FRACTAL GEOMETRY METHODS
(Section 10-3 in *Computer Graphics*)

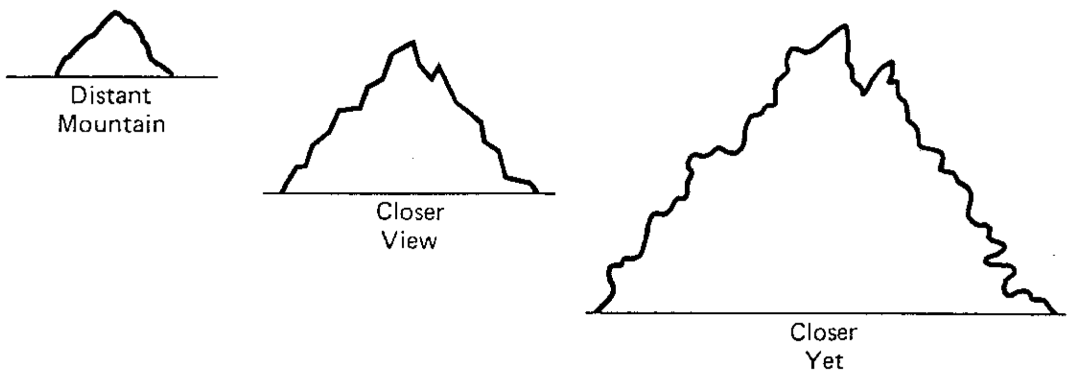
SWEEP REPRESENTATIONS
(Section 10-4 in *Computer Graphics*)

CONSTRUCTIVE GEOMETRY METHODS
(Section 10-5 in *Computer Graphics*)

QUADTREES AND OCTREES
(Section 10-6 in *Computer Graphics*)

Fractal Geometry Methods

- used to model irregular and fragmented features (of mountains and clouds, for example)
- a fractal shape has a fractal dimension in addition to its spatial dimension(s)
  - the length of a smooth curve between two points is precise
  - a fractal curve contains infinite detail at each point along the curve - the closer you get, the more detail you see

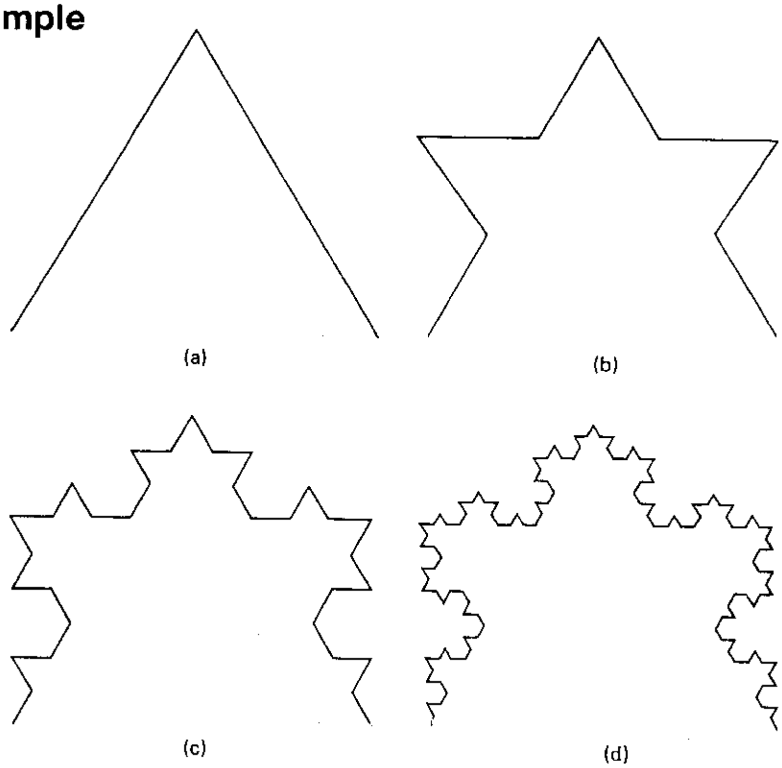


Fractal Geometry Methods, continued

- a fractal curve is generated by applying repeatedly a transformation function to points within a region of space
- detail in the final display is determined by
  - the number of iterations
  - the resolution of the display
- generating levels of detail
  - let $P_0 = (x_0, y_0)$ be the initial point
  - $P_1 = F(P_0)$
  - $P_2 = F(P_1)$
  - $P_3 = F(P_2)$
- either regular or random variations can be generated along the curve at each iteration

Fractal Geometry Methods, continued

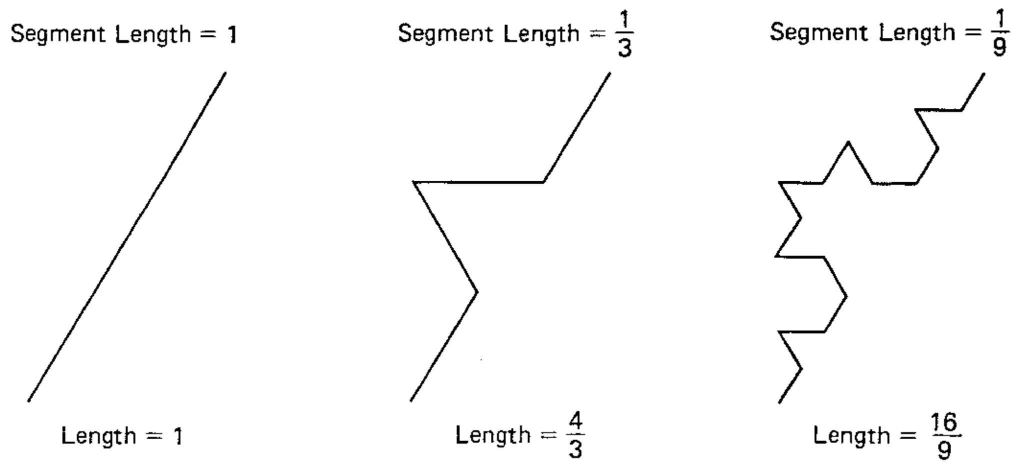
- example



  - pattern (a) is reduced by 1/3
  - the reduced pattern is used to replace the middle third
  - the result is pattern (b)

example, continued

length increases with each iteration

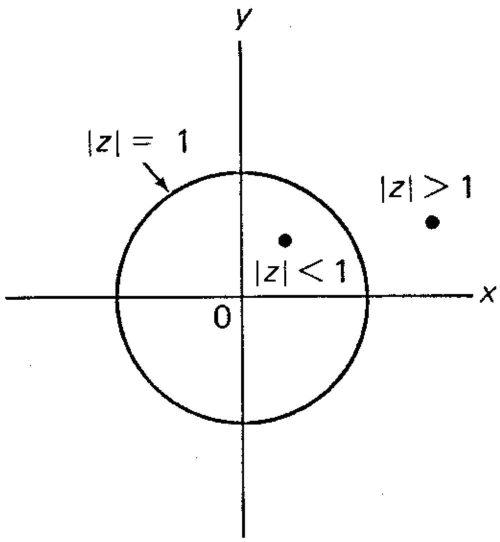


Fractal Geometry Methods, continued

- many fractal curves are generated with functions in the complex plane
  - each two-dimensional point (x, y) is represented as $z = x + iy$
  - the complex function f(z) maps points repeatedly from one position to another
  - iteration can cause points to
    - diverge to infinity
    - converge to a finite limit
    - remain on some curve

Fractal Geometry Methods, continued

- example: $f(z) = z^2$
  - transforms points according to their relation to the unit circle



- for abs(z) > 1, the sequence transforms the point toward infinity
- for abs(z) < 1, the sequence transforms the point toward the origin

example:

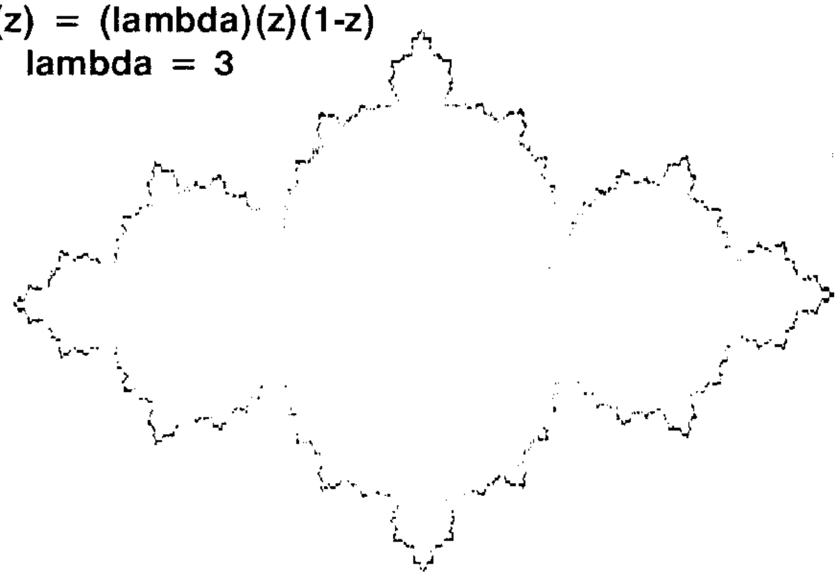
$x = 0.3, \quad y = 0.5$

$$
\begin{aligned}
f(z) &= z^2 \\
&= x^2 - y^2 + 2ixy \\
&= 0.09 - 0.25 + 2i(0.3)(0.5) \\
&= -0.16 + 0.30i
\end{aligned}
$$

- for abs(z) = 1, the sequence keeps the point on the circle

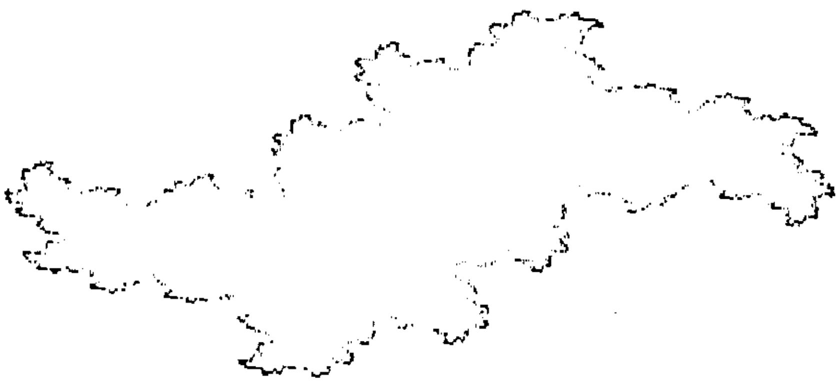
Fractal Geometry Methods, continued

- two fractal curves generated with the inverse of the function
  f(z) = (lambda)(z)(1-z)
  - lambda = 3



(a)

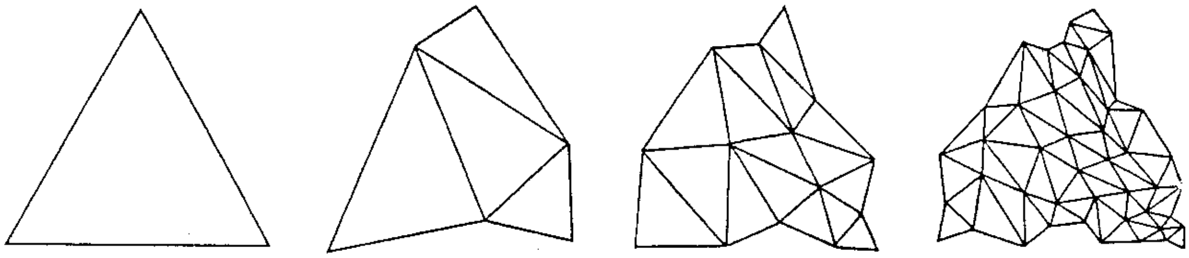
  - lambda = 2 + i



(b)

fractal surfaces

- procedures are similar to fractal curve procedures



- a triangle becomes a fractal surface by random displacement of points on the boundary
  - select a random point on each leg of the triangle
  - apply random displacement distances to each of the three points
  - join displaced points with straight lines
  - repeat the process

fractal surfaces using quaternions

- $q = q_0 + iq_1 + jq_2 + kq_3$,

- the quaternion formulation requires 4 terms; 1 term is discarded

- points in space are transformed iteratively

- each point is tested to be inside or outside the surface

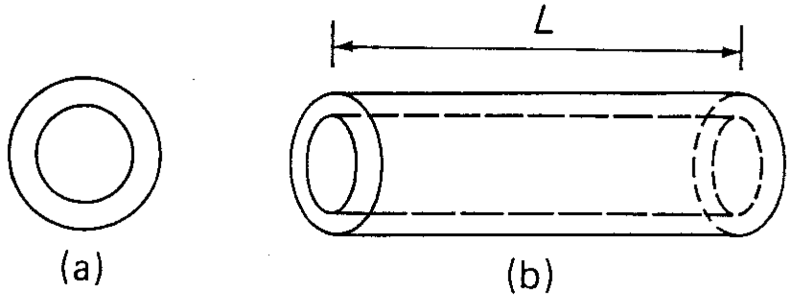
- any inside point that connects to an outside point is a surface point, keeping points near the surface

display of fractal surfaces

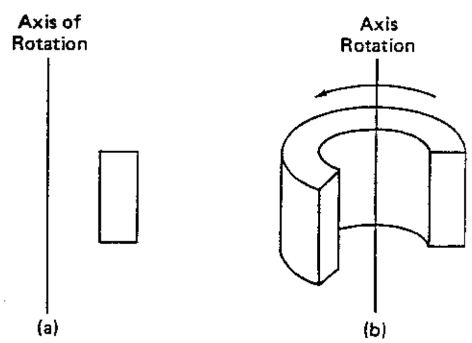
- represent each point on the surface as a small cube
- determine the lighting and color of each surface cube
- remove hidden surfaces
- see figure 10-31 on page 211
  - terrain reflects light
  - clouds absorb and scatter light
- each part of the fractal object has infinite detail
- see figure 10-32 on page 212

Sweep Representations

- solids with translational or rotational symmetry can be formed by sweeping a two-dimensional figure through a region of space
- translational sweep using a cross section of the intended object



- rotational sweep using a cross section parallel to the axis of rotation

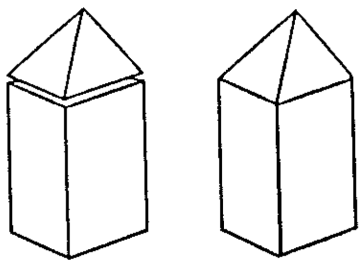


Constructive Solid-geometry Methods

- begin with a set of primitive shapes
  - blocks
  - pyramids
  - cylinders
  - cones
  - spheres
  - etc.
- apply set operations
  - union
  - intersection
  - difference

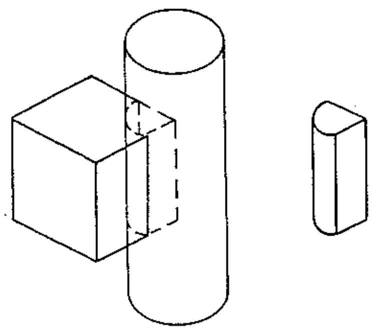
Constructive Solid-geometry Methods, continued

- union



(a) (b)

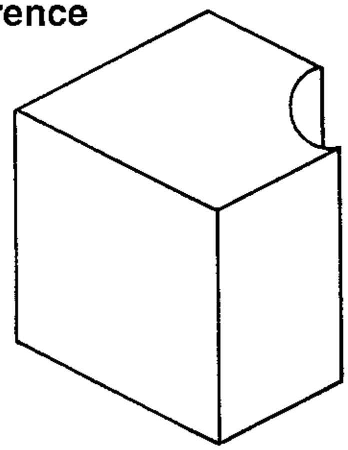
- intersection



(a) (b)

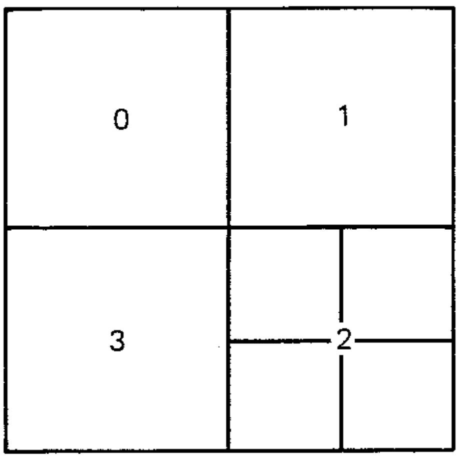
  - find points within the boundaries of both objects or
  - use octree representations

- difference

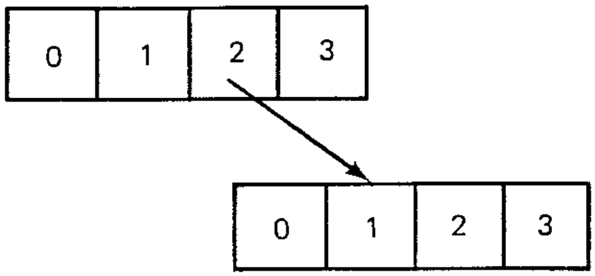


Quadtrees and Octrees

- quadtrees: tree structures which can represent planar objects
  - each node has one field for each quadrant
  - if the quadrant is homogeneous (all the pixels are the same), the field stores its description
  - if the quadrant is heterogeneous, it is divided into four subquadrants

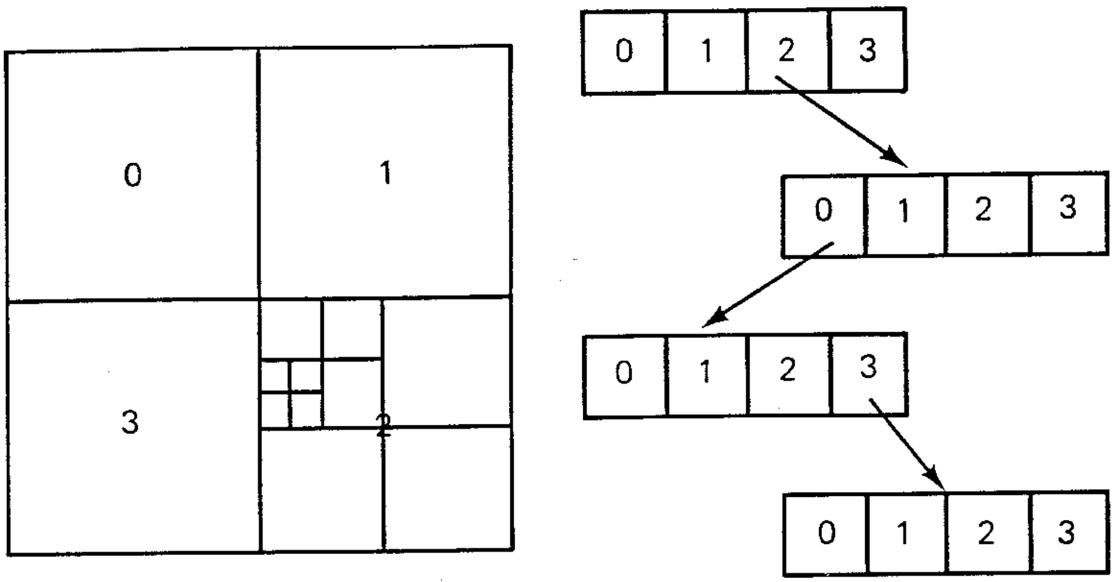


Region of a Two-Dimensional Space



Quadtree Representation

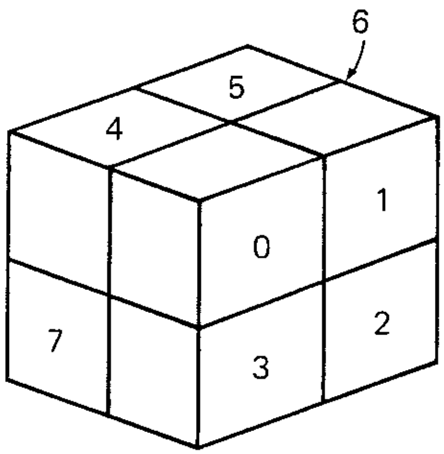
- subdivision continues until all quadrants
- are homogeneous



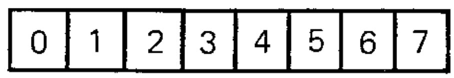
- 2^n-by-2^n pixels requires at most n levels
- quadtrees produce significant savings when large homogeneous areas exist

Quadtrees and Octrees, continued

- **octrees: tree structures which can represent solid objects**
  - **each node has one field for each octant**
  - **if all the voxels in an octant are homogeneous (including "void"), the field stores its description**
  - **if the voxels are heterogeneous, the octant is divided into eight suboctants**



Region of a
Three-Dimensional
Space



Data Elements
in the Representative
Octree Node

octree implementation

- a box is defined around the object
- octants are tested to generate the octree representation
- octrees can be unioned, intersected and differenced
- octrees support
  - transformation
  - hidden-surface removal
  - shading
  - conversion to a quadtree representation for mapping to a frame buffer
- spatial coherence produces savings

- Fractal Geometry Methods
- Sweep Representations
- Constructive Geometry Methods
- Quadtrees and Octrees