FRACTAL-GEOMETRY METHODS

Natural objects, such as mountains and clouds, don't have smooth surface or regular shapes instead they have fragmented features, and Euclidean methods do not realistically model these objects.

Natural objects can be realistically described with fractal-geometry methods, where procedures rather than equations are used to model objects.

In computer graphics, fractal methods are used to generate displays of natural objects and visualizations of various mathematical and physical systems.

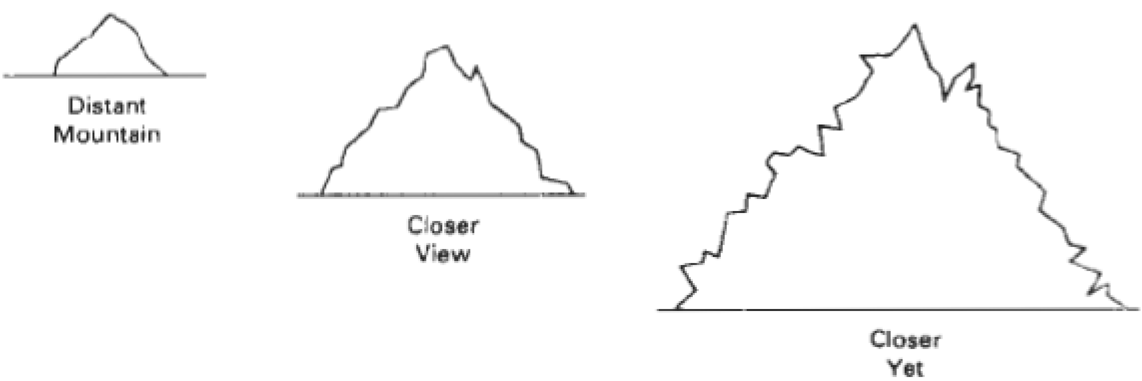
A fractal object has two basic characteristics:

i. Infinite detail at every point
ii. Certain self-similarity between the object parts and the overall features of the object

We describe a fractal object with a procedure that specifies a repeated operation for producing the detail in the object subparts. Natural objects are represented with procedures that theoretically repeat an infinite number of times. Graphics displays of natural objects are, of course, generated with a finite number of steps.

If we zoom in on a continuous Euclidean shape, no matter how complicated, we can eventually get the zoomed-in view to smooth out.

But if we zoom in on a fractal object, we continue to see as much detail in the magnification as we did in the original view.



Zooming in on a graphics display of a fractal object is obtained by selecting a smaller window and **repeating the fractal procedures** to generate the detail in the new window.

A consequence of the infinite detail of a fractal object is that it has no definite size. As we consider more and more detail, the size of an object tends to infinity, but the coordinate extents of the object remain bound within a finite region of space.

The amount of variation in the object detail with a number called the ***fractal dimension.***

In graphics applications, fractal representations are used to model terrain, clouds, water, trees and other plants, feathers, fur, and various surface textures, and just to make pretty patterns.

Fractal-Generation Procedures

A fractal object is generated by **repeatedly applying a specified transformation function** to points within a region of space.

If $\mathbf{P_0}$ = (x0, y0, z0) is a selected initial point, each iteration of a transformation function F generates successive levels of detail with the calculations

In general, the transformation function can be applied to a specified point set, or we could apply the transformation function to an initial **set** of primitives, such as straight lines, curves, color areas, surfaces, and solid objects.

Also, we can use either deterministic or random generation procedures at each iteration. The transformation function may be defined in terms of geometric transformations (scaling, translation, rotation), or it can be set up with nonlinear coordinate transformations and decision parameters.

Although fractal objects, by definition, contain infinite detail, we apply the transformation function a finite number of times. Therefore, the objects we display actually have finite dimensions.

A procedural representation approaches a "true" fractal as the number of transformations is increased to produce more and more detail.

The **amount of detail** included in the final graphical display of an object **depends on the number of iterations performed** and the resolution of the display system

We cannot display detail variations that are smaller than the size of a pixel.

To **see more** of the object detail, we **zoom in on selected sections** and **repeat the transformation function** iterations.

Classification of Fractals

i. Self-similar fractals

Self-similar fractals have parts that are scaled-down versions of the entire object.

Starting with an initial shape, the object subparts are constructed by apply a scaling parameter ‘s’ to the overall shape.

The same scaling factors can be used for all subparts, or different scaling factors can be used for different scaled-down parts of the object.

If random variations are applied to the scaled-down subparts, the fractal is said to be statistically self-similar. The parts then have the same statistical properties.

Statistically self-similar fractals are commonly used to model trees, shrubs, and other plants.

ii. Self-affine fractals

Self-affine fractals have parts that are formed with different scaling parameters, $\mathbf{s_x}$, $\mathbf{s_y}$, $\mathbf{s_z}$ in different coordinate directions.

Random variations can also be used to obtain statistically self-affine fractals.

Terrain, water, and clouds are typically modeled with statistically self-affine fractal construction methods.

iii. Invariant fractal

Invariant fractal sets are formed with nonlinear transformations.

This class of fractals includes self-squaring fractals, such as the Mandelbrot set, which are formed with squaring functions in complex space; and self-inverse fractals, formed with inversion procedures.

Fractal Dimension

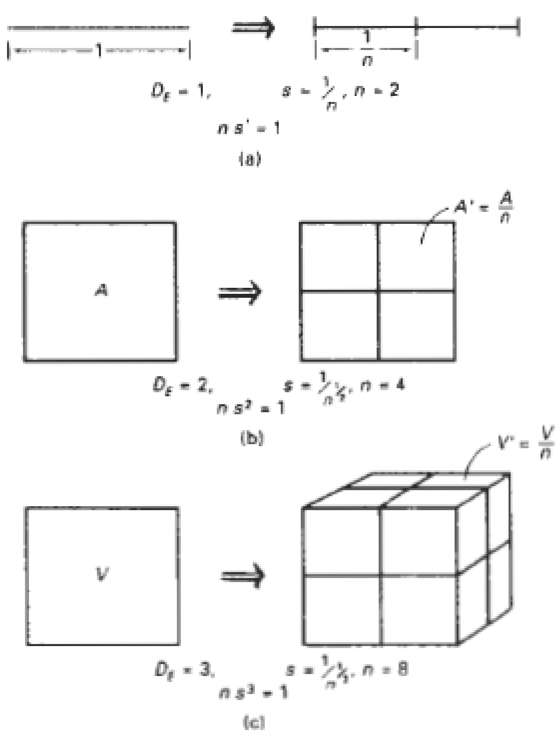
The detail variation in a fractal object can be described with a number D, called the **fractal dimension**, which is a measure of the roughness, or fragmentation, of the object.

More jagged-looking objects have larger fractal dimensions.

Iterative procedures can be set up to generate fractal objects using a given value for the fractal dimension **D.**

An expression for the fractal dimension of a self-similar fractal, constructed with a single scalar factor **s**, is obtained by analogy with the subdivision of a Euclidean object.

Suppose an object is composed of clay or elastic. If it is deformed into a line then its topological dimension D_t is 1, if it is deformed into a plane or a disk then the topological Dimension is 2 and if it is deformed into a ball or a cube then its topological dimension is 3



The relationships between the scaling factor s; and the number of subparts n for subdivision of a unit straight-line segment, A square, and a cube can be shown

If take a line segment having length L and divide it into n pieces each piece having length ‘*l*’

The scaling factor s = 1/n

If it is broken into two pieces, $s^1 = 1/2$, the unit line segment is divided into two equal-length subparts.

Similarly, the square is divided into four equal-area subparts, $s^2 = 1/4$

The cube is divided into eight equal-volume subparts $s^3 = 1/8$

For each of these objects, the relationship between the number of subparts and the scaling factor is $n \cdot s^D = 1$. In analogy with Euclidean objects, the fractal dimension D for self-similar objects can be obtained from

$$n \cdot s^D = 1.$$

Solving this expression for D, the fractal similarity dimension, we have

$$D = (\log n) / (\log (1/s))$$

The fractal dimension gives the measure of the roughness or fragmentation of objects and is always greater than the corresponding topological dimension

For a self-similar fractal constructed with different scaling factors for the different parts, the fractal similarity dimension is obtained from the implicit relationship where $\mathbf{s_k}$ is the scaling factor for subpart number k.

Geometric Construction of Deterministic Self-Similar Fractals

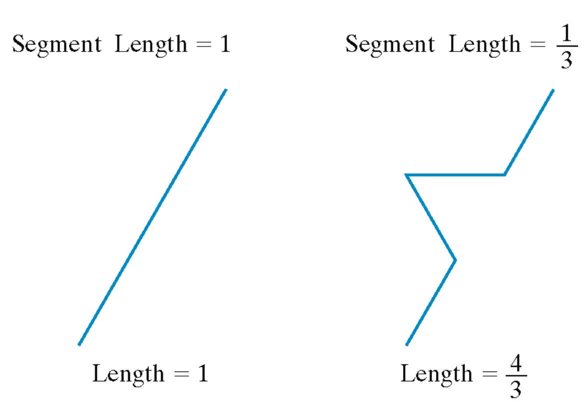
To geometrically construct a deterministic (nonrandom) self-similar fractal, we start with a given geometric shape, called the ***initiator.*** Subparts of the initiator are then replaced with a pattern, called the generator.

Koch Curve

1. Begin with a line segment

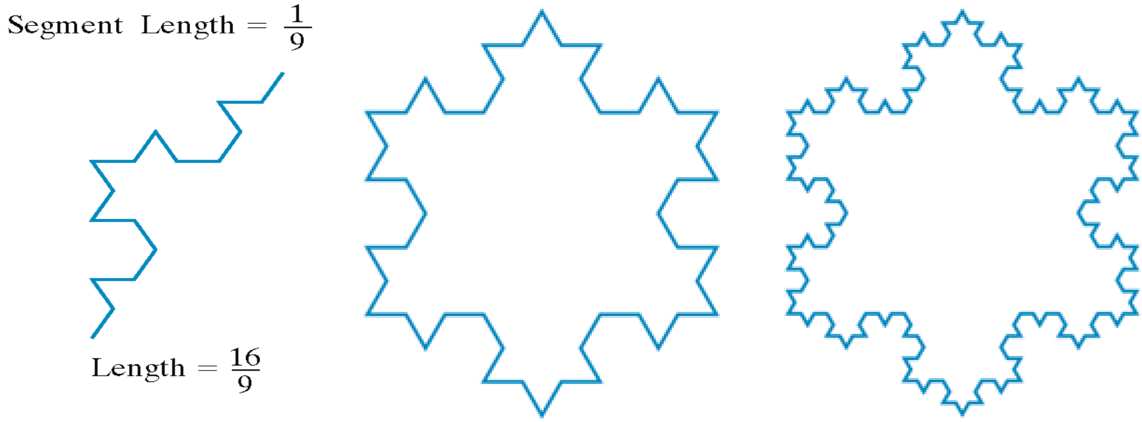
2. Divide it into thirds i.e. scaling factor = 1/3 and replace the center third by the two adjacent sides of an equilateral triangle

3. There are now 4 equal length segments each 1/3 the original length , so the new curve has 4/3 length of the original length



5. Repeat the process for each of the four segments

6. The cure has gained more wiggles and its length now is 16/9 times the original



7. Repeat this indefinitely and the length every time increases by 4/3 factor. The curve will be infinite but is folded in lots of tiny wiggles

8. Its topological dimension is 1 and it's Fractal dimension can be calculated as follows

> We have to assemble 4 such curves to make the original curve so N = 4 and Scaling factor S = 3 as each segment has 1/3 the original segment length
>
> So the fractal dimension is D = (log 4) / (log (3)) = 1.2618

Peano Curve

It is also called space filling curve and is used for filling two dimensional object e.g. a square

Steps to generate a Peano curve

1. Sub-divide a square into 4 quadrants and draw the curve which connects the center points of each

2. Further subdivide each of the quadrants and connect the centers of each of these finer divisions before moving to the next major quadrant

3. The third approximation subdivides again it again connects the centers of the finest level before stepping to the next level of detail

The above process is indefinitely continued depending upon the degree of roughness of the curve generated

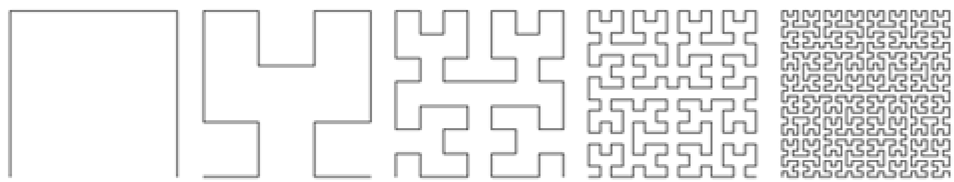
- The curve never crosses itself
- There is no limit to the subdivision
- The curve fills the square
- With each subdivision the length increases by a factor of 4 and since there is no limit to subdivision there is no limit to the length
- The curve constructed is topologically equivalent to the line $D_t = 1$ but it is so twisted and folded that it exactly fills up a square

- The Fractal dimension of the curve:

  At each subdivision the scale changes by 2 but length changes by 4

  For the square it takes 4 curves of half scale to build the full sized object so the Dimension is given by

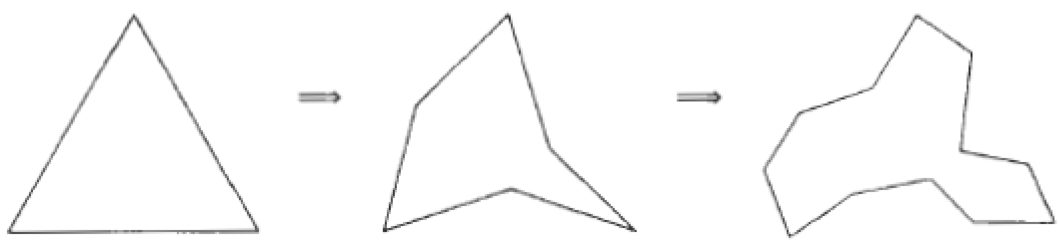
  D = (log 4) / (log (2)) So, the Fractal Dimension is 2 and the Topological Dimension is 1



Geometric Construction of Statistically Self-Similar Fractals

One way to introduce some randomness into the geometric construction of a self-similar fractal is to choose a generator randomly at each step from a set of predefined shapes. Another way to generate random self-similar objects is to compute coordinate displacements randomly.

A random snowflake pattern can be created by selecting a random, midpoint displacement distance at each step.

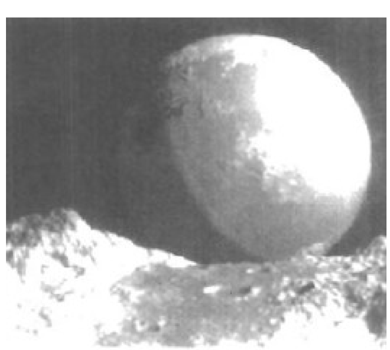


A modified "snowflake" pattern using random midpoint displacement.

Affine Fractal-Construction Methods

Highly realistic representations for terrain and other natural objects can be obtained using affine fractal methods that model object features as ***fractional Brownian motion.***

This is an extension of standard Brownian motion, a form of "random walk**",** that **describes** the erratic, zigzag movement of particles in a gas or otter fluid.



Starting from a given position, we generate a straight-line segment in a random direction and with a random length. We then move to the endpoint of the first line segment