Procedural Modeling

Generating complexity

- By hand
  - Expensive (Hollywood)
- Scanning real world (Laser scanners)
  - Ground truth result, but need the object
- Alternative techniques:
  - Physics-based modeling
  - Procedural modeling
  - Volume graphics
  - Image-based modeling

Procedural Modeling

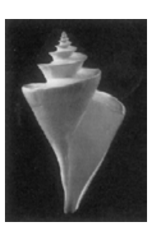
- Goal:
  - Describe 3D models algorithmically
- Best for models resulting from ...
  - Repeating processes
  - Self-similar processes
  - Random processes
- Advantages:
  - Automatic generation
  - Concise representation
  - Parameterized classes of models

Procedural Models

- Subdivision Surfaces.
- Sweeps.
- Fractals.
- Grammar-based modeling.
- Cellular Automata.

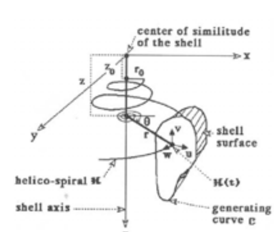
Sweeps

- Create 3D polygonal surface models of seashells.

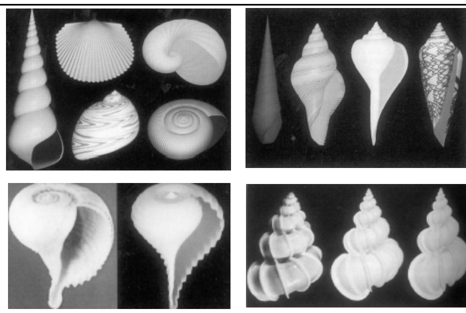


Sweeps

- Sweep generating curve around helico-spiral axis.



Seashell-examples



Fractals--Motivation

- Natural objects such as mountains and clouds have irregular or fragmented features which are not perfectly smooth.
- Procedure-based fractal methods are well suited for modeling these natural phenomena.
- Fractals are typically generated by recursive procedures.

Fractals

Useful for describing natural 3D phenomenon

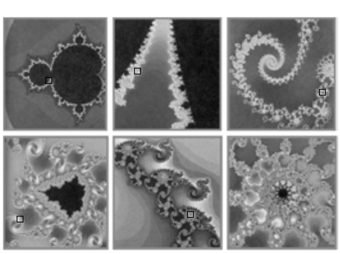
- Terrain, Plants, Clouds, Water, Feathers, Fur, etc.

Fractals

Two key properties:

1. Fractals exhibit infinite detail
   - as we zoom in around a point, we see more and more detail.
   - this is essentially true of the real world as well.
   - but we could generate infinite detail with random numbers.
2. Fractals also exhibit self-similarity
   - small-scale features look like large scale features.
   - sub-parts are scaled down copies of the whole. For instance, by affine transformations.
   - often consider only statistical, rather than exact, self-similarity.

Fractals



Mandelbrot set

Fractal Generation

- Deterministically self-similar fractals
  - Parts are scaled copies of original
- Statistically self-similar fractals
  - Parts have same statistical properties as original

Fractal Generation

- Deterministically self-similar fractals
  - Parts are scaled copies of original
- Statistically self-similar fractals
  - Parts have same statistical properties as original

Deterministic Fractal Generation

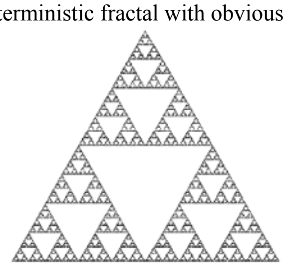
General Procedure:

- Starting with an initial shape.
- Replace subparts with scaled copy of original.

Deterministic Fractal Generation

The Sierpinski Gasket:

This is a deterministic fractal with obvious self-similarity.

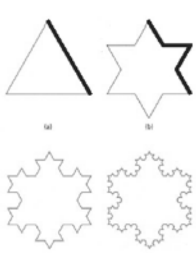


Sample Code

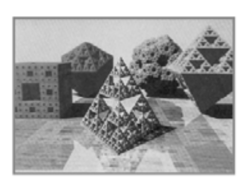
```
sierpinski(Vec2& a, Vec2& b, Vec2& c, int depth)
{
  if( depth>0 )
  {
    Vec2 v0 = (a+b)/2;
    Vec2 v1 = (a+c)/2;
    Vec2 v2 = (b+c)/2;

    sierpinski(a, v0, v1, depth-1);
    sierpinski(b, v1, v2, depth-1);
    sierpinski(c, v2, v0, depth-1);
  }
  else
    // Output triangle a,b,c in black
}
```


Deterministic Fractal Generation



Deterministic Fractals



Fractal dimensions-example

- the Koch snowflake:

$s=N^{1/d}$, where d is the fractal dimension.

Here, $N=4$, $s=3$, so, $d=\ln 4/\ln 3=1.26$.

Fractal Dimension

Measures the "roughness" of the object

- more jagged objects have larger fractal dimensions
- and they fill more of space
  - e.g., a curve with dimension 1.99 almost covers the plane

We haven't really said how to compute these dimensions

- we could do it quite easily for the Koch snowflake
- but for more general objects, it's quite difficult
- there are many approximations available
- one of them is the box-covering method
  - subdivide into a number of small boxes with scaling factor s
  - count number of boxes n covered by object
  - use these to estimate the fractal dimension

Fractal Dimension

- The fractal dimension of an object is always greater than the corresponding Euclidean dimension, which is simply the least number of parameters needed to specify the object. E.g:
  - A Euclidean curve is one-dimensional.
  - A Euclidean surface is two-dimensional.
  - A Euclidean solid is three-dimensional.
- For a fractal curve that stays completely within a 2D plane, the fractal dimension D is greater than one.
  - The closer to one D is, the smoother the fractal curve.
  - If $D = 2$, we have a peano curve, that completely fills a finite region of 2D plane.
  - If $2< D <3$, , the curve self-intersects and the area could be covered an infinite number of times.
- Spatial fractal curves fills a volume of space if D=3, self-intersecting if D > 3.

Fractal Generation

- Deterministically self-similar fractals
  - Parts are scaled copies of original
- Statistically self-similar fractals
  - Parts have same statistical properties as original

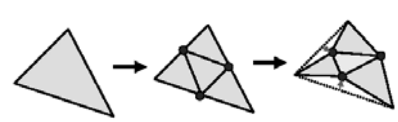
Adding Randomness to Fractals

- **Deterministic fractals are certainly nice**
  - infinite self-similar detail can be handy
- **But we don't want all our objects to look the same**
  - suppose we want to synthesize terrain surfaces
  - we want terrains that are distinct
- **To address this problem, we use randomness**
  - also referred to as "noise" or sometimes "chaos"

Statistical Fractal Generation

General procedure:

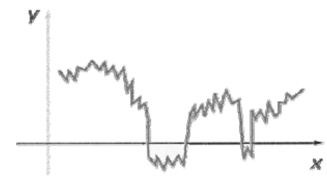
- start with a shape
- replace subparts with a self-similar random pattern



Fractional Brownian Motion

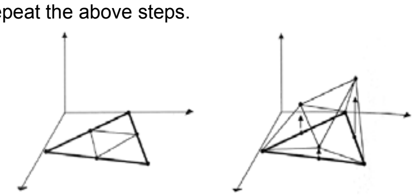
- **This is one powerful class of randomized fractals**
  - it's a kind of random walk
  - start at some point
  - generate a segment in a random direction with random length
  - repeat for next segment
  - need to provide control of fractal dimension
- **In practice, this is not easy**
  - functions with the right statistical properties are expensive
  - typically involves Fast Fourier Transform computations

Fractal Brownian Motion

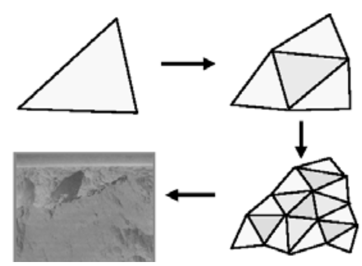


Random Midpoint Displacement

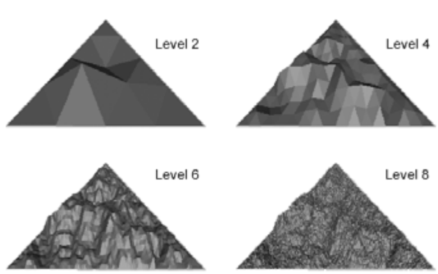
- **A more tractable approximation of fractional Brownian motion.**
  - start with some initial figure (e.g., line or triangle)
  - split at midpoints & add random displacement.
  - Repeat the above steps.



Fractal Mountain

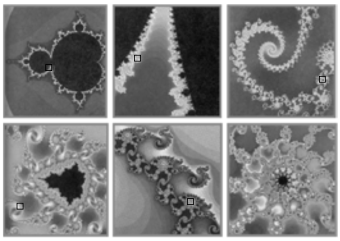


Fractal Terrain



Mountain

Mandelbrot Set



Mandelbrot Set

- Is the set of complex values $z=x+iy$ that do not diverge under the squaring transformation:

  $z_0 = z$

  $z_k = (z_{k-1})^2 + z_0,\ k = 1, 2, 3, \ldots$
- The color is decided by the number of iterations needed to test for the convergence of the current value.