

3D Modelling

Lecture 11 Terrains and Fractals

Patrick Min

Utrecht University

March 17, 2005

Outline

- 1 Introduction
- 2 Terrains
- 3 Fractals
- 4 Assignment 3



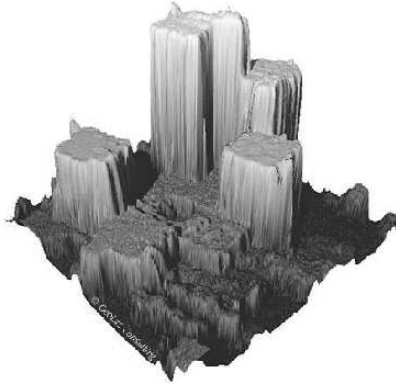
<http://www.fractal-landscapes.co.uk/>

Terrains

- we can generate a terrain using a simple regular grid mesh
- common name: **height field**
- slightly less common: **elevation grid**

Height fields

A height field from real range data:



A practical aside: VRML

VRML = Virtual Reality Modeling Language

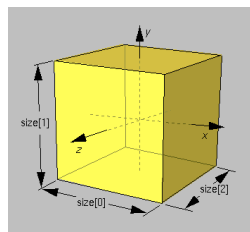
- scenegraph based 3D file format
- good web browser plugins available (e.g. Cortona)
- more on scenegraphs later

<http://www.web3d.org/x3d/specifications/vrml/vrml97>

VRML Example

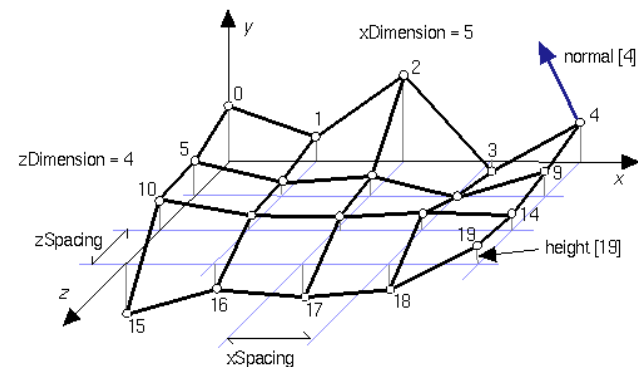
- VRML provides several geometric primitives
- for example, the Box, Sphere, and IndexedFaceSet (general mesh)
- a geometry node useful for terrains: ElevationGrid

```
Shape {
  geometry Box {
    size 2.0 0.5 2.0
  }
}
```



VRML ElevationGrid

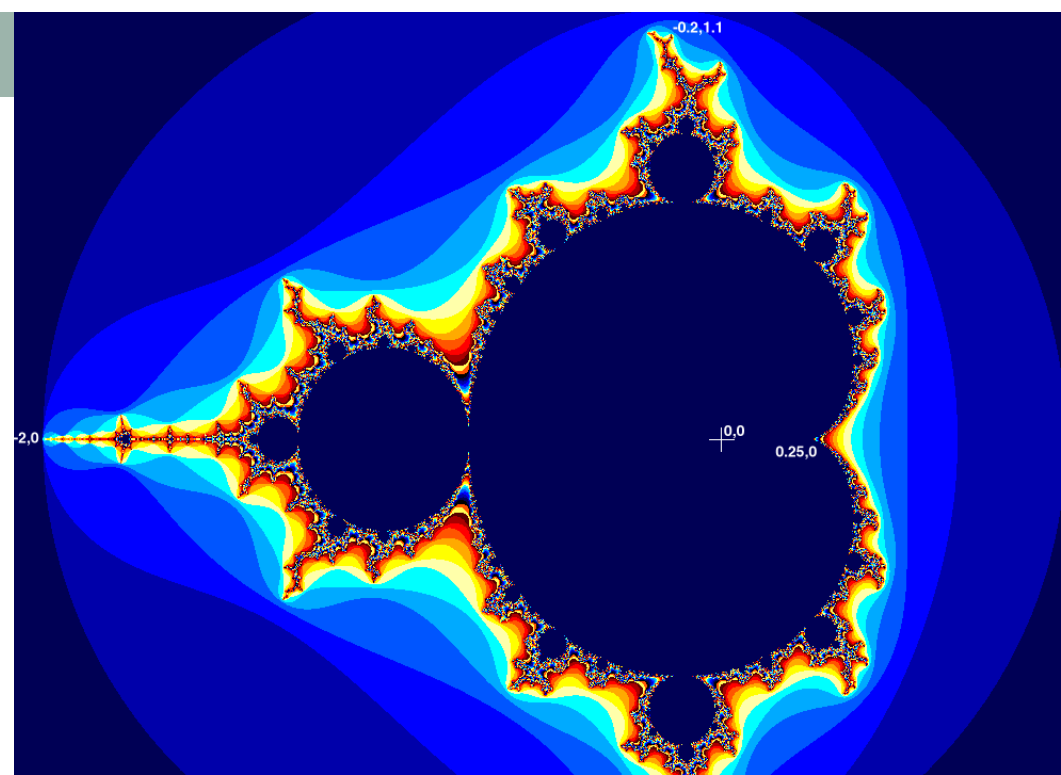
```
ElevationGrid {
  height [1.0, 0.5, 2.0, -0.25, 0.75,
  ...]
  xDimension 5
  zDimension 4
}
```



VRML IndexedFaceSet

```
IndexedFaceSet {
  coord Coordinate {
    point [-0.50 -0.50 0.50
           0.50 -0.50 0.50
           -0.50 0.50 0.50
           0.50 0.50 0.50
           -0.50 0.50 -0.50
           0.50 0.50 -0.50
           -0.50 -0.50 -0.50
           0.50 -0.50 -0.50]
    coordIndex [0 1 3 2 -1
                2 3 5 4 -1
                4 5 7 6 -1
                6 7 1 0 -1
                1 7 5 3 -1
                6 0 2 4 -1]
  }
}
```

The same cube as in the OFF example, now in VRML



the Mandelbrot set

= the set of complex values c that do not diverge under the squaring transformation:

- $z_0 = c$ with c a complex number
- $z_{n+1} = z_n^2 + z_0$, $n = 0, 1, 2, \dots$
- this is called a "self squaring fractal"

→ The boundary of the convergence region is a fractal

- compute this for the interval $Re(c) \in [-2.25, 0.75]$ and $Im(c) \in [-1.25, 1.25]$ to get the famous picture
- assign colour depending on when (at what value of n) the formula diverges

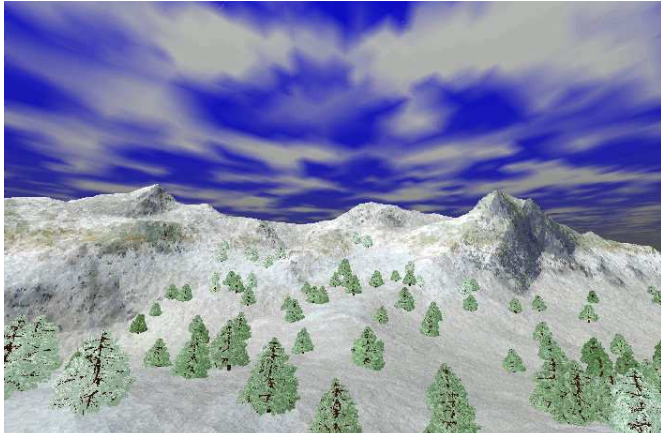
Fractals

A defining property: self-similar with infinite resolution

Fractals

are a form of **procedural modelling**: creating a model using a program

- Useful for describing natural 3D phenomena:
 - terrain, plants, clouds, water, feathers, fur, ...

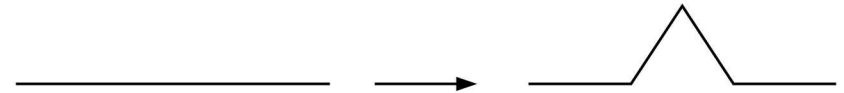


3D Modelling Lecture 11: Terrains and Fractals

Fractal self-similarity: deterministical vs. statistical

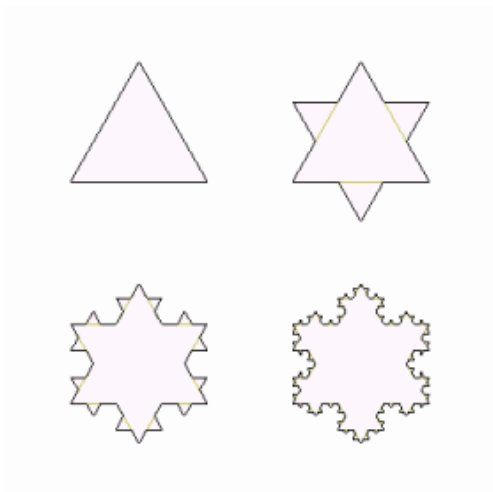
Deterministically self-similar, example:

- starting shape: a line segment
- production rule: add 1/3 "triangle" in the middle



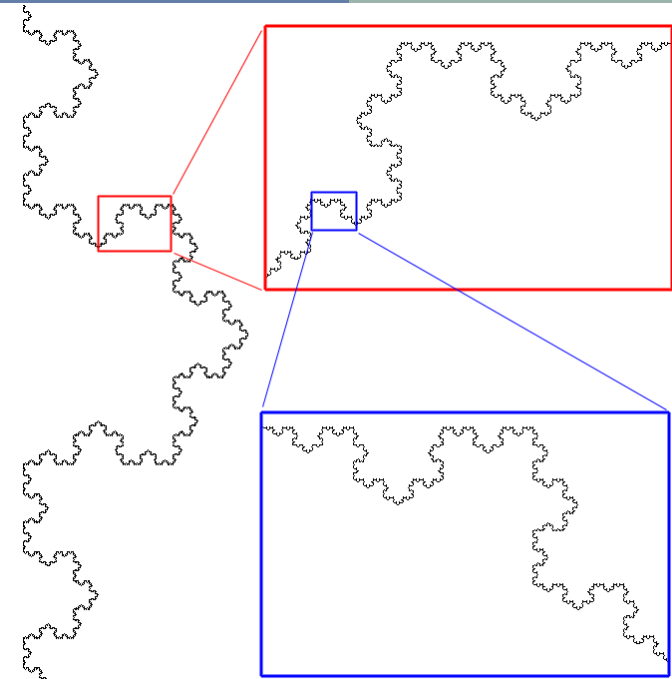
3D Modelling Lecture 11: Terrains and Fractals

the Koch Snowflake



Helge von Koch, 1904

3D Modelling Lecture 11: Terrains and Fractals



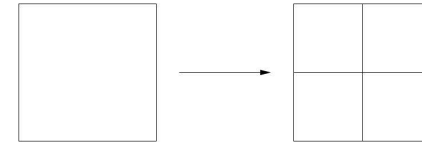
3D Modelling Lecture 11: Terrains and Fractals

Fractal dimension

- describes the detail variation in a fractal object
- specified as $D = a.b$, with a = the Euclidean dimension (topological dimension), and b = fraction of “filling up the next dimension”
- for fractal surfaces, typically $2 < D \leq 3$

Fractal dimension

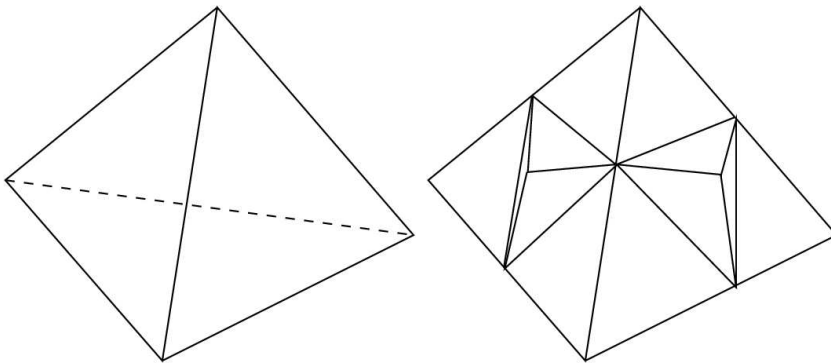
- Given a self-similar fractal with 1D scaling factor s , and a subdivision into n subparts at each step, $ns^D = 1$
- So $D = \frac{\ln n}{\ln(1/s)}$



- here $s = 1/2$, $n = 4$, so $D = 2$
- Koch snowflake fractal dimension $D = \ln 4 / \ln 3 \approx 1.26$

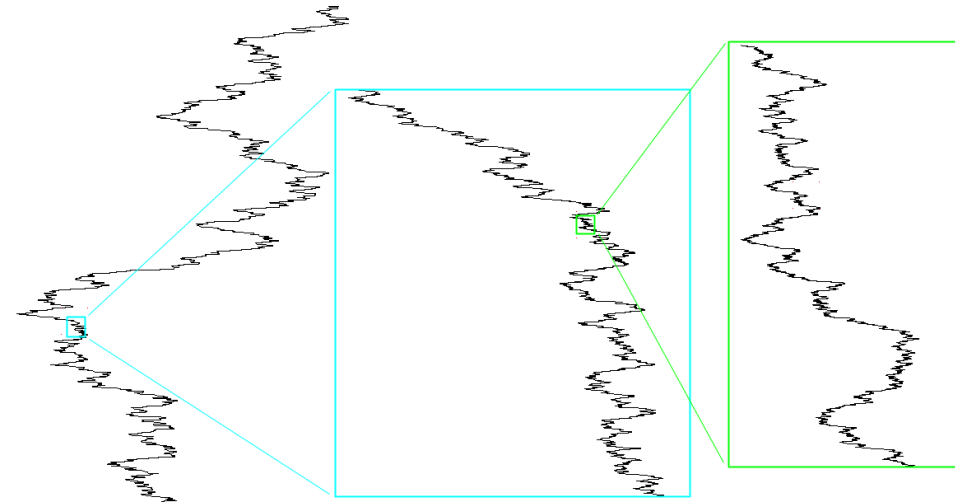
Fractal dimension

Example: subdividing a tetrahedron

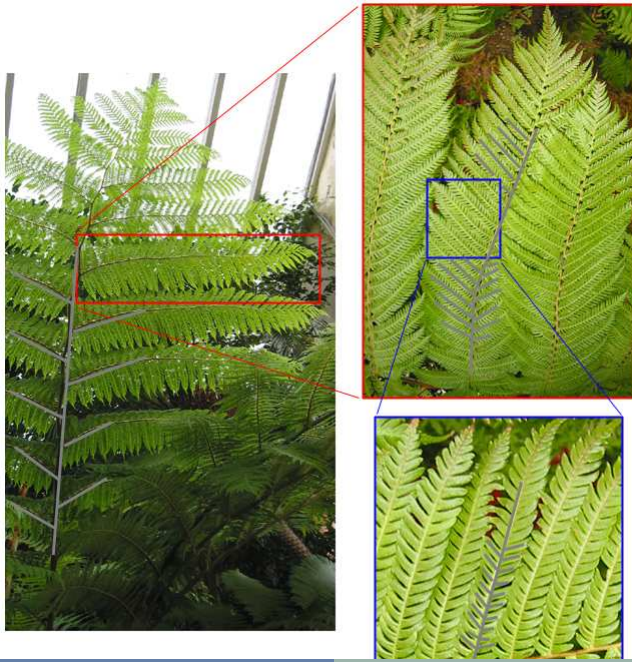


- here $s = 1/2$, $n = 6$ (each face is converted to 6 new faces)
- so $D = \ln 6 / \ln 2 \approx 2.58$

Statistical self-similarity



Terrains and fractals: assignment 3



In assignment 3, we will use fractals to generate terrains

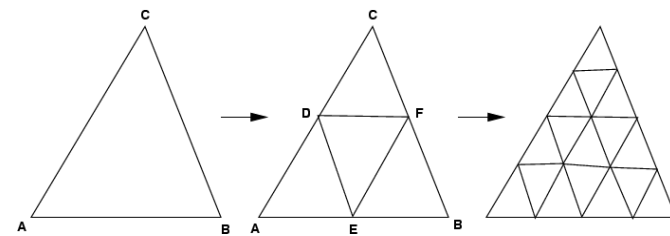
Assignment 3: Fractal Mountains

General information:

- you have to do this assignment with one partner
- (for the 4th assignment, you have to change partners)
- the deadline is in 2 weeks (Thursday, March 31st)
- submit:
 - Java or C/C++ executable
 - source
 - resulting model files
 - report

Fractal Mountains

First, write a program that can do triangle subdivisions:



Next, add support for the reading of OFF files

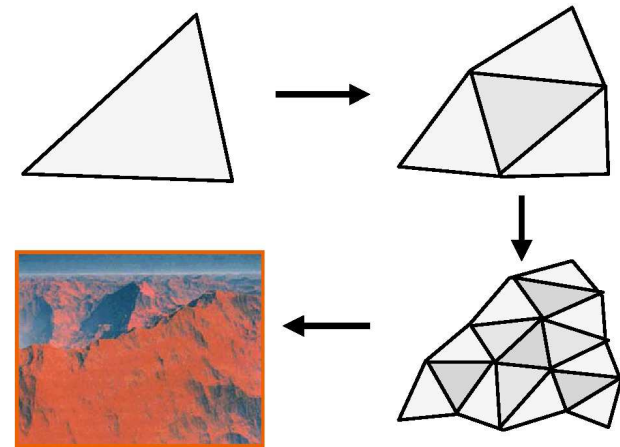
Fractal mountains

Then add randomness to the choice of midpoint locations:

- randomly perturb the location of each new midpoint
- choose an appropriate mesh data structure
- watch out for gaps in your mesh!

Fractal mountains

- add VRML (IndexedFaceSet) export to your program



Fractal mountains

For extra credit:

- visualize the mountain-growing on screen
- add height-dependent colouring
- add differently sized trees (depending on height)
- add an “erosion” procedure
- ...

→ More details about this assignment are on the course web page

Reading

- Watt, 4.1-4.3

Good fractal web sites:

- <http://astronomy.swin.edu.au/~pbourke/fractals/>
(the self-similarity examples and a few other images are from this website)
- <http://www.fractal-landscapes.co.uk/>