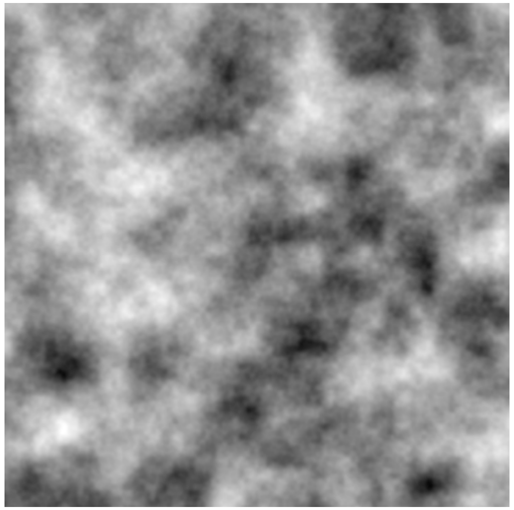
Computer Graphics

Procedural Methods - Fractals

Mark Pauly

Geometric Computing Laboratory

Self-similarity

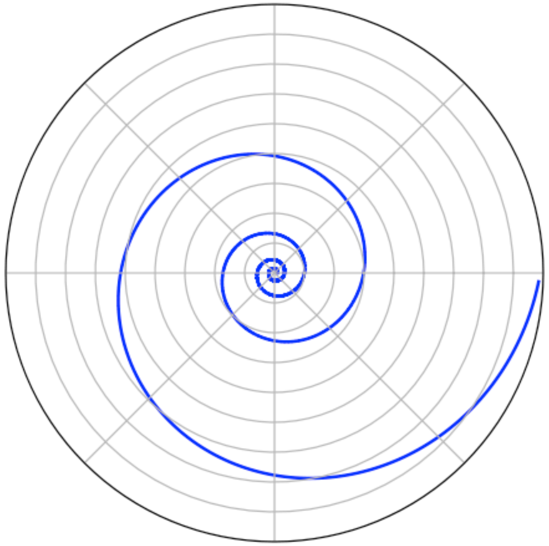


fBm

- fBm exhibits *self similarity*
- In the case of fBm, the self-similarity is only *statistical*

Sea Shells

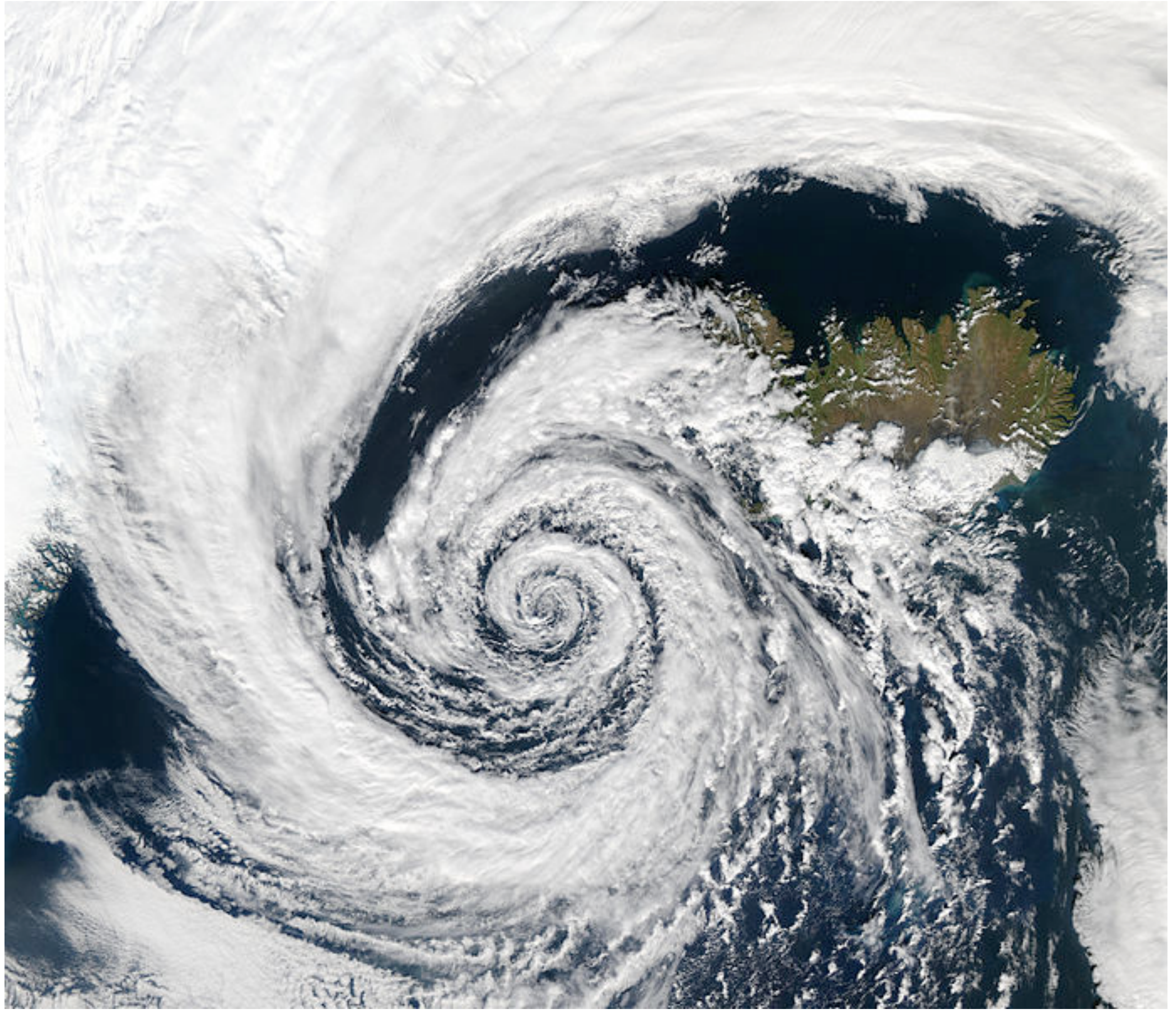
Logarithmic Spirals



$$x(t) = ae^{bt}\cos(t)$$
$$y(t) = ae^{bt}\sin(t)$$

$$e^{2\pi b}x(t) = e^{2\pi b}\Big(ae^{bt}\cos(t)\Big) = ae^{b(t+2\pi)}\cos(t+2\pi) = x(t+2\pi)$$
$$e^{2\pi b}y(t) = e^{2\pi b}\Big(ae^{bt}\sin(t)\Big) = ae^{b(t+2\pi)}\sin(t+2\pi) = y(t+2\pi)$$

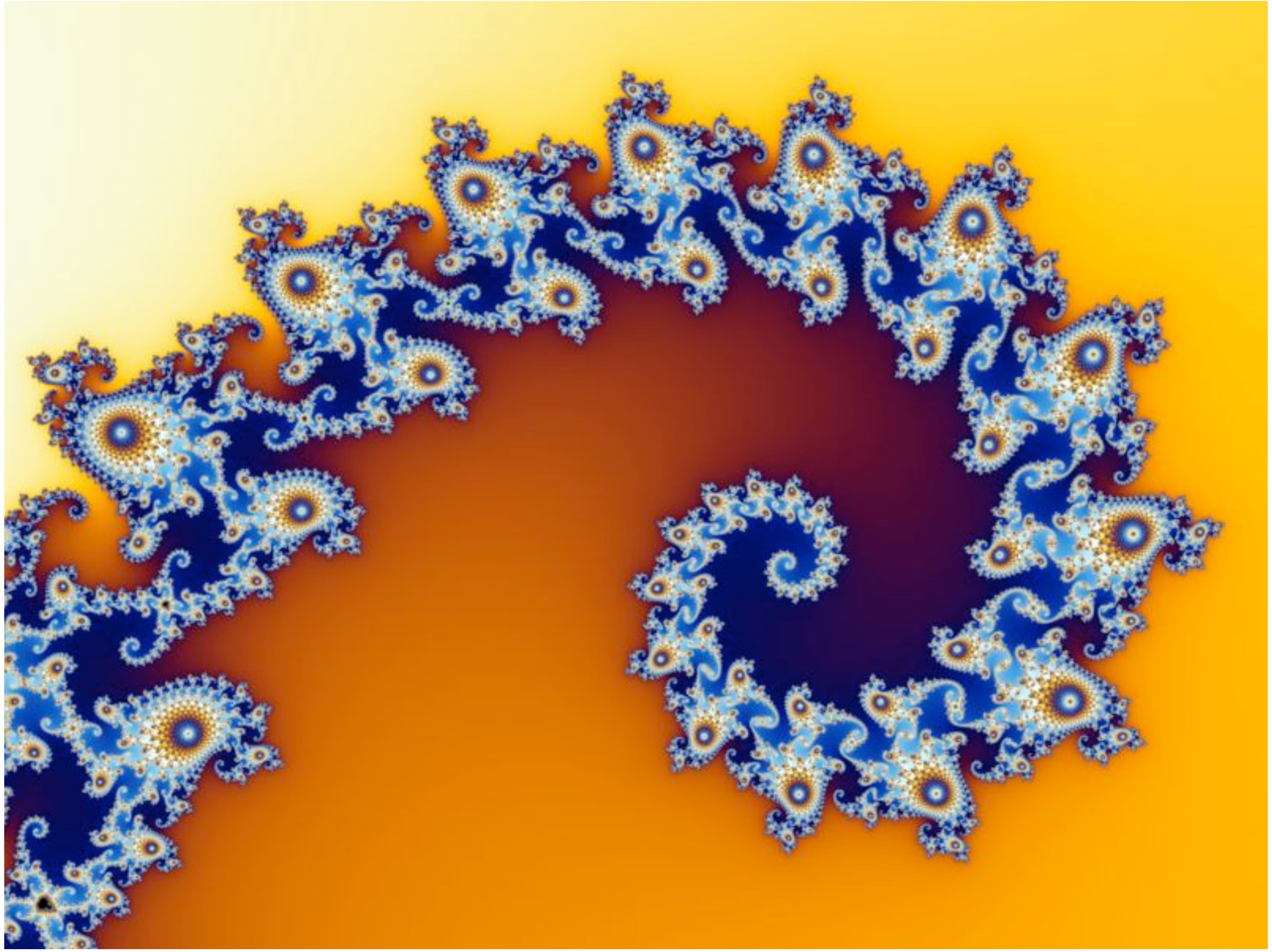
Low Pressure System



Milky Way

Romanescu

Mandelbrot Set



Snowflakes

Fern

Leaves

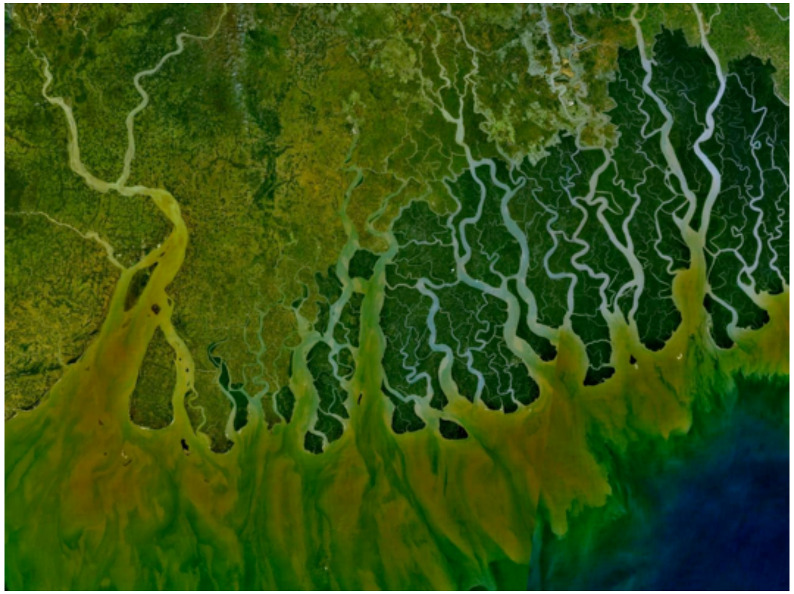


Trees

Lightning

Mountains

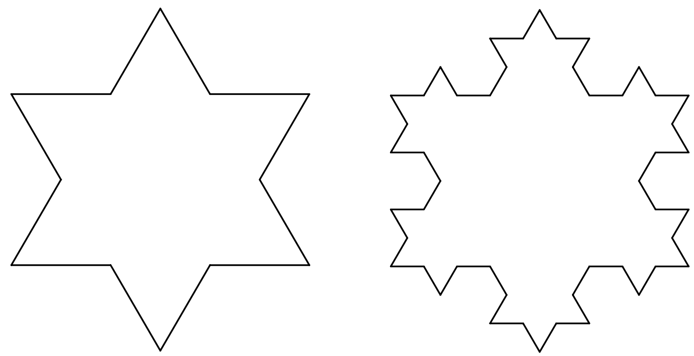
Rivers and Fjords



Shorelines

Fractals

- Colloquial definition:
  - Repetition of a given form over a range of scales
  - Self-similar
- Detail at multiple levels of magnification
- Fractal dimension exceeds topological dimension
- Does not have to be self-similar!



How Long is the Coast of Britain?

unit = 200km
total length=2400km

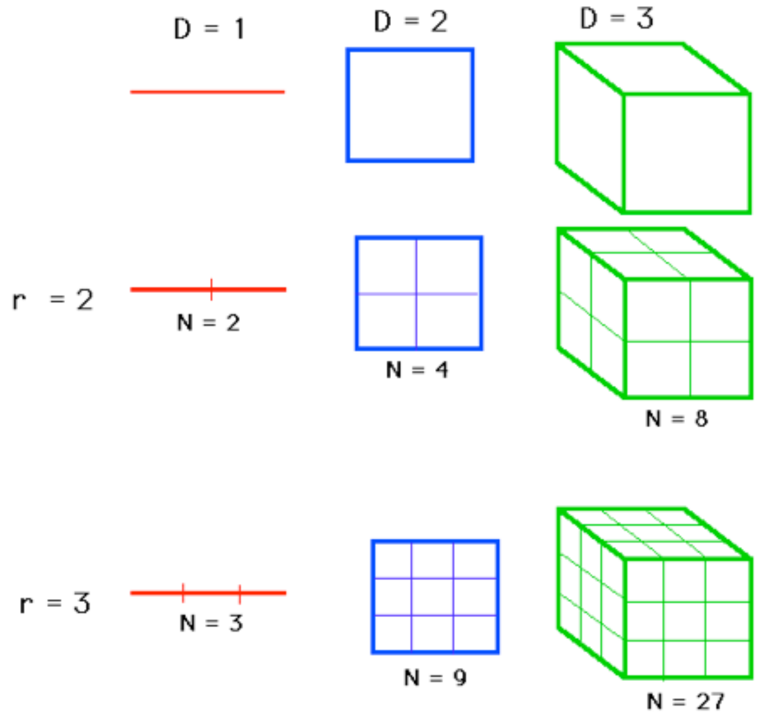
unit = 100km
total length=2800km

unit = 50km
total length=3400km

The “Coastline Paradox”

Integer (Euclidean) Dimension

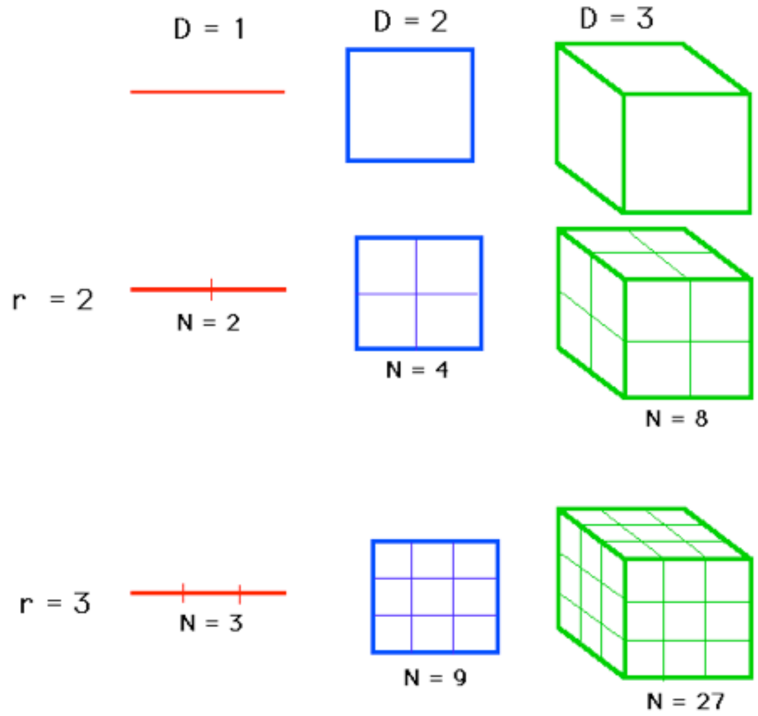
- Take an object residing in Euclidean dimension D and reduce size by $1/r$ in each spatial direction
- Exactly $N = r^D$ self-similar objects cover the original



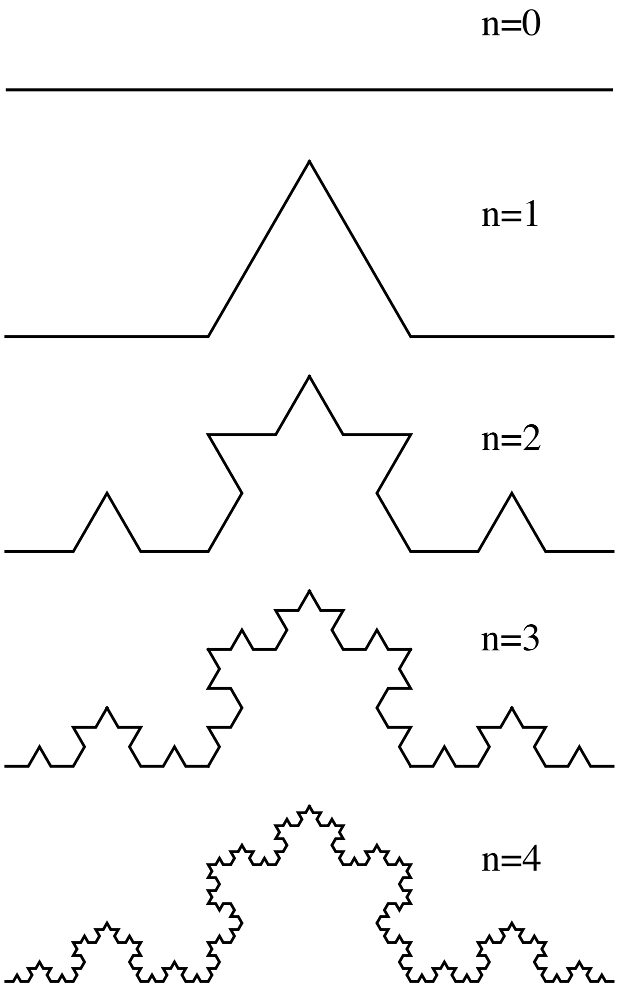
Hausdorff (Fractal) Dimension

- Take an object residing in Euclidean dimension D and reduce size by $1/r$ in each spatial direction
- Exactly $N = r^D$ self-similar objects cover the original
- Use this scaling relationship to *define* dimension:

$$N = r^D$$
$$\log(N) = D\log(r)$$
$$D = \frac{\log(N)}{\log(r)}$$



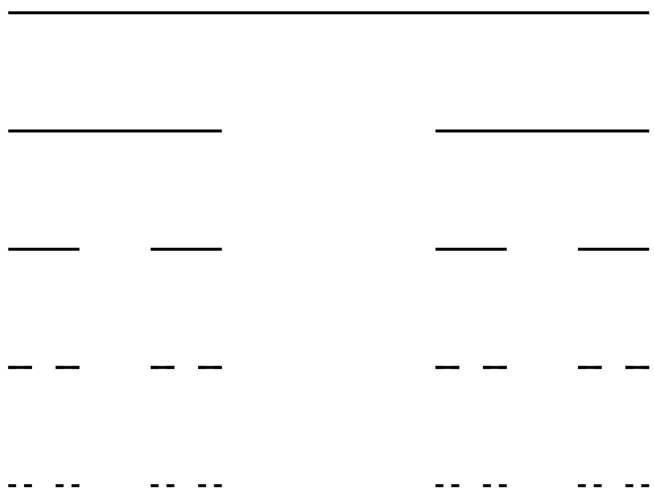
Fractal Dimension Example: Koch Curve



$$D = \frac{\log(N)}{\log(r)}$$

$$\begin{aligned} N &= 4 \\ r &= 3 \\ D &= \log(4)/\log(3) \\ &= 1.26185951\ldots \end{aligned}$$

Fractal Dimension Example: Cantor Dust



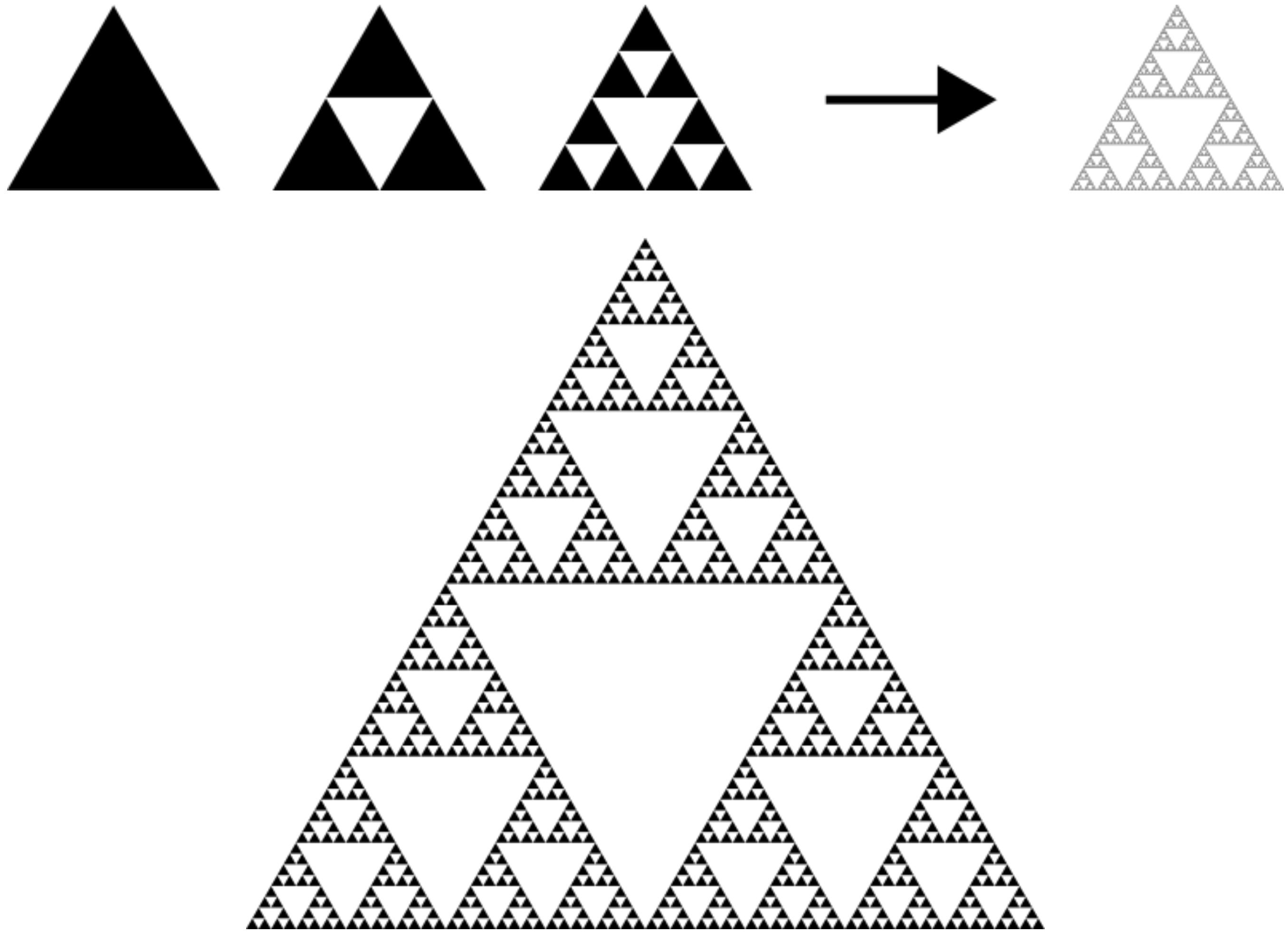
$$D = \frac{\log(N)}{\log(r)}$$

$$\begin{aligned} N &= 2 \\ r &= 3 \\ D &= \log(2)/\log(3) \\ &= 0.630929754\ldots \end{aligned}$$

- How much is left?
- Total length of deleted intervals:

$$\sum_{n=0}^{\infty} \frac{2^n}{3^{n+1}} = \frac{1}{3} + \frac{2}{9} + \frac{4}{27} + \frac{8}{81} + \cdots = \frac{1}{3}\left(\frac{1}{1 - \frac{2}{3}}\right) = 1$$

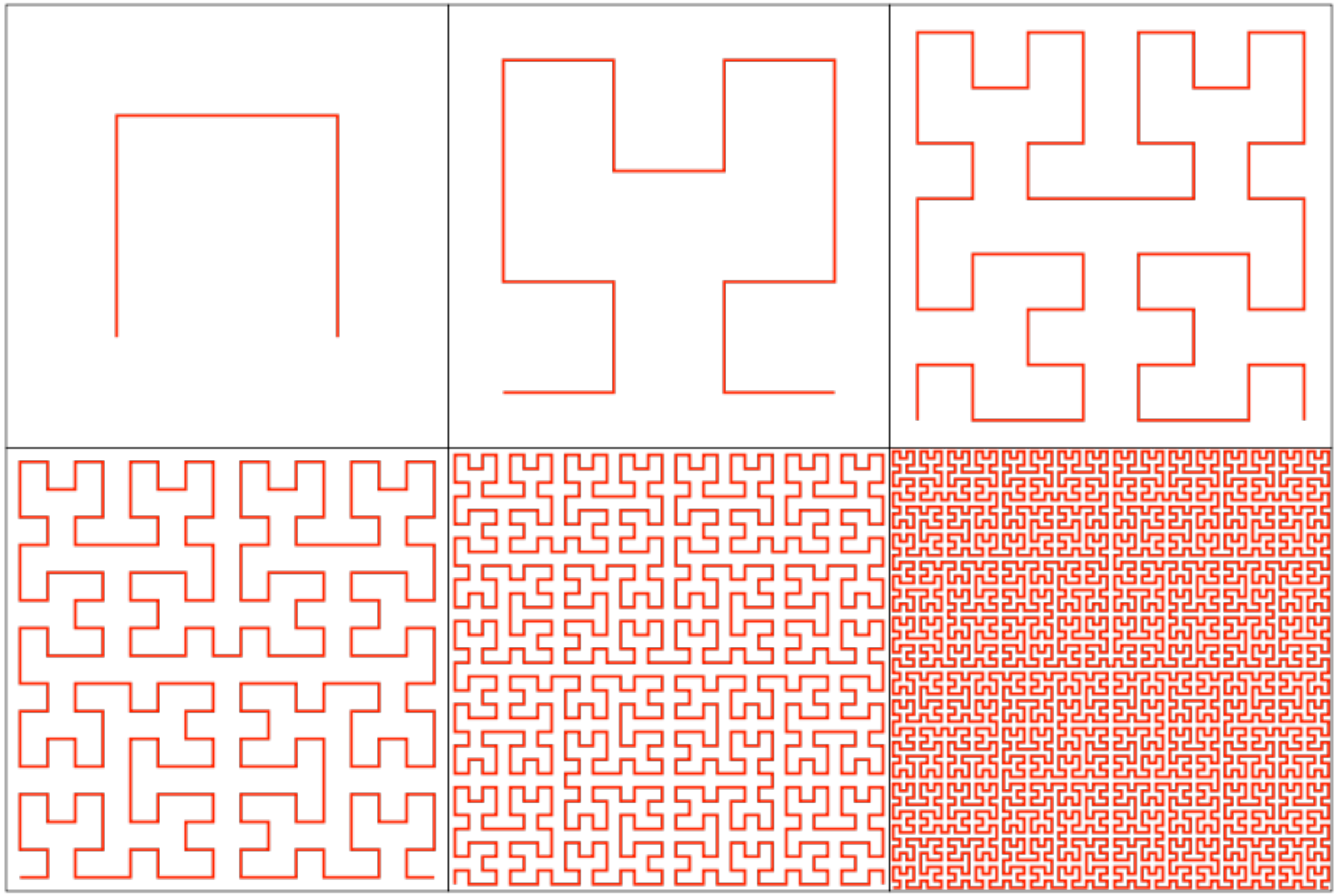
Fractal Dim Example: Sierpinski triangle



Fractal Dimension

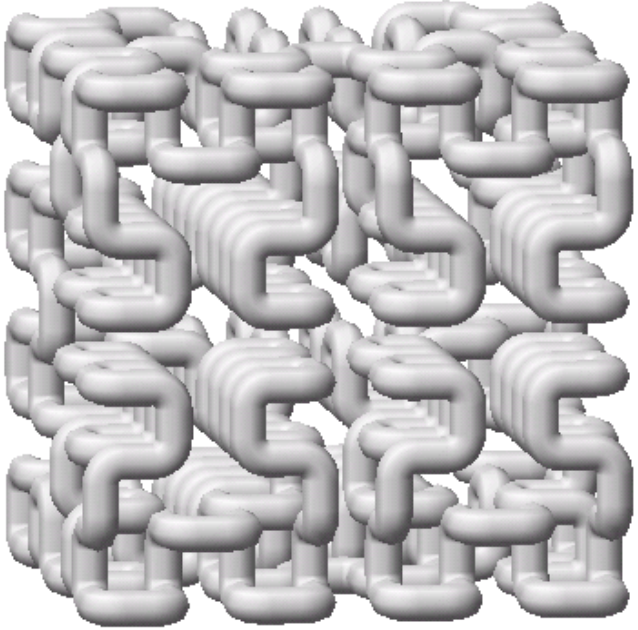
- Fractals have fractal dimension, e.g., D = 1.2
- Continuous “slider” for the visual complexity of a fractal
  - “smoother” $\Longleftrightarrow$ “rougher”
- Integer component typically indicates underlying Euclidean dimension of the fractal, in this case a line (“1.” in 1.2)
  - Exception: examples like Cantor Dust, where iterations cut parts away
- Fractional part is called the *fractal increment* (“.2” in 1.2)
- Fractal increment varies from 0.0 to 0.999…
  - fractal transitions from (locally) occupying only its underlying Euclidean dimension (the line), to filling some part of the next higher dimension (the plane)

Hilbert Curve



source: wikipedia

Hilbert Curve



source: wikipedia

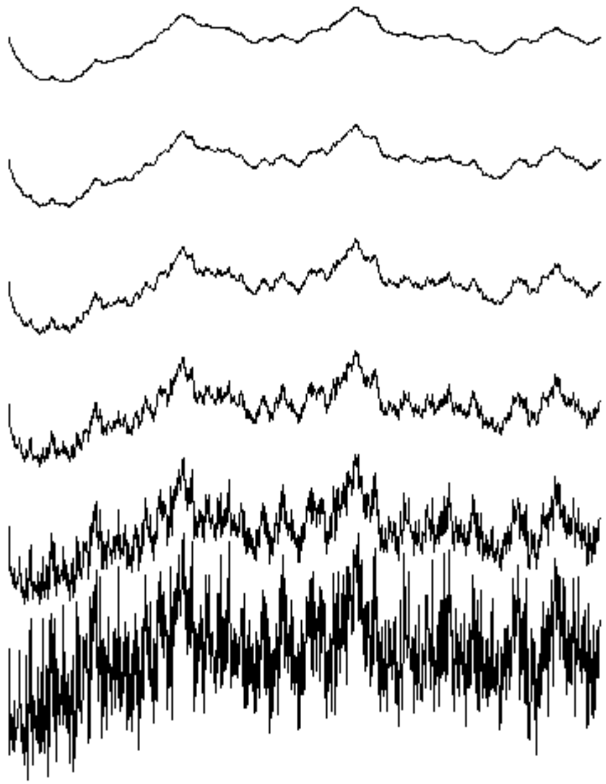
Fractional Brownian Motion (fBm)

$$f_s(x) = \sum_i l^{-iH} f(l^i x)$$

```
double fBm(vector point, double H, double lacunarity, int octaves) {
    double value = 0.0;
    // inner loop of fractal construction
    for (int i = 0; i < octaves; i++) {
        value += Noise(point) * pow(lacunarity, -H*i);
        point *= lacunarity;
    }
    return value;
}
// See Ebert et al: Texturing & Modeling: A Procedural Approach
```

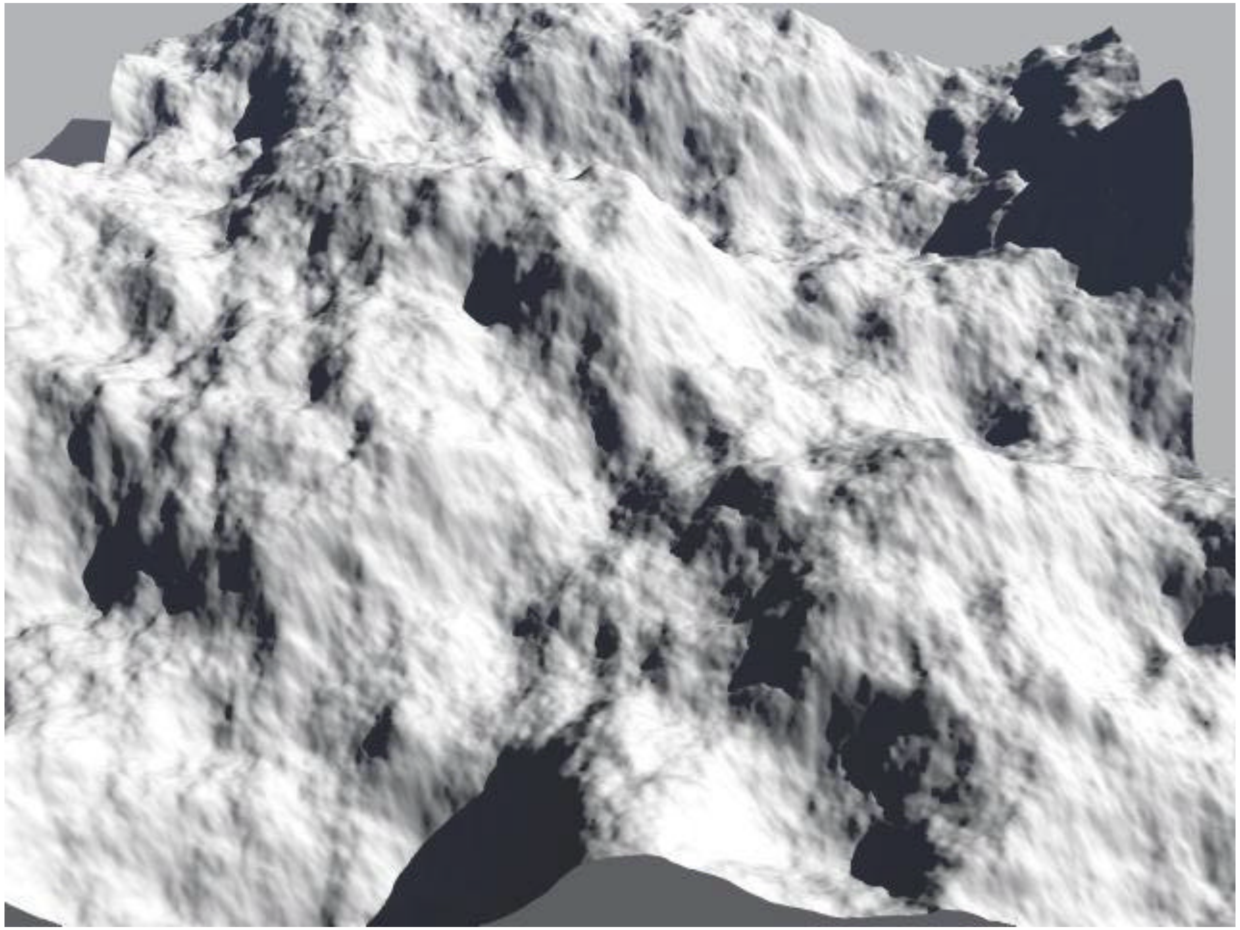
- Parameter H actually equals $1.0 -$ [fractional increment]
- When $H = 1$, the fBm is relatively smooth
- As $H \to 0$, the function becomes more like white noise

Roughness Control



Plot of fBm for H varying from 1.0 to 0.0 in increments of 0.2

2D fBm



Ordinary fractional Brownian motion (fBm) terrain patch of fractal dimension ~2.1

Fractal Brownian Motion

- fBm is statistically homogeneous and isotropic
  - Homogeneous means "the same everywhere"
  - Isotropic means "the same in all directions"
- Fractal phenomena in nature are rarely so simple and well-behaved.

Multifractals

- Fractal system which has a different fractal dimension in different regions
- Heterogeneous fBM
  - Typical implementations *don't* just spatially vary the H parameter
  - One strategy: scale higher frequencies in the summation by the value of the previous frequency.
  - Many possibilities: heterogenous terrain, hybrid multifractal, ridged multifractal
    - See the Texturing & Modeling book [Ebert et al.] for details

Multifractals

- “Multiplicative-cascade” multifractal variant of fBm

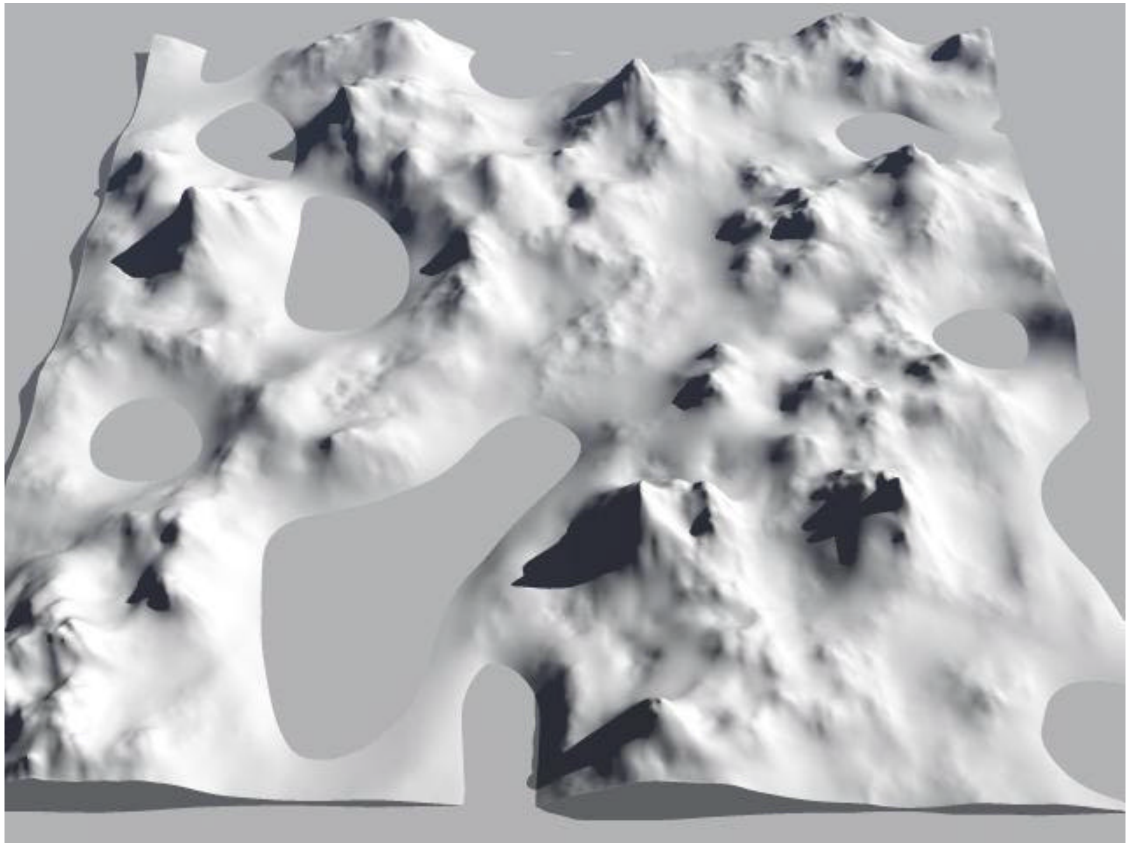
```
double multifractal(vector point, double H, double lacunarity, int octaves, double offset) {
    double value = 1.0;
    for (int i = 0; i < octaves; i++) {
        value *= (Noise(point) + offset) * pow(lacunarity, -H*i);
        point *= lacunarity;
    }
    return value;
}
// See Ebert et al: Texturing & Modeling: A Procedural Approach
```

fBm vs. Multifractals

```
double fBm(vector point, double H, double lacunarity, int octaves) {
    double value = 0.0;
    // inner loop of fractal construction
    for (int i = 0; i < octaves; i++) {
        value += Noise(point) * pow(lacunarity, -H*i);
        point *= lacunarity;
    }
    return value;
}

double multifractal(vector point, double H, double lacunarity, int octaves, double offset) {
    double value = 1.0;
    for (int i = 0; i < octaves; i++) {
        value *= (Noise(point) + offset) * pow(lacunarity, -H*i);
        point *= lacunarity;
    }
    return value;
}
```

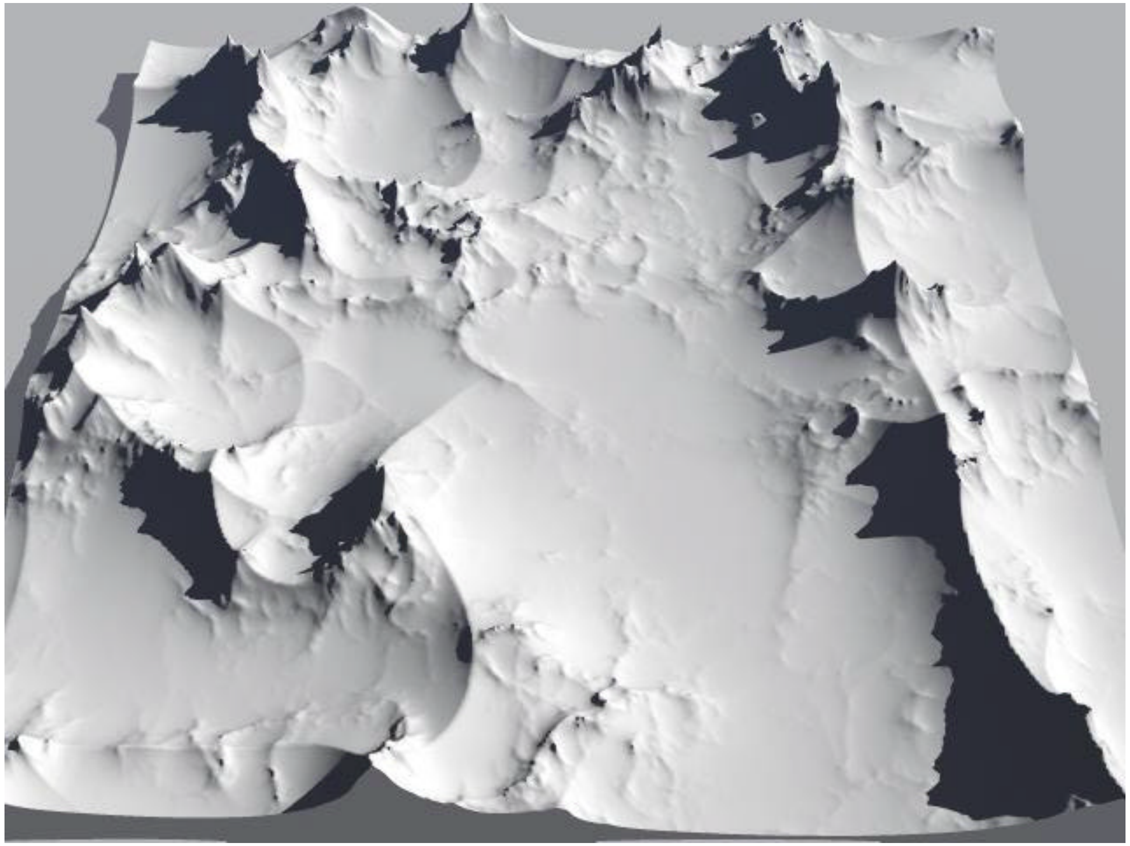
Heterogeneous fBm



A hybrid multifractal terrain patch made with a Perlin noise basis: the "alpine hills" Bryce 4 terrain model. There is a flat ground plane added at altitude zero to mask details below.

source: Ken Musgrave

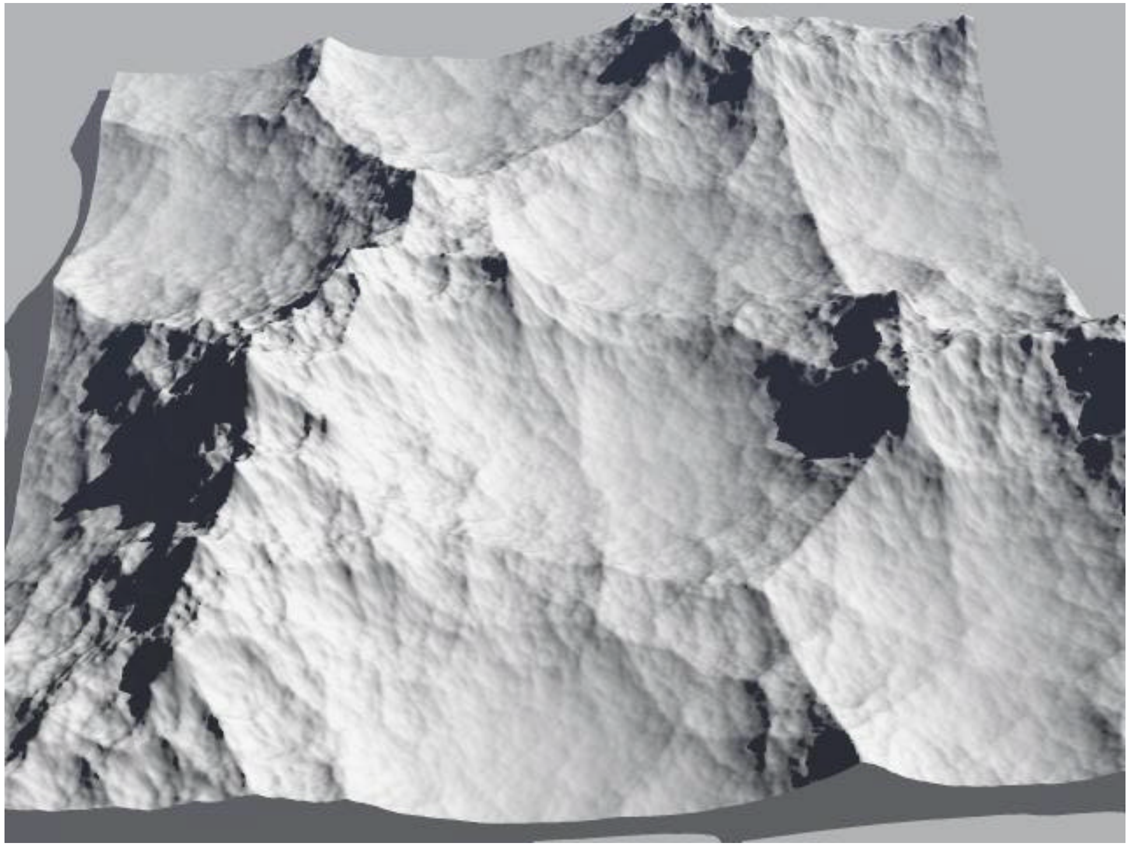
Heterogeneous fBm



The “ridges” terrain model from Bryce 4:
a hybrid multifractal made from one minus the absolute value of Perlin noise.

source: Ken Musgrave

Heterogeneous fBm



A hybrid multifractal made from Worley's Voronoi distance-squared basis.

source: Ken Musgrave

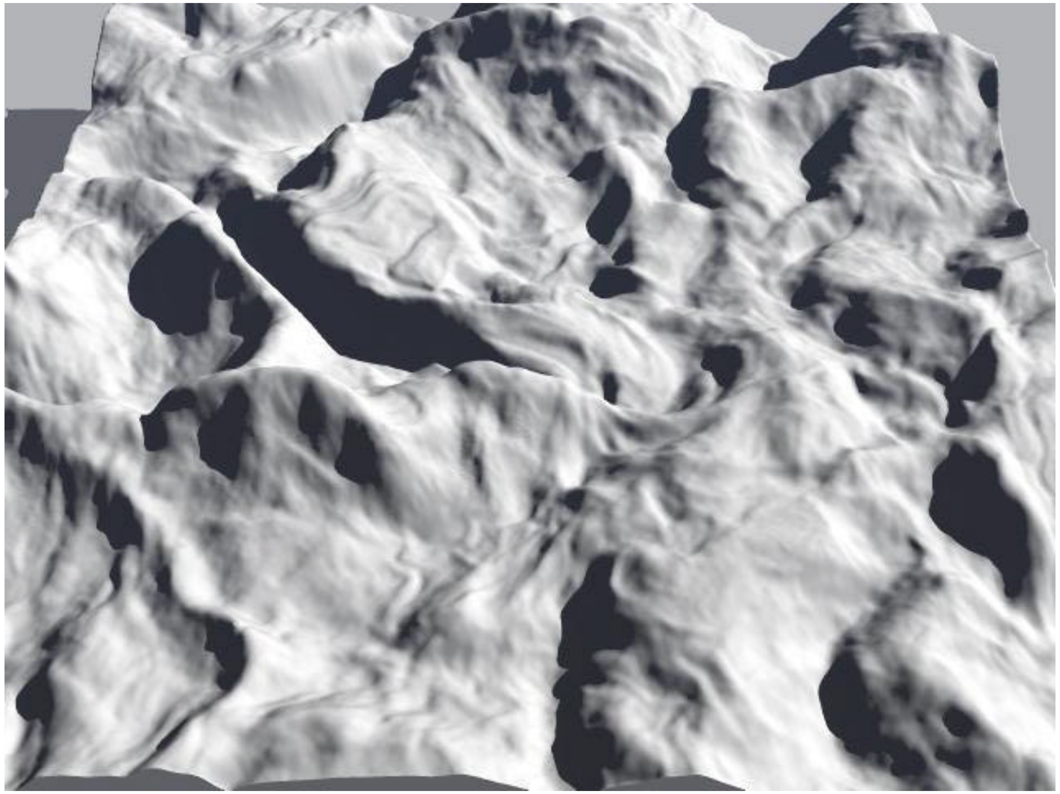
Heterogeneous fBm

A hybrid multifractal made from Worley's Voronoi distance basis.

source: Ken Musgrave

Domain Distortion

- Use fBm to perturb evaluation point of another noise function!

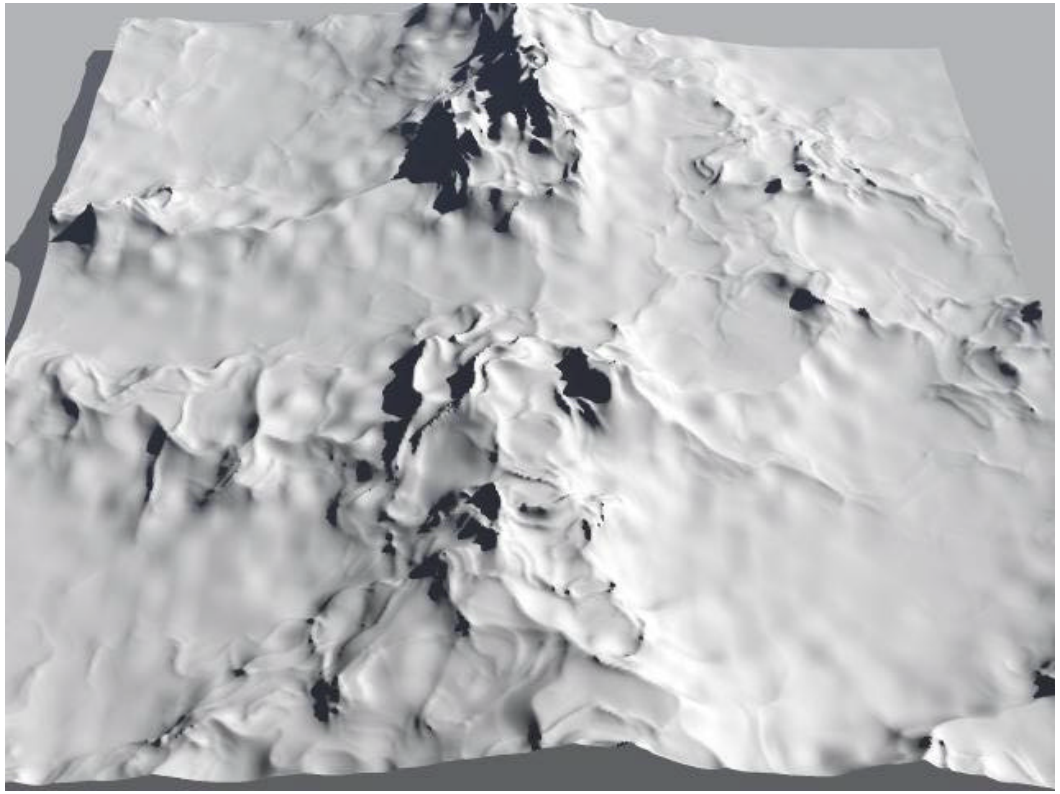


fBm distorted with fBm.

source: Ken Musgrave

Domain Distortion

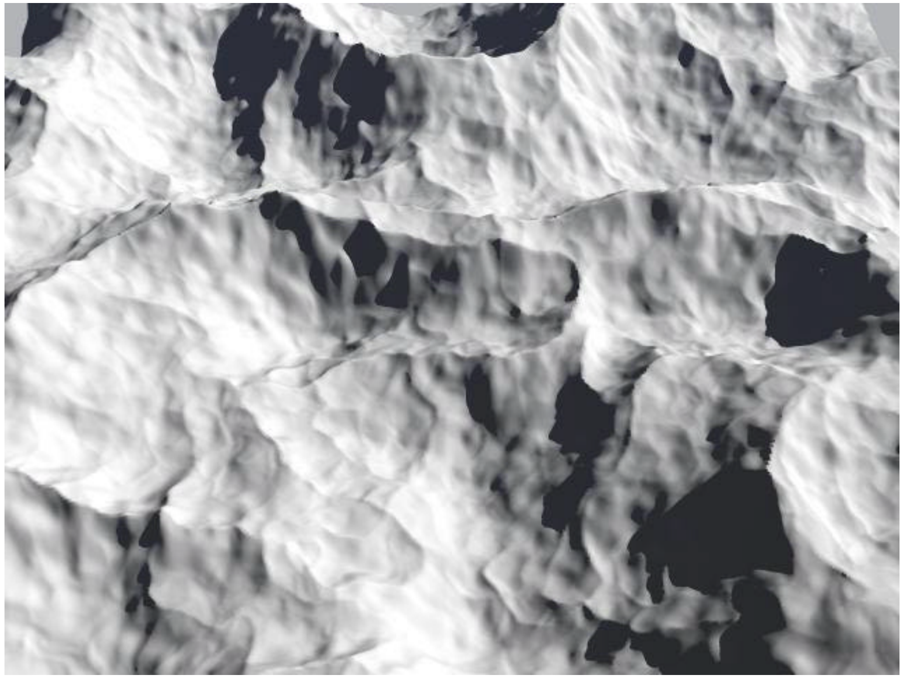
- Use fBm to perturb evaluation point of another noise function!



A sample of the "warped ridges" terrain model in Bryce 4:
the "ridges" model distorted with fBm.
source: Ken Musgrave

Domain Distortion

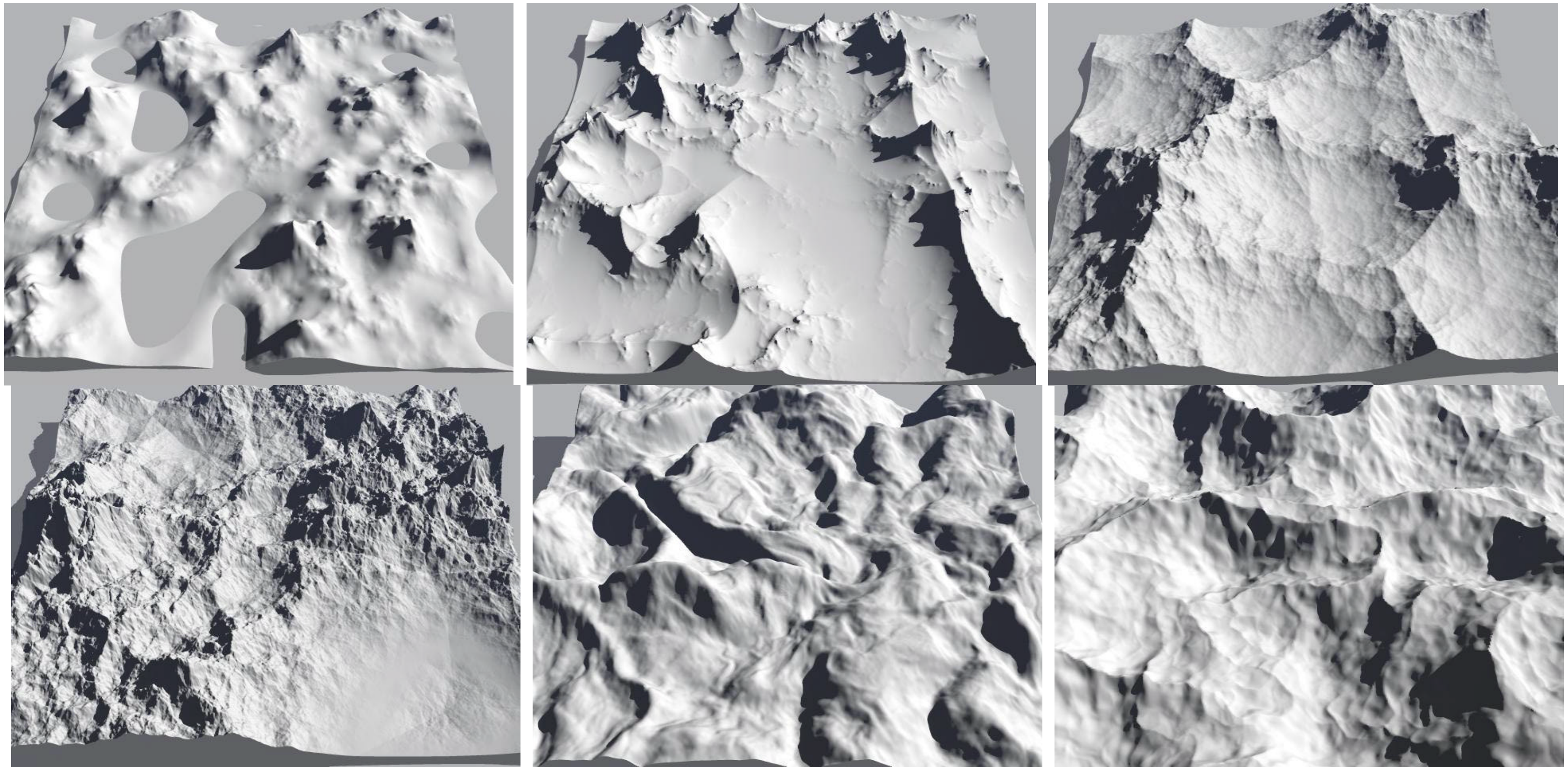
- Use fBm to perturb evaluation point of another noise function!



A sample of the "warped slickrock" terrain model in Bryce 4:
fBm constructed from one minus the absolute value of Perlin noise; distorted with fBm.

source: Ken Musgrave

Heterogeneous fBm



source: Ken Musgrave

Simulating Mountain Formation

- Real terrains undergo a number of weathering processes which result in erosion
  - wind, rain/rivers, freezing, etc.
- This is a complex physical process, however, we can obtain plausible results using some simple “ad hoc” rules.

Erosion + Vegetation

Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation

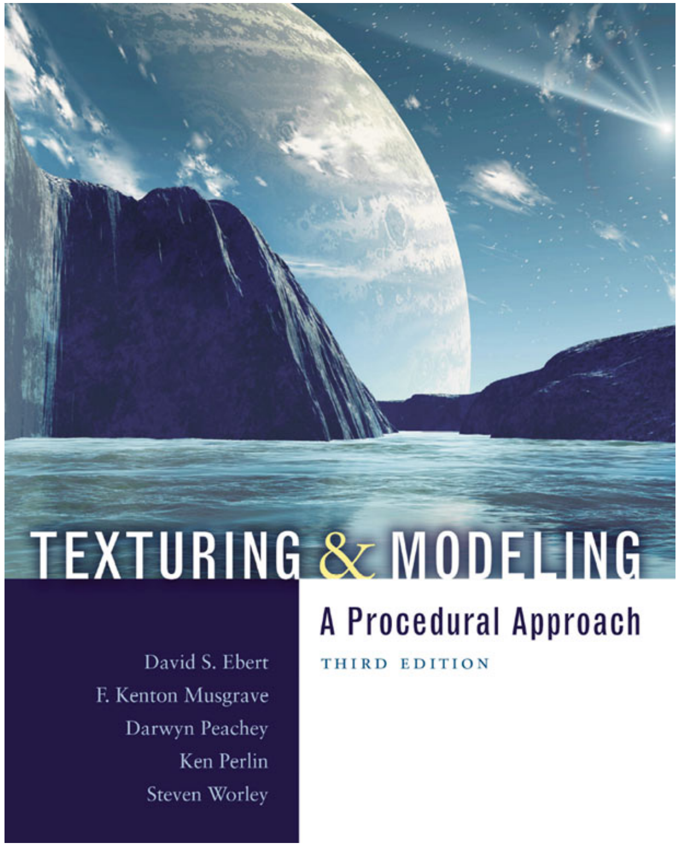
GUILLAUME CORDONNIER, Univ. Grenoble Alpes & CNRS (LJK) and Inria
ERIC GALIN, Univ Lyon, Université Lyon 2, CNRS, LIRIS
JAMES GAIN, University of Cape Town
BEDRICH BENES, Purdue University
ERIC GUÉRIN, Univ Lyon, INSA-Lyon, CNRS, LIRIS
ADRIEN PEYTAVIE, Univ Lyon, Université Lyon 1, CNRS, LIRIS
MARIE-PAULE CANI, Univ. Grenoble Alpes & CNRS (LJK) and Inria

We introduce a novel framework for interactive landscape authoring that supports bi-directional feedback between erosion and vegetation simulation. Vegetation and terrain erosion have strong mutual impact and their interplay influences the overall realism of virtual scenes. Despite their importance, these complex interactions have been neglected in computer graphics. Our framework overcomes this by simulating the effect of a variety of geomorphological agents and the mutual interaction between different material and vegetation layers, including rock, sand, humus, grass, shrubs, and trees. Users are able to exploit these interactions with an authoring interface that consistently shapes the terrain and populates it with details. Our method, validated through side-by-side comparison with real terrains, can be used not only to generate realistic static landscapes, but also to follow the temporal evolution of a landscape over a few centuries.

Final Procedural Scene for Inspiration

Created with Terragen

Literature


TEXTURING & MODELING
A Procedural Approach
THIRD EDITION
David S. Ebert
F. Kenton Musgrave
Darwyn Peachey
Ken Perlin
Steven Worley