SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | Approved by AICTE

COMPUTER GRAPHICS & MULTIMEDIA SYSTEMS

SCS1302

UNIT III – PART III

Syllabus

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY — FACULTY OF COMPUTING

SCS1302	COMPUTER GRAPHICS AND MULTIMEDIA SYSTEMS	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- To gain knowledge to develop, design and implement two and three dimensional graphical structures.
- To enable students to acquire knowledge of Multimedia compression and animations.
- To learn creation, Management and Transmission of Multimedia objects.

UNIT 1 BASICS OF COMPUTER GRAPHICS — 9 Hrs.

Output Primitives: Survey of computer graphics - Overview of graphics systems - Line drawing algorithm - Circle drawing algorithm - Curve drawing algorithm - Attributes of output primitives - Anti-aliasing.

UNIT 2 2D TRANSFORMATIONS AND VIEWING — 8 Hrs.

Basic two dimensional transformations - Other transformations - 2D and 3D viewing - Line clipping - Polygon clipping - Logical classification - Input functions - Interactive picture construction techniques.

UNIT 3 3D CONCEPTS AND CURVES — 10 Hrs.

3D object representation methods - B-REP , sweep representations, Three dimensional transformations. Curve generation - cubic splines, Beziers, blending of curves- other interpolation techniques, Displaying Curves and Surfaces, Shape description requirement, parametric function. Three dimensional concepts – Introduction - Fractals and self similarity- Successive refinement of curves, Koch curve and peano curves.

Syllabus

UNIT 4 METHODS AND MODELS **8 Hrs.**

Visible surface detection methods - Illumination models - Halftone patterns - Dithering techniques - Polygon rendering methods - Ray tracing methods - Color models and color applications.

UNIT 5 MULTIMEDIA BASICS AND TOOLS **10 Hrs.**

Introduction to multimedia - Compression & Decompression - Data & File Format standards - Digital voice and audio - Video image and animation. Introduction to Photoshop - Workplace - Tools - Navigating window - Importing and exporting images - Operations on Images - resize, crop, and rotate - Introduction to Flash - Elements of flash document - Drawing tools - Flash animations - Importing and exporting - Adding sounds - Publishing flash movies - Basic action scripts - GoTo, Play, Stop, Tell Target

Max. 45 Hours

TEXT / REFERENCE BOOKS

1. Donald Hearn, Pauline Baker M., "Computer Graphics", 2nd Edition, Prentice Hall, 1994.
2. Tay Vaughan ,"Multimedia", 5th Edition, Tata McGraw Hill, 2001.
3 Ze-Nian Li, Mark S. Drew ,"Fundamentals of Multimedia", Prentice Hall of India, 2004.
4 D. McClelland, L.U.Fuller ,"Photoshop CS2 Bible", Wiley Publishing, 2005.
5 James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice, 2nd
6 Edition in C, Audison Wesley, ISBN - 981 -235-974-5
7 William M. Newman, Roberet F. Sproull, " Principles of Interactive Computer Graphics", Second Edition, Tata McGraw-Hill Edition.

Course Objective(CO)

CO1: Construct lines and circles for the given input.

CO2: Apply 2D transformation techniques to transform the shapes to fit them as per the picture definition.

CO3: Construct splines, curves and perform 3D transformations

CO4: Apply colour and transformation techniques for various applications.

CO5: Analyse the fundamentals of animation, virtual reality, and underlying technologies.

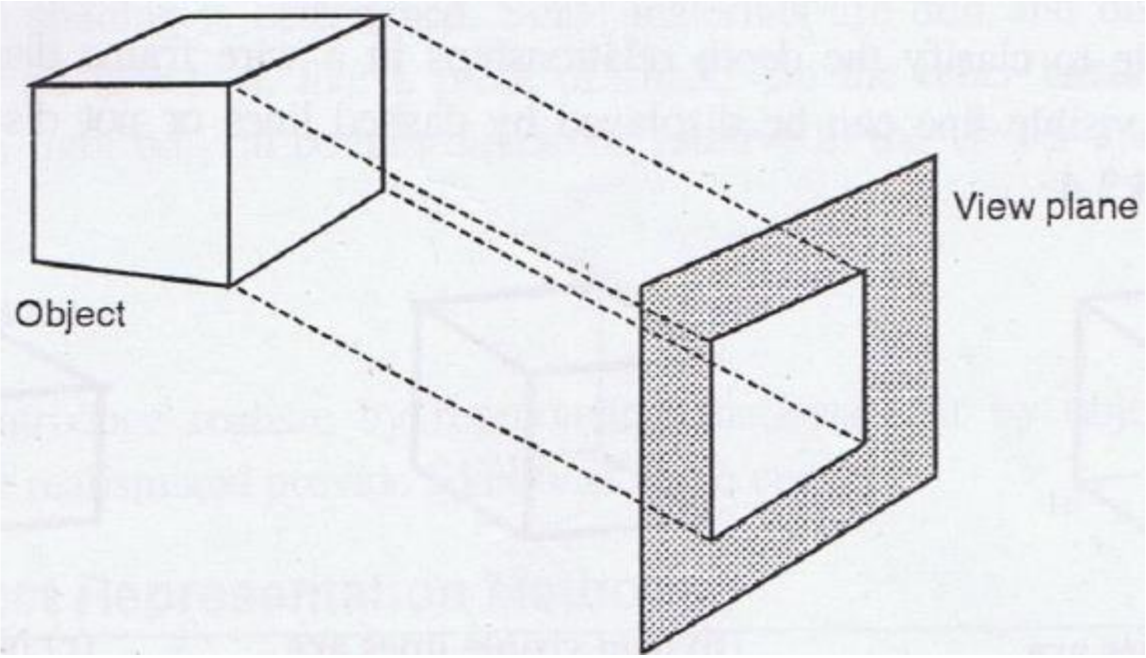
CO6: Develop photo shop applications

Three dimensional concepts

- Parallel Projection
- Perspective Projection
- Depth Cueing
- Visible Line and Surface Identification
- Surface Rendering
- Exploded and Cutaway Views
- Three-Dimensional and Stereoscopic Views

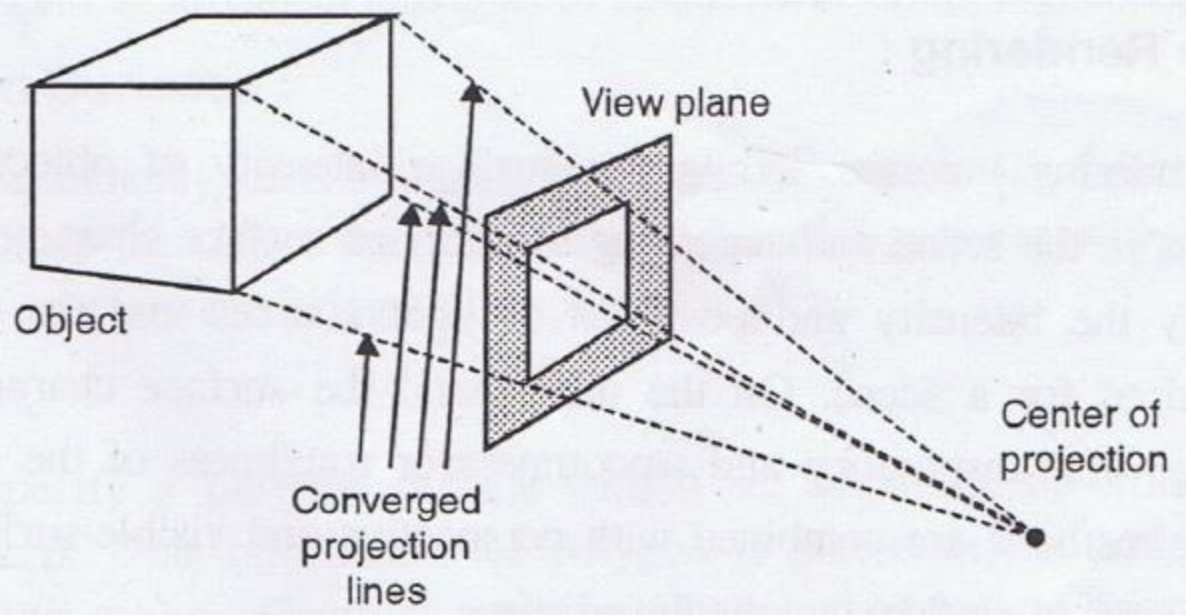
Parallel Projection

In this method a view plane is used. z co-ordinate is discarded. The 3 d view is constructed by extending lines from each vertex on the object until they intersect the view plane. Then connect the projected vertices by line segments which correspond to connections on the original object.



Perspective Projection

Here the lines of projection are not parallel. Instead, they all converge at a single point called the `center of projection' or `projection reference point'. The object positions are transformed to the view plane along these converged projection lines. In this method, Objects farther from the viewing position appear smaller.

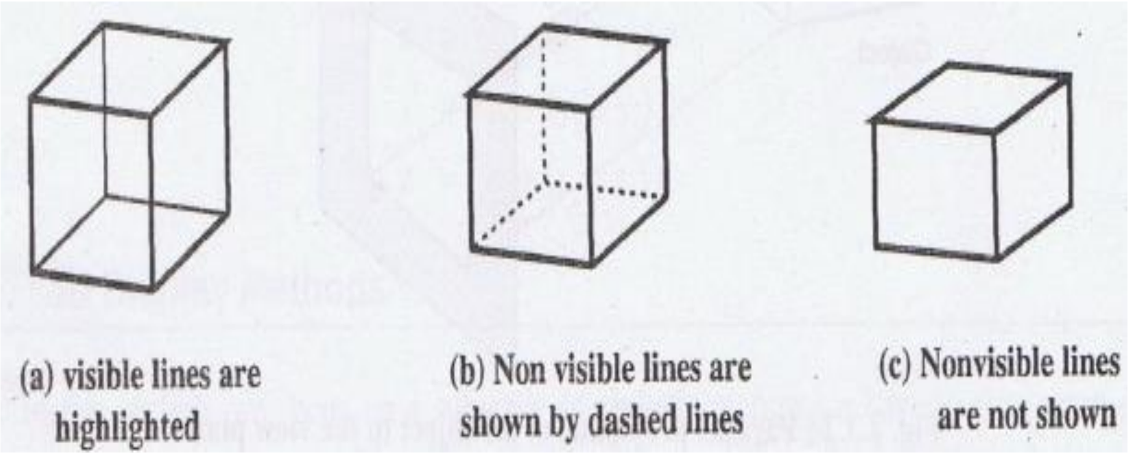


Depth Cueing

Depth information is added. The depth of an object can be represented by the intensity of the image. The parts of the objects closest to the viewing position are displayed with the highest intensities. Objects farther away are displayed with decreasing intensities.

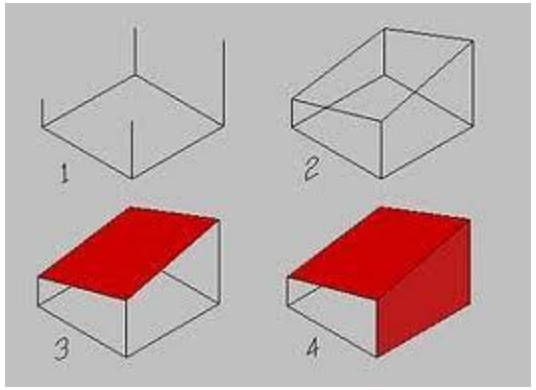
Visible Line and Surface Identification

Visible lines are displayed in different color. Invisible lines either displayed in dashed lines or not at all displayed. Removing invisible lines also removes the info about the backside of the object. Surface rendering can be applied for the visible surfaces, so that hidden surfaces will become obscured.



Surface Rendering

Surface intensity of objects will be according to the lighting conditions in the scene and according to assigned surface characteristics. This method is usually combined with the previous method to attain a degree of realism.

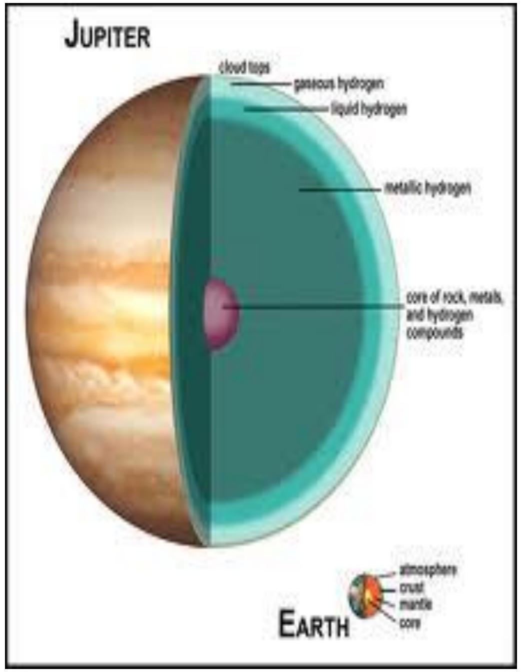


Exploded and Cutaway Views

- To show the internal details, we can define the object in hierarchical structures.

Exploded view

Cutaway View

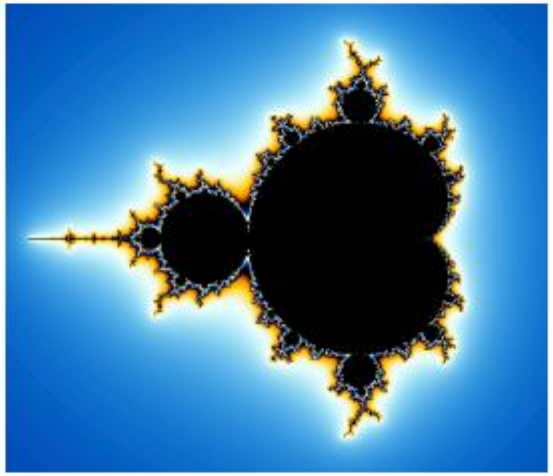


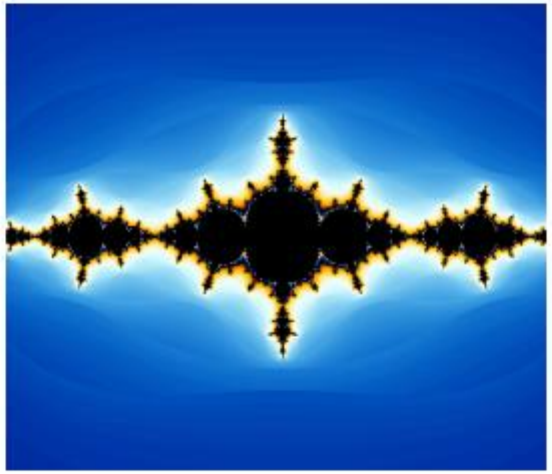
Fractals

- Fractals are very complex pictures generated by a computer from a single formula. They are created using iterations. This means one formula is repeated with slightly different values over and over again, taking into account the results from the previous iteration.

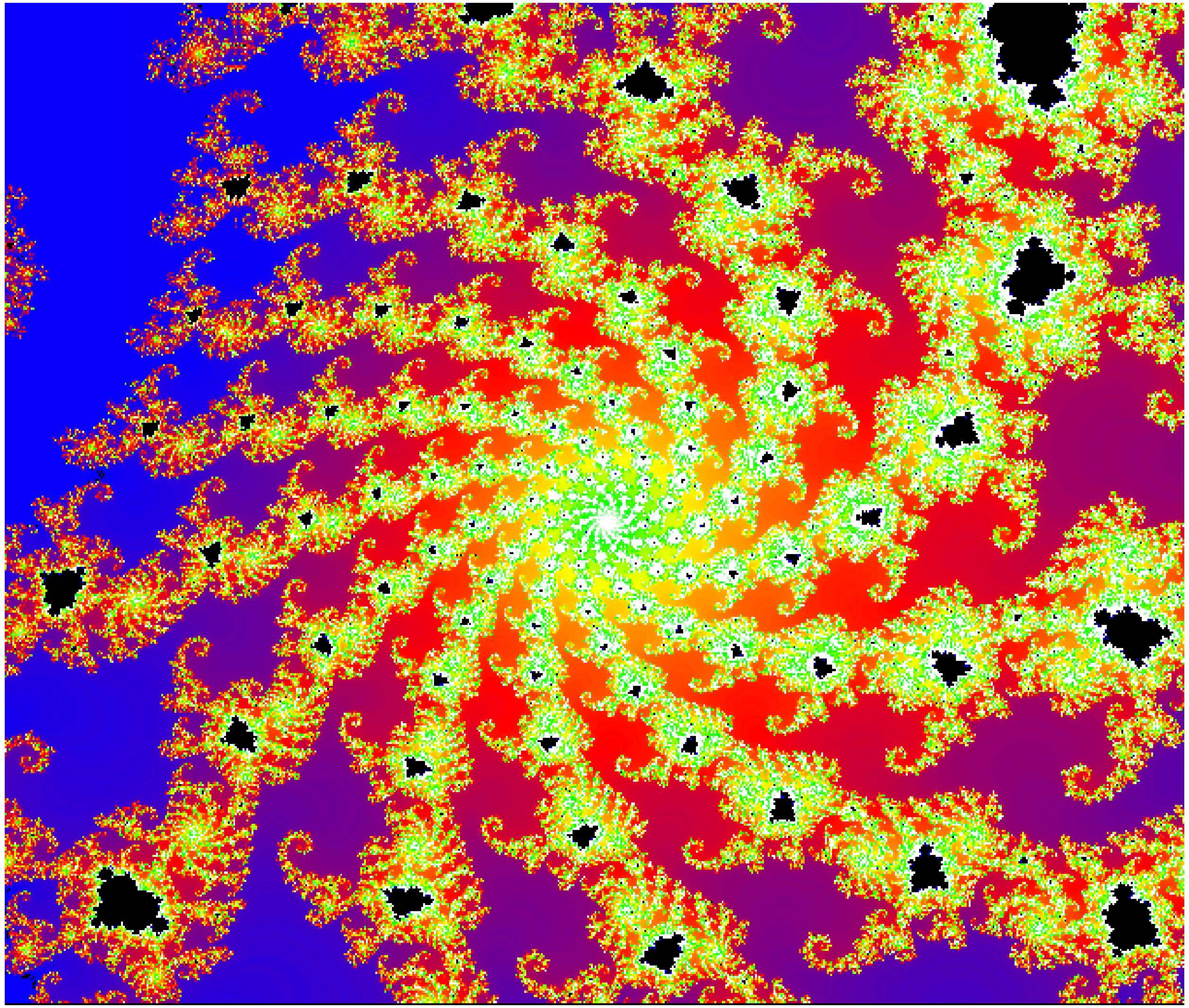
Fractals are used in many areas such as −

- Astronomy − For analyzing galaxies, rings of Saturn, etc.
- Biology/Chemistry − For depicting bacteria cultures, Chemical reactions, human anatomy, molecules, plants,
- Others − For depicting clouds, coastline and borderlines, data compression, diffusion, economy, fractal art, fractal music, landscapes, special effect, etc.



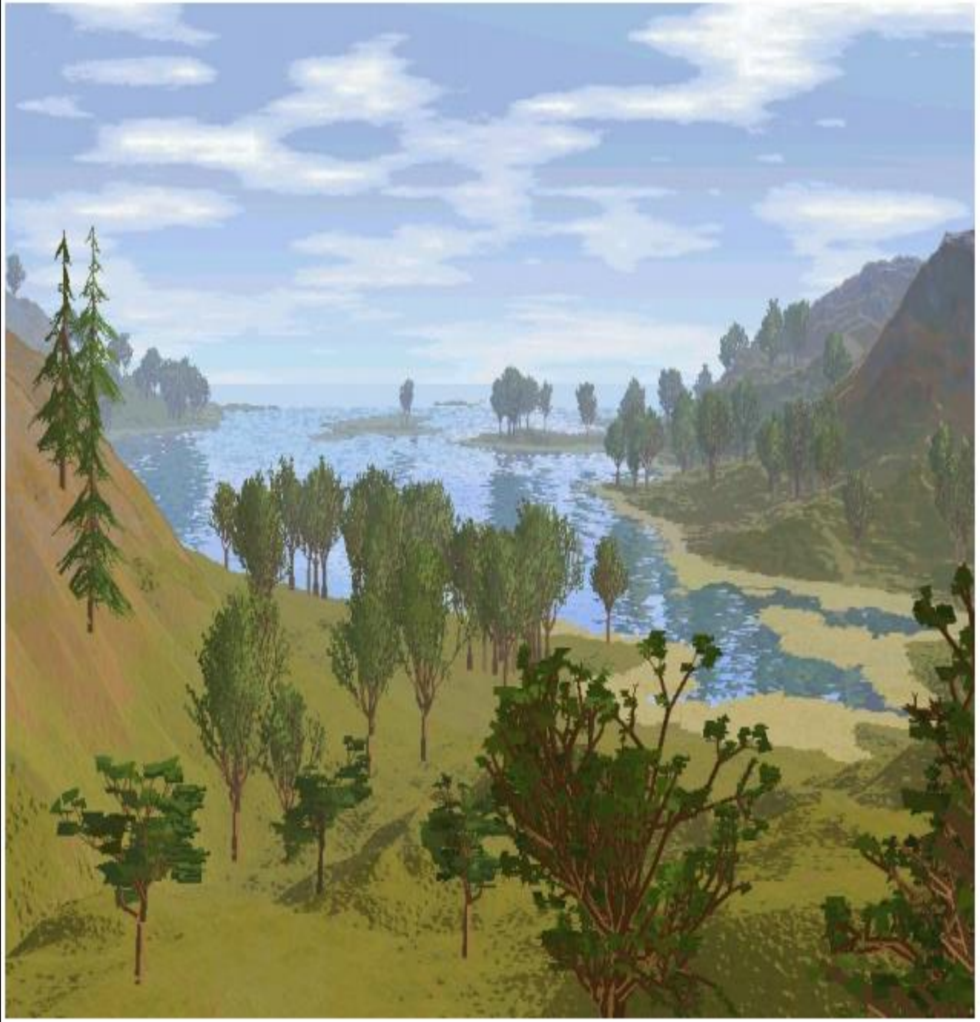


Example: Mandelbrot Set

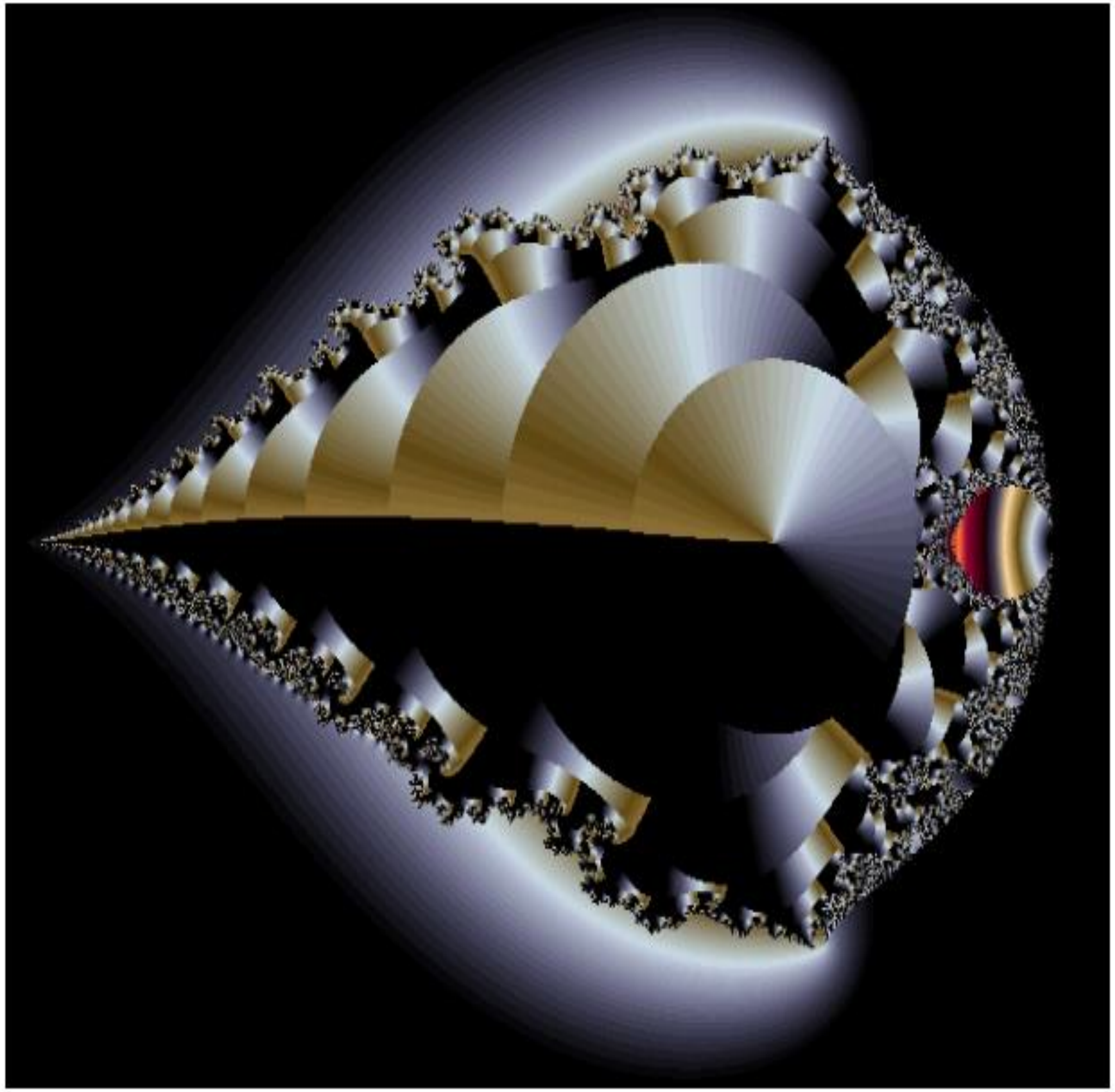


Example: Fractal Terrain

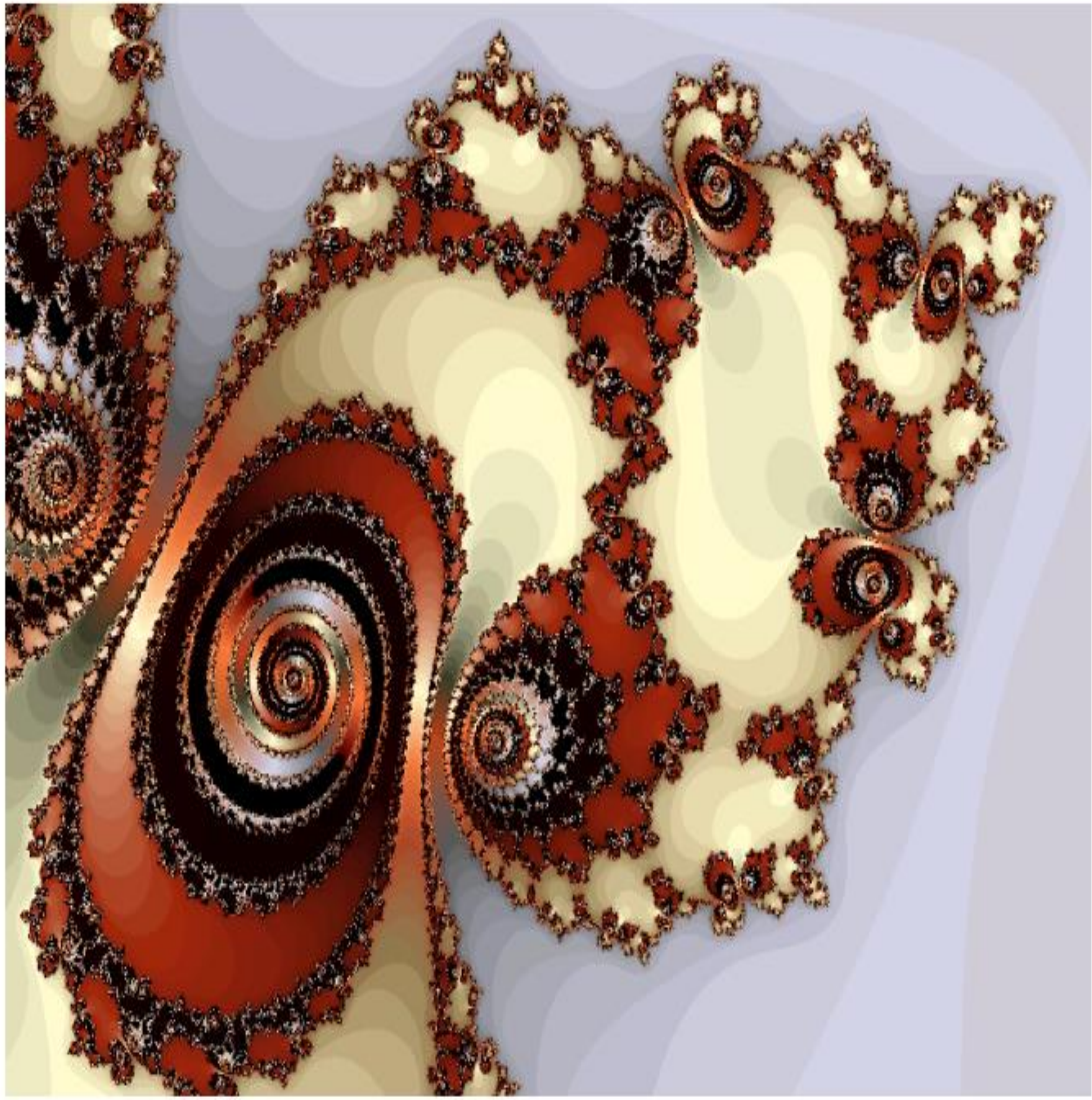
Courtesy: Mountain 3D Fractal Terrain software



Example: Fractal Art

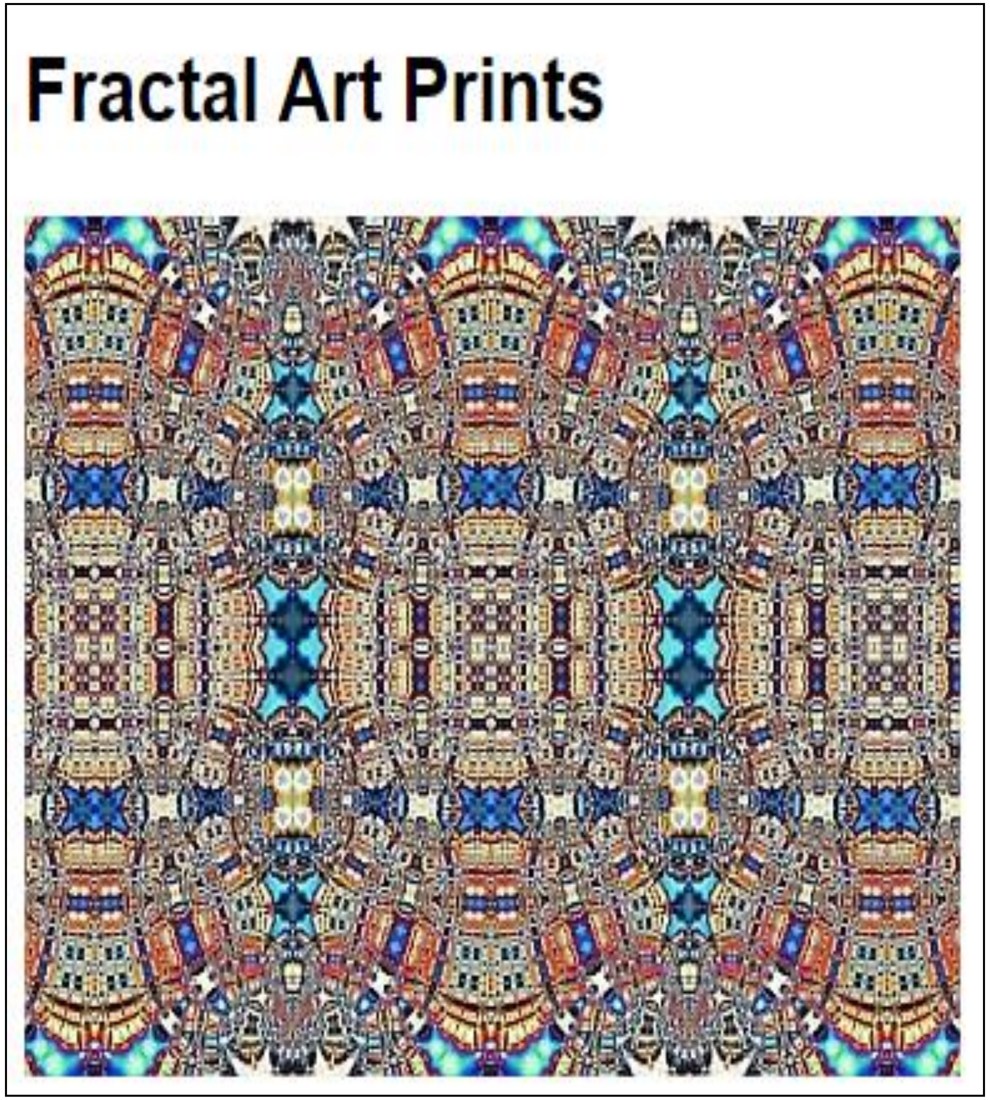


Application: Fractal Art



Uses of Fractals

- **Fractals** are **used** to model soil erosion and to analyze seismic patterns as well.
- The most useful use of **fractals** in computer science is the **fractal** image compression.
- Fractals are used to predict or analyze various biological processes or phenomena such as the growth pattern of bacteria, the pattern of situations such as nerve dendrites, etc.
- Fractals are used to describe the roughness of surfaces. A rough surface is characterized by a combination of two different fractals.
- **Medicine:** Biosensor interactions can be studied by using fractals.



Three common techniques for generating fractals

- **Iterated function systems** — These have a fixed geometric replacement rule. These are a method of generating fractals using self-similarity. An IFS image is defined as being the sum of geometric transforms of itself. It turns out that simply specifying the transforms along with a weight for each transform is enough to determine the image.

  Examples of this type are the Cantor set, Sierpinski carpet, Sierpinski gasket, Peano curve, Koch snowflake, Harter-Heighway dragon curve, T-Square, Menger sponge, are some examples of such fractals.

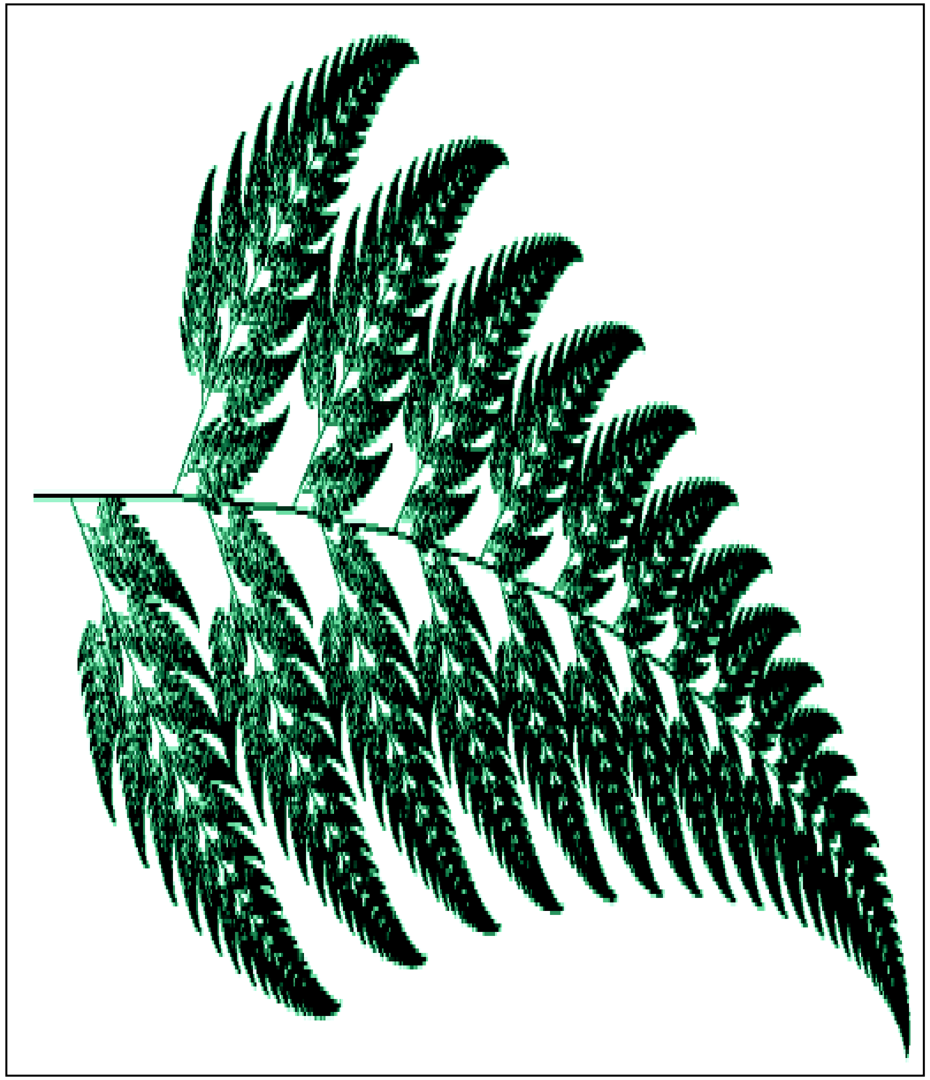
- **Escape-time fractals** — These **fractals** are generated by iterating a formula on each point in a given space. If a point diverges as the formula is iterated, it **escapes**; otherwise, it remains bounded.

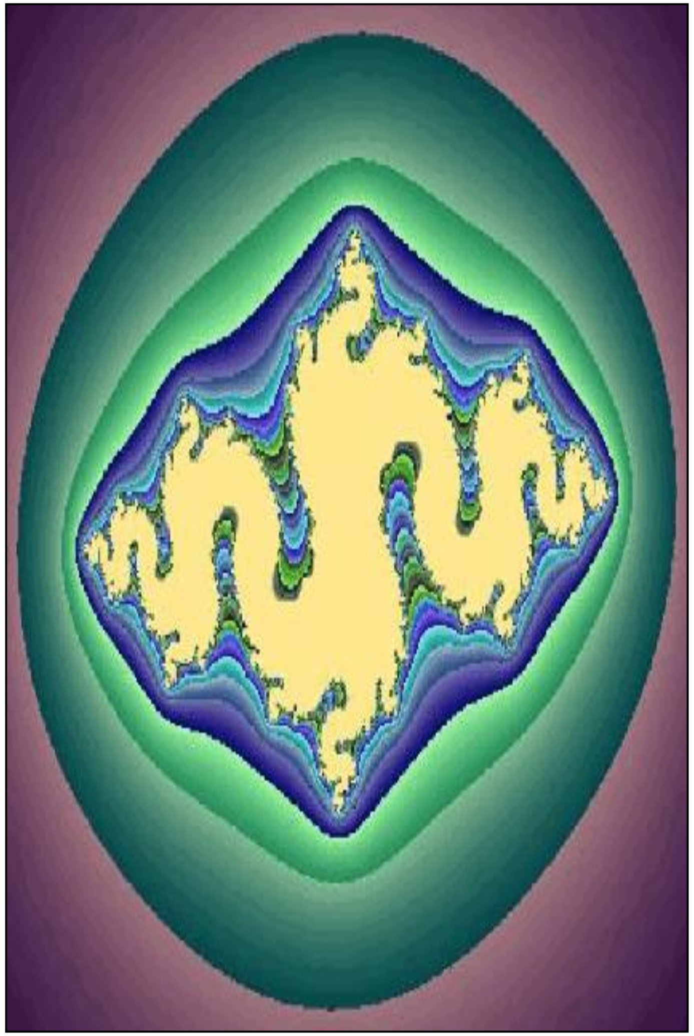
  Examples of this type are the Mandelbrot set, the Burning Ship fractal and the Lyapunov fractal.

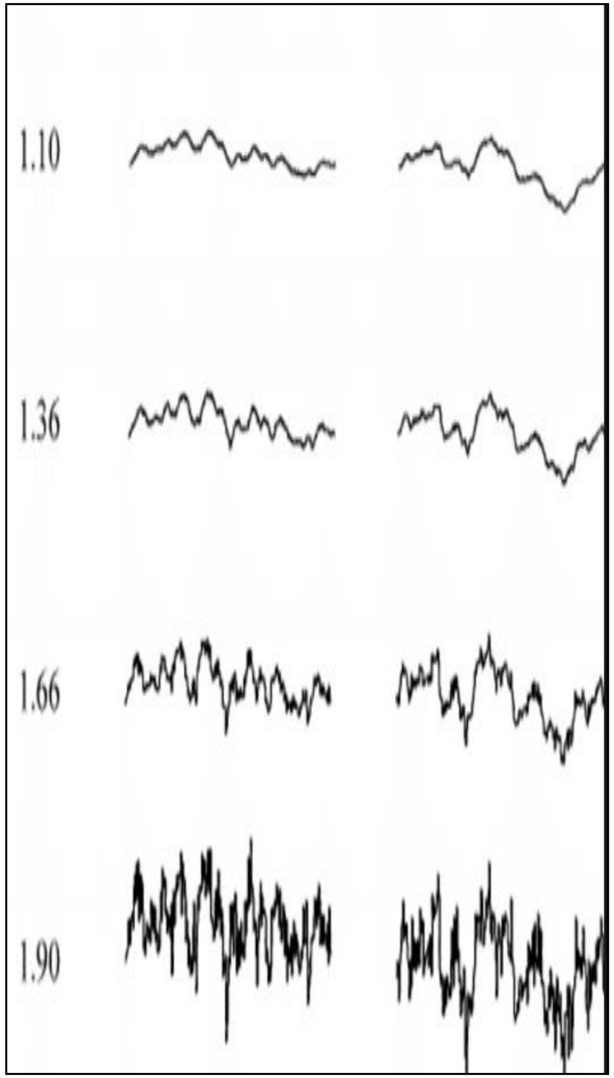
- **Random fractals**, generated by stochastic rather than deterministic processes.

  Examples of this type are the fractal landscapes, Lévy flight and the Brownian tree. The latter yields so-called mass- or dendritic fractals for example, Diffusion Limited Aggregation or Reaction Limited Aggregation clusters.

Iterated Function Systems, Escape-time fractals and Random fractals







Types of self-similarity

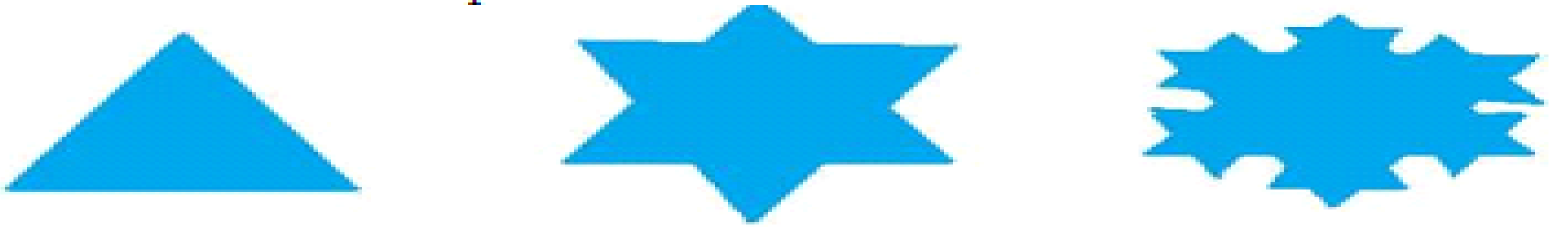
Fractals can also be classified according to their self-similarity. There are three types of self-similarity found in fractals:

- **Exact self-similarity** — This is the strongest type of self-similarity; the fractal appears identical at different scales. Fractals defined by iterated function systems often display exact self-similarity.
- **Quasi-self-similarity —** This is a loose form of self-similarity; the fractal appears approximately (but not exactly) identical at different scales. Quasi-self-similar fractals contain small copies of the entire fractal in distorted and degenerate forms. Fractals defined by recurrence relations are usually quasi-self-similar but not exactly self-similar.
- **Statistical self-similarity —** This is the weakest type of self-similarity; the fractal has numerical or statistical measures which are preserved across scales. Most reasonable definitions of "fractal" trivially imply some form of statistical self-similarity. (Fractal dimension itself is a numerical measure which is preserved across scales.) Random fractals are examples of fractals which are statistically self-similar, but neither exactly nor quasi-self-similar.

Fractals

Generation of Fractals

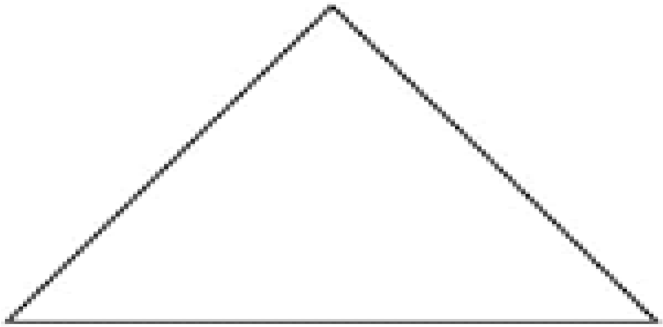
Fractals can be generated by repeating the same shape over and over again as shown in the following figure. In figure (a) shows an equilateral triangle. In figure (b), we can see that the triangle is repeated to create a star-like shape. In figure (c), we can see that the star shape in figure (b) is repeated again and again to create a new shape.
We can do unlimited number of iteration to create a desired shape. In programming terms, recursion is used to create such shapes.



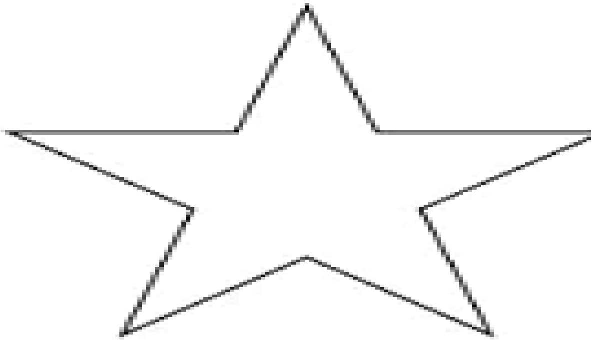
(a) Zeroth Generation (b) First Generation (c) Second Generatio

Geometric Fractals

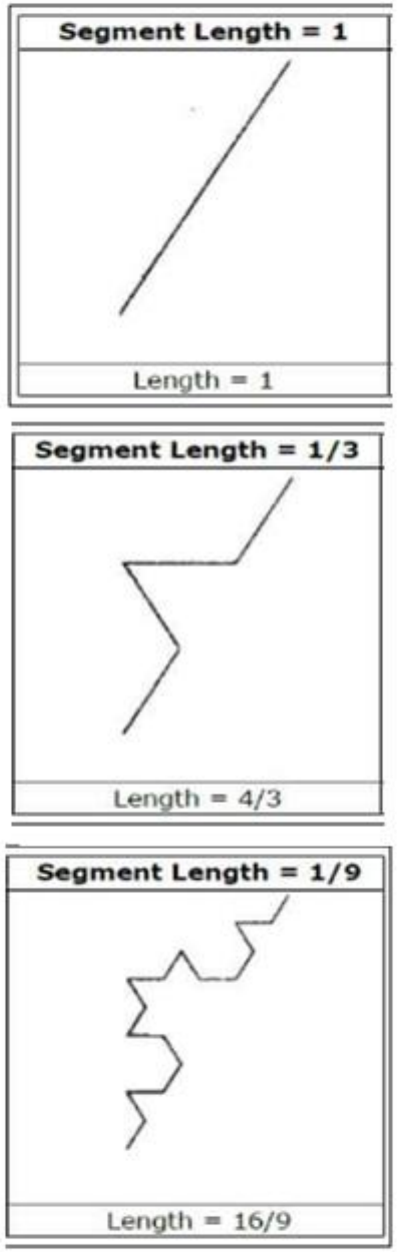
Geometric fractals deal with shapes found in nature that have non-integer or fractal dimensions. To geometrically construct a deterministic (nonrandom) self-similar fractal, we start with a given geometric shape, called the initiator. Subparts of the initiator are then replaced with a pattern, called the generator.



Initiator



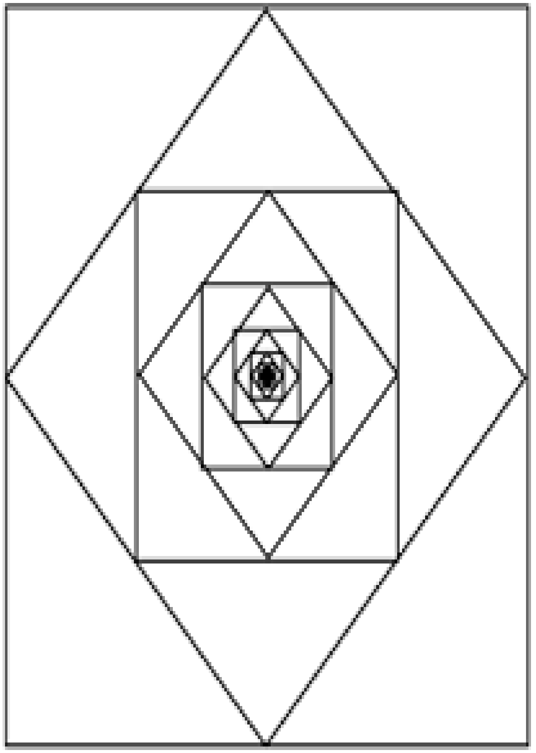
Generator



Fractals

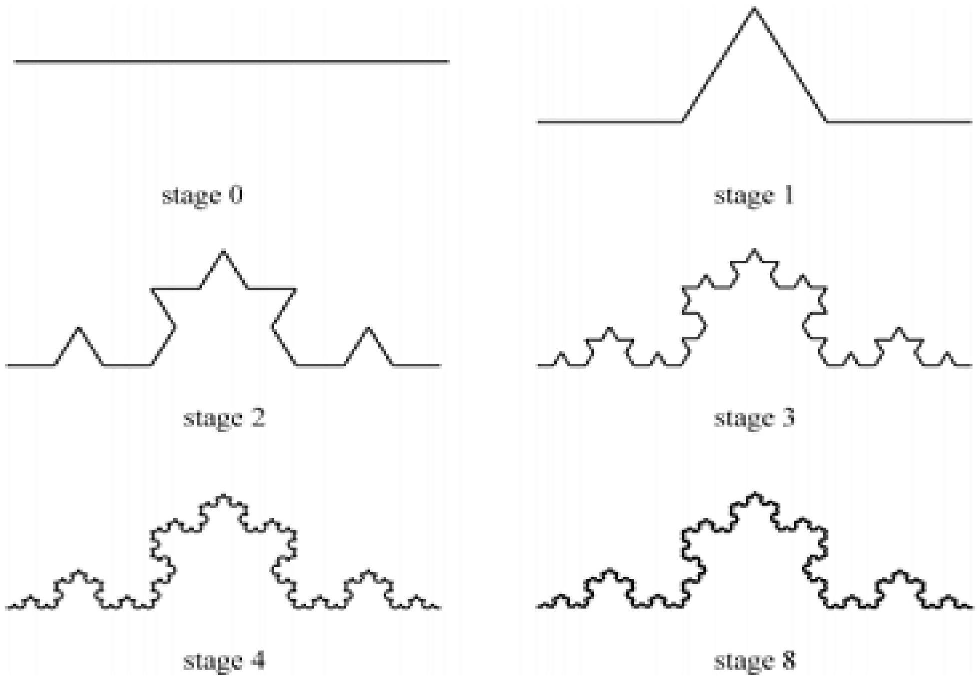
Strictly Self-Similar Fractals

A geometric figure is self-similar if there is a point where every neighborhood of the point contains a copy of the entire figure. For example, imagine the figure formed by inscribing a square within another square, rotated by 45°. Then inside the inner square, inscribe another square in the same manner, and so on ad infinitum. Of course, we can't really draw this figure, since it contains infinitely many nested squares, but an approximation of the result is shown below.



Koch Curve

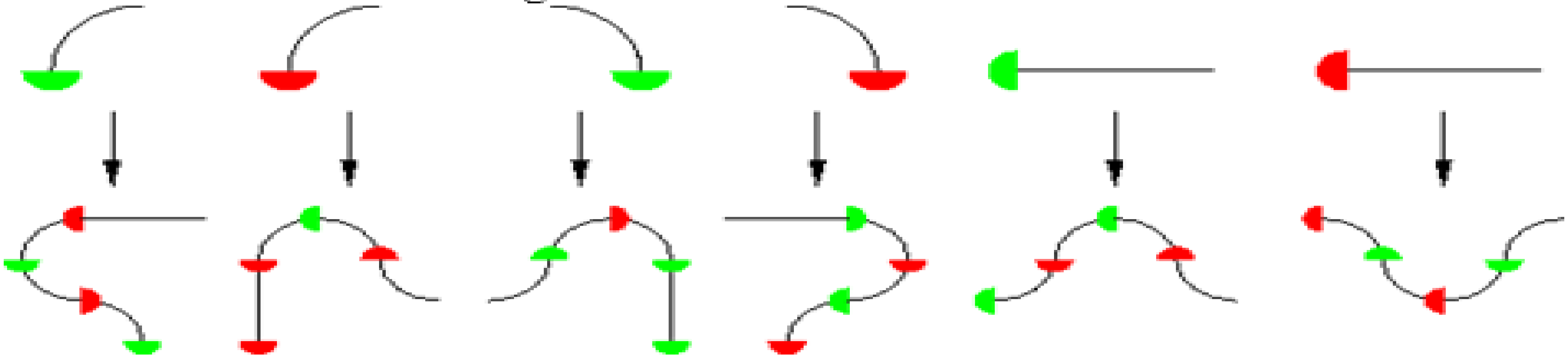
Swedish mathematician Helge von Koch introduced the Koch curve in 1904, as an example of a curve that is continuous but nowhere differentiable. It is the result of a simple geometric construction that is easily generalized to give a wide variety of examples. Begin with a line segment in the plane – for example, the interval [0, 1] along the x-axis (stage 0). On the middle third of this interval, construct an equilateral triangle upward (so each side of the triangle has length 1/3, and one side is the interval [1/3, 2/3]), and remove the base (the open interval (1/3, 2/3)). The result is four line segments of length 1/3, arranged as shown below (stage 1). For stage 2, perform the same construction on each of these four line segments so that the new segments protrude outward from those of stage 1 (see below) and continue to repeat the process indefinitely on all the smaller line segments formed at each stage. A few stages of this process are shown below.



Peano Curve and Fractal Curves

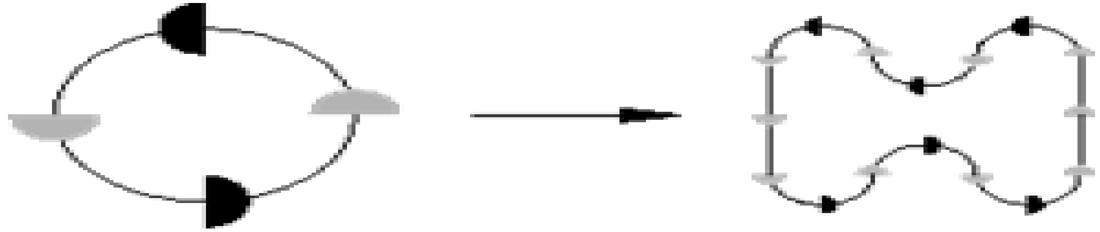
There are examples of curves (in the sense of continuous maps from the real line to the plane) that completely cover a two-dimensional region of the plane. We give a construction of such a Peano curve, adapted from David Hilbert's example. The construction is inductive, and is based on replacement rules. We consider building blocks of six shapes,

the length of the straight pieces being twice the radius of the curved ones. A sequence of these patterns end-to-end represents a curve, if we disregard the red and green half-disks. The replacement rules are the following:

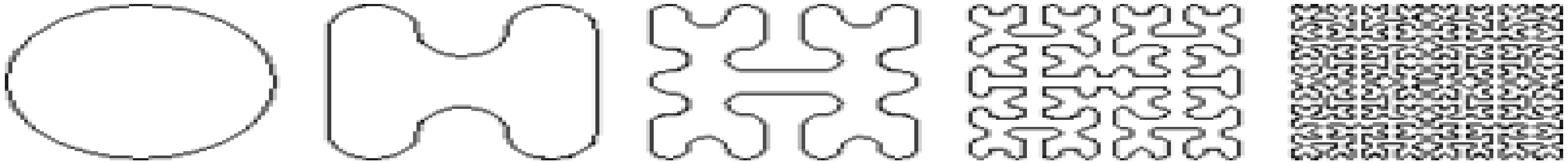


Peano Curve and Fractal Curves

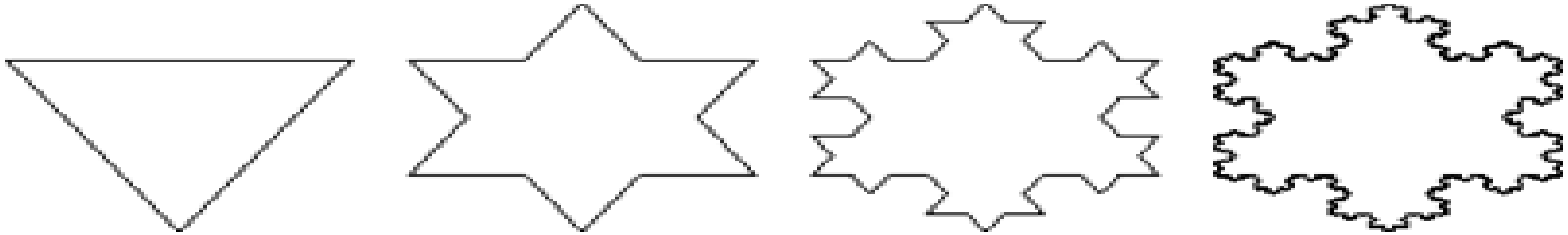
The rules are applied taking into account the way each piece is turned. Here we apply the replacement rules to a particular initial pattern:



(We scale the result so it has the same size as the original.) Applying the process repeatedly gives, in the limit, the Peano curve. Here are the first five steps:



The same idea of replacement rules leads to many interesting fractal, and often self-similar, curves. For example, the substitution  leads to the Koch snowflake when applied to an initial equilateral triangle, like this (the first three stages and the sixth are shown):



Fractals

- https://www.youtube.com/watch?v=BTiZD7p_oTc
- https://www.youtube.com/watch?v=WFtTdf3I6Ug
- https://www.youtube.com/watch?v=XwWyTts06tU
- Mathematical expressions to generate pretty pictures
- Evaluate math functions to create drawings (approach infinity -> converge to image)
- Utilizes recursion on computers
- Popularized by Benoit Mandelbrot (Yale university)
- Defined in terms of self-similarity

Examples of Fractals

- Clouds
- Grass
- Fire
- Modeling mountains (terrain)
- Coastline
- Branches of a tree
- Surface of a sponge
- Cracks in the pavement
- Designing antennae (www.fractenna.com)

FREE SOFTWARE

- Free fractal generating software
  - Fractint
  - FracZoom
  - Astro Fractals
  - Fractal Studio
  - 3DFract

Thank You