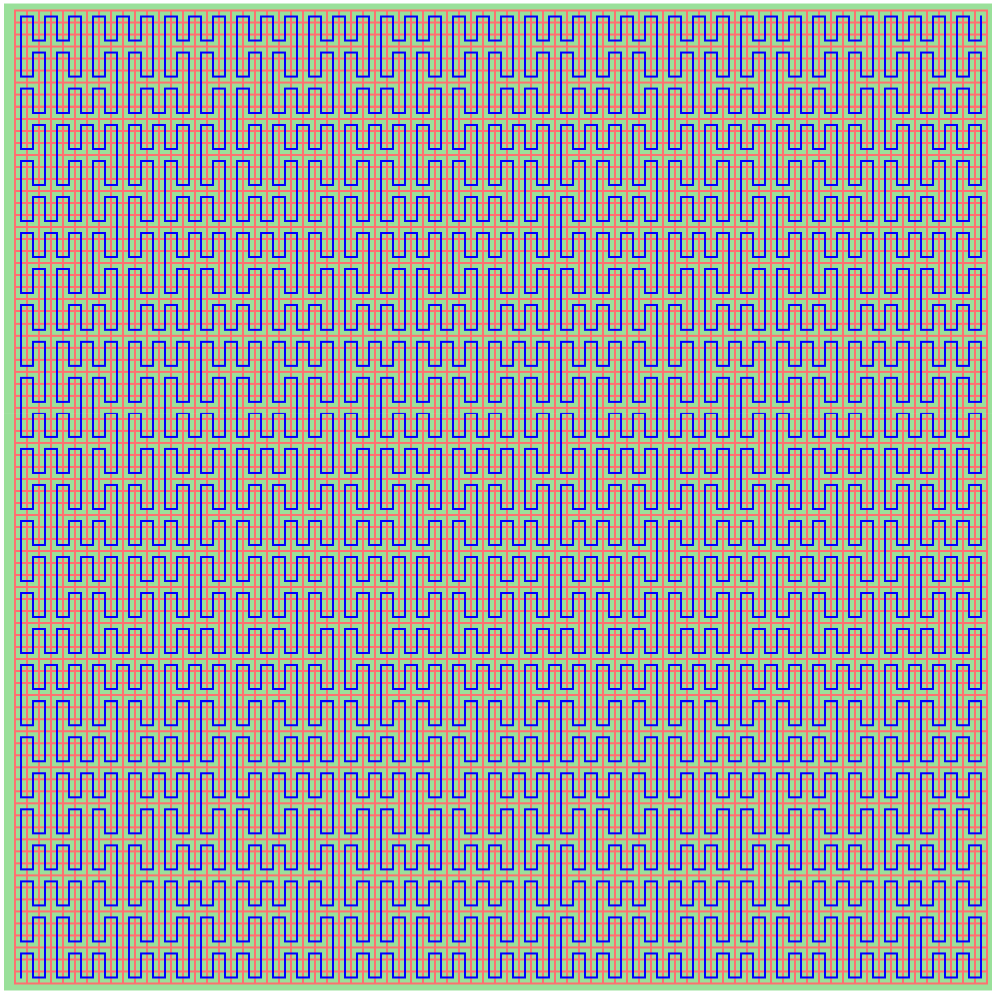
Peano space filling curves

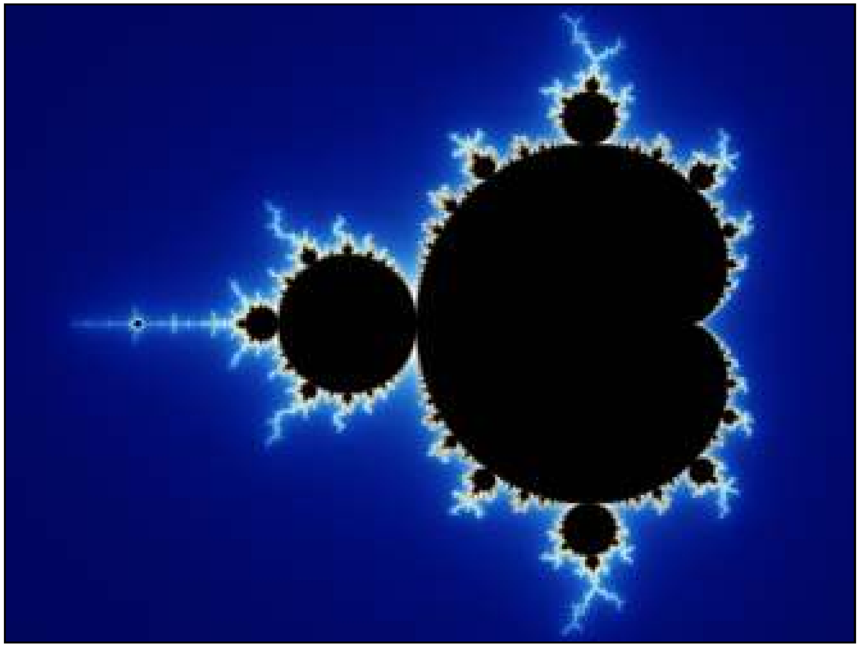


Peano space filling curves



Peano space filling curves

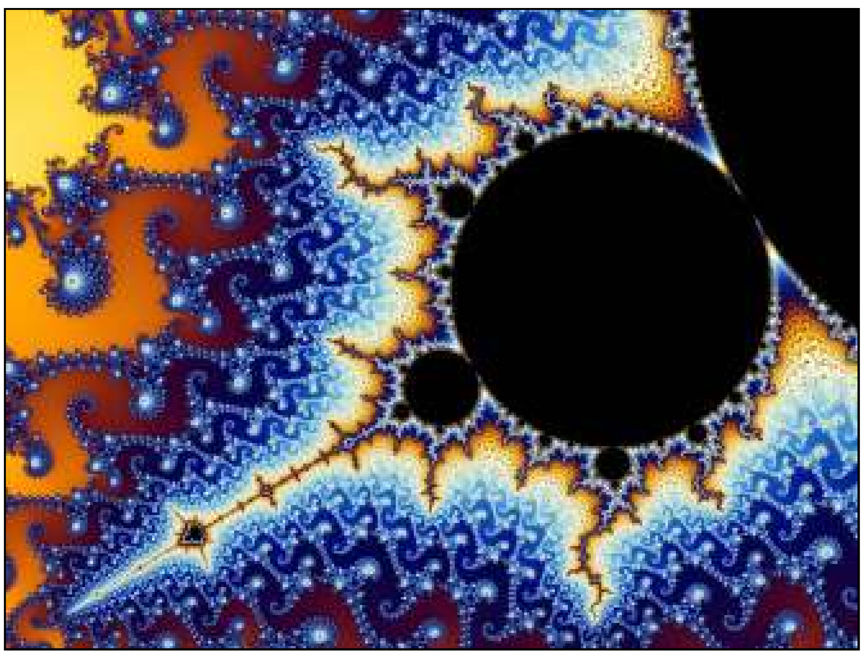


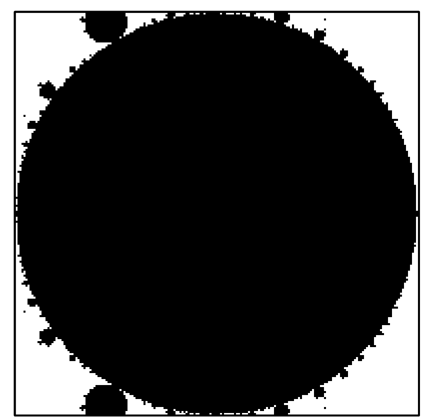


Mandelbrot set

float $m(x_0, y_0, i_{max})$
  $x = x_0$
  $y = y_0$

  for $i = 0$ to i_{max}
    if $(x^2 + y^2 > 4)$ return i
    $(x,y) \leftarrow (x_0 + x^2 - y^2,\ y_0 + 2xy)$
  end for
  return 0
end





Fractals

- Build complexity from repetition
- Structure repetition
  - Frequency, amplitude and lacunarity (space filling)
- Beneficial features
  - Fine structure at **all** scales
  - Not regular
  - Self similar
  - **Compact description**

Stochastistic self similarity

- Natural objects such as farns or coastlines are not exactly identical when magnified
  - But characteristics are similar
  - Magnified coast line similar to original
  - This phenomenon is captured by stochastic self-similarity



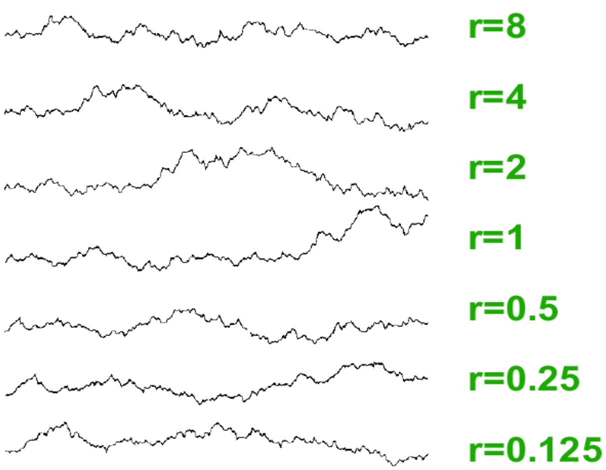
Stochastistic fractals

- Used to model
  - Terrain, clouds, waves, tree bark, etc.
  - Useful for generating 3D wood/marble textures
  - Simulation of **Brownian motion**

Brownian motion

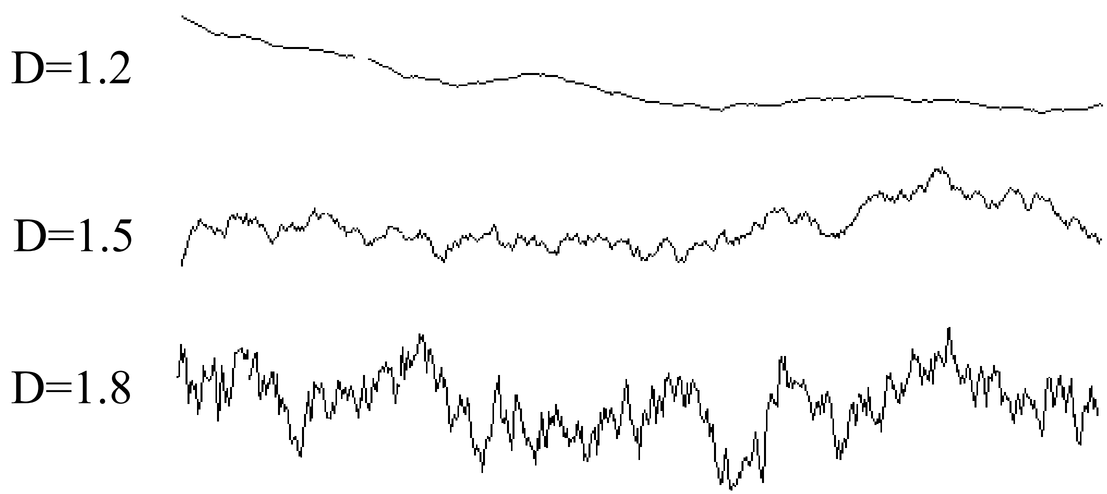
- Non-stationary stochastic process {X(t)}
- Increments X(t+s)-X(t) are Gaussian distributed with expected value of zero
- Variance of increments is proportional to time difference → $var\,(X(t+s)\text{-}X(t)) \sim |s|$
- Statistically self-similar

$X(t)$ Statistically similar to $\frac{1}{\sqrt{r}}X(rt)$



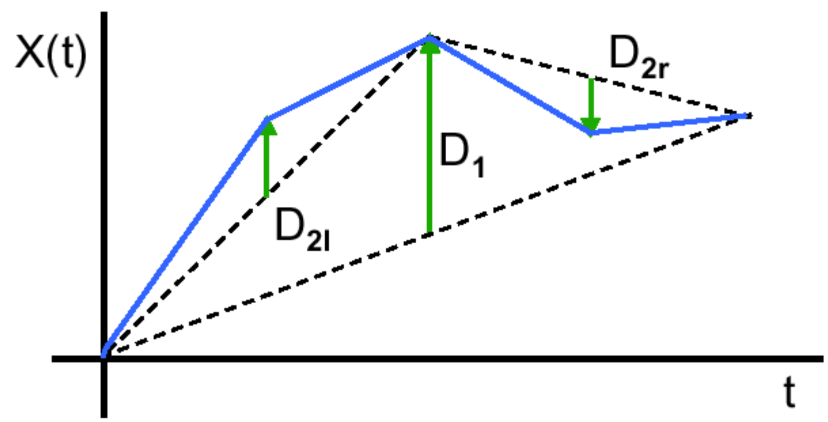
Fractal Brownian motion

- Brownian motion has fractal dimension 1.5
- Dimension D of natural objects ranges from 1.15 and 1.25
    - Use of fractal Brownian motion (fBm)



Brownian motion application

- Midpoint displacement (for $D \in [1,2]$)
  - Step 0: $X(1/2) = (X(0) + X(1))/2 + D_0$
  - Step n: linear interpolation between neighboring points and displacement by D_n
    - D_n is a Gaussian distributed random variable with $E(D_n)=0$ and $var(D_n)=(1-2^{2-2D})/2^{n(4-2D)}$
  - $X(t)$ and $1/2^{(2-D)}X(2t)$ are statistically self-similar

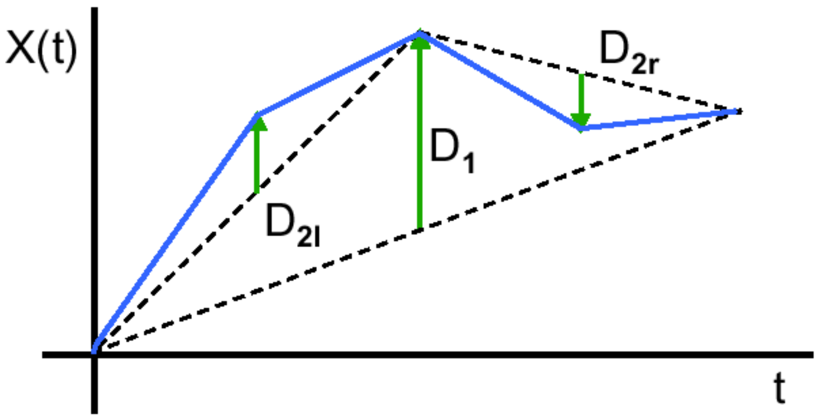


Brownian motion application

- Midpoint displacement (for $D \in [1,2]$)
- Case of D=1.5, with $E(D_n) = 0$ (mean = 0)
  - Start with $var(D_n)=0.5$ (for initial grid spacing of 1)
  - In each step, scale $var(D_n)$ by 1/2
  - Same as $var(D_n)=0.5 \cdot (1/2)^n$
  - Use **`Random`** class in Java

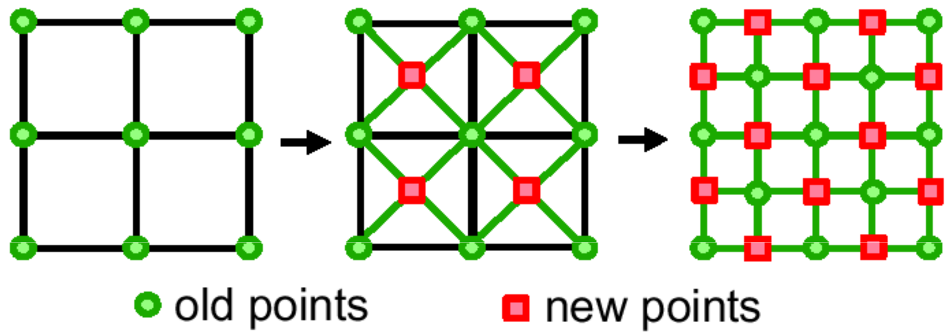
    `nextDouble()` (in [0,1])

    `nextGaussian()` (mean = 0, var = 1)



Midpoint displacement in 2D

- Create refined grid

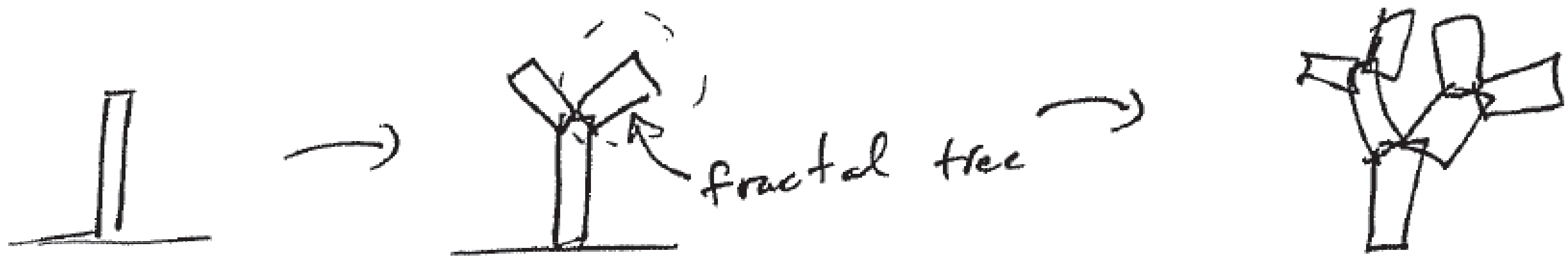


  - In every step the grid resolution is scaled by a factor of $r = \frac{1}{\sqrt{2}}$
  - The variance of random displacements is scaled by

$$r^{4-2D} = \left(\frac{1}{2}\right)^{2-D} \qquad \text{for D = 1.5} \qquad r = \left(\frac{1}{2}\right)^{0.5} = \frac{1}{\sqrt{2}}$$

Fractal trees

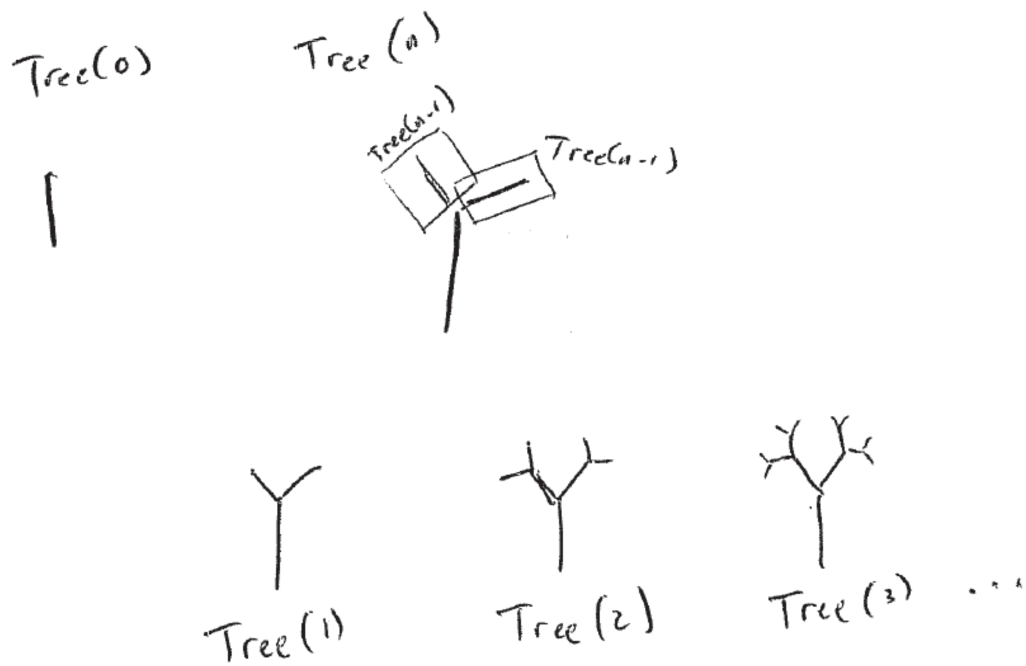
- Hierarchical fractal modeling



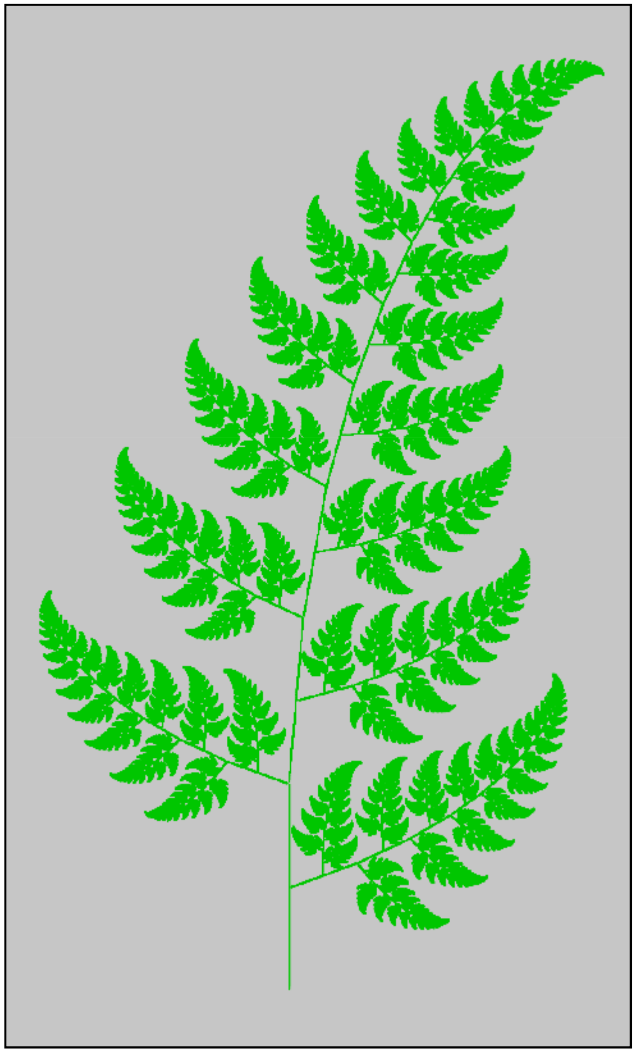
- Random angles, lengths, placements

Fractal trees

- At end of recursion, add leaf
- Decrease branch length and radius as recursion depth increases



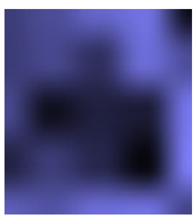
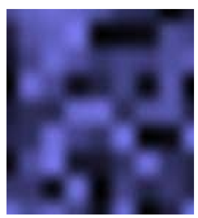
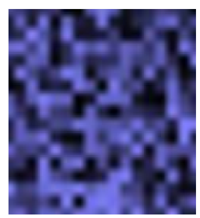
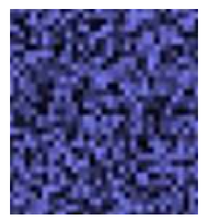
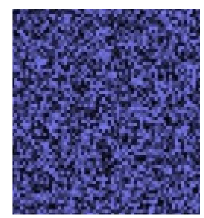
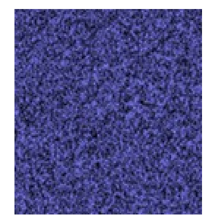
Fractal fern

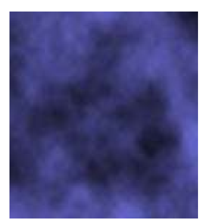


Procedural textures

Perlin Noise

- Sum of band limited functions (octaves)
  - In uv space compute a pseudorandom number and a gradient
  - Interpolation (linear, cubic) between integer uv's provides a smooth band limited function

1/6 * ( +  +  +  +  + )

= 

Procedural textures

Perlin Noise