3D Object Representations

Introduction

Object Representation

- Graphics scenes can contain many different kinds of objects and material surfaces
  - Trees, flowers, clouds, rocks, water, bricks, wood paneling, rubber, paper, steel, glass, plastic and cloth
- So no single method can be used to describe all the characteristics of these different shapes/materials

3D Object Representations

- Boundary representation
  - Describing a 3D object as a set of surfaces that separate the object interior from the environment
    - Eg) Polyhedra, curved boundary surfaces

- Space-partitioning
  - Describe the interior properties by partitioning the spatial region into a set of small, non overlapping, contiguous solids (usually cubes)
    - Eg) Volumetric data, trees(data structures)

- Procedural methods
  - using Fractals, shape grammars for accurate representation of natural objects.

3D Object Representations

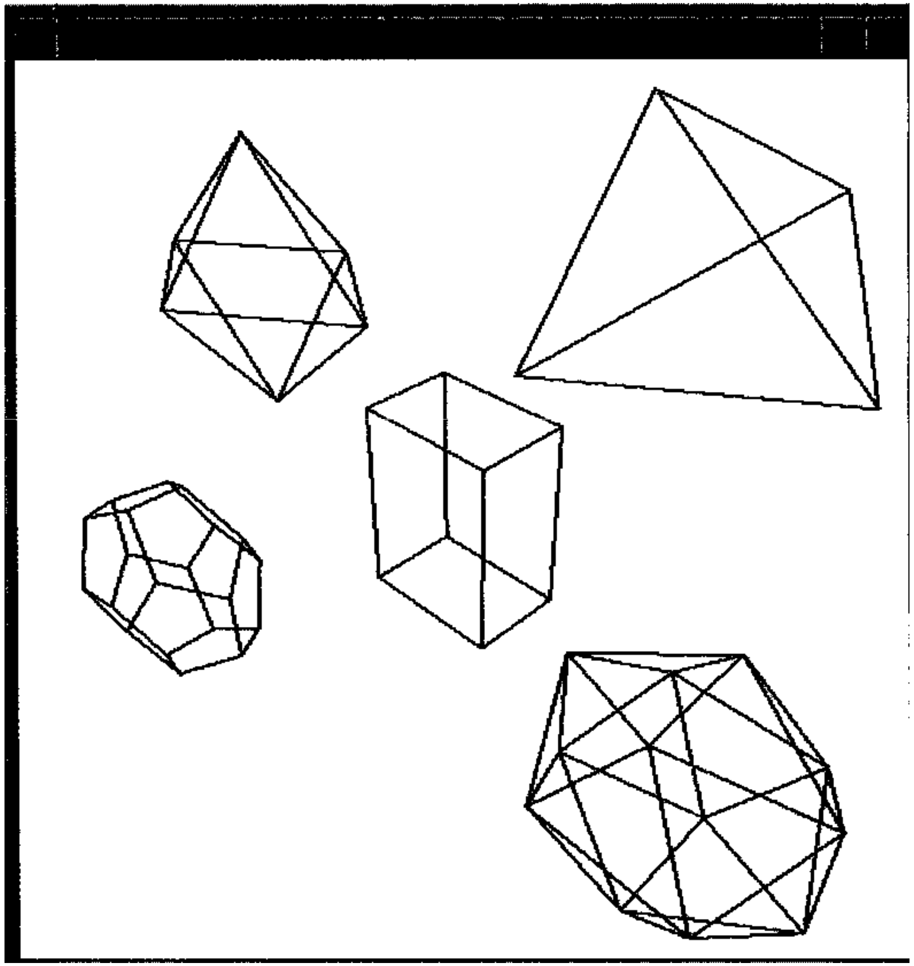
- Constructive solid geometry
  - Creates a new volume by applying set operations on two specified volumes

- Physically-based modeling:
  - Methods that simulates the behavior of objects in terms of the interaction of external and internal forces.
    - Eg : movement of rope in air, a piece of cloth

Basic Boundary Representations

- Polyhedra (a set of surface polygons)
  - triangles, quadrilaterals
- Quadric surfaces (second degree equations)
  - sphere, ellipsoid, torus
- Superquadrics (additional parameters)
  - superellipse (2D), superellipsoid (3D)
- Spline surfaces
  - Bézier, B-spline, rational splines (NURBS)

Polyhedra examples



Polygon Surfaces

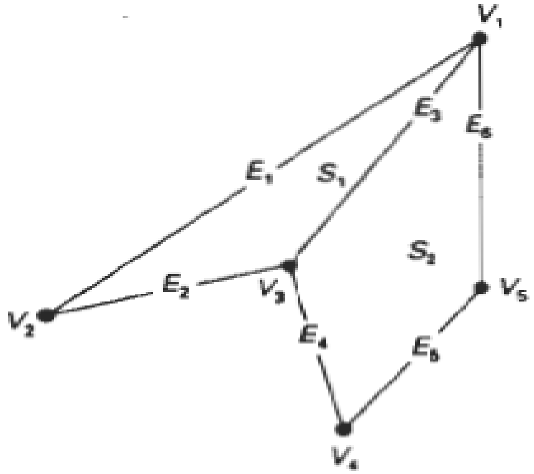
- A polyhedron is a 3D solid which consists of a collection of surface polygons, usually joined at their edges
- Simplifies and speeds up surface rendering as surfaces are described as linear equations.
- Also referred as standard graphics objects.
- Polyhedron can be represented
  - By precise surface features
  - Polygon mesh

Polygon Table

- Specify polygon surface as
  - Set of vertices & associated attribute parameters

- Polygon info is stored as data tables
  1. Geometric tables – vertex & orientation
  2. attribute tables – degree of transparency, reflectivity and texture characteristics

- Geometric data is stored as 3 lists
  - Vertex table, edge table & polygon table
  - The edge table includes pointer to the polygon table so that common edges can be identified.



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

Figure 10-2
Geometric data table representation for two adjacent polygon surfaces, formed with six edges and five vertices.

E_1:	V_1, V_2, S_1
E_2:	V_2, V_3, S_1
E_3:	V_3, V_1, S_1, S_2
E_4:	V_3, V_4, S_2
E_5:	V_4, V_5, S_2
E_6:	V_5, V_1, S_2

Figure 10-3
Edge table for the surfaces of Fig. 10-2 expanded to include pointers to the polygon table.

Plane Equations

- To produce a display of 3D object we process the input data representation for the object through several procedures.(WC to DC)

- For some of the processes, information about the spatial orientation of the individual surface components of the objects are needed.

- The information is obtained from the vertex coordinate values and the equations that describe the polygon surfaces.

Plane Equations

- Equation of a plane surface
  - $Ax + By + Cz + D = 0$

- (x,y,z) is any point on the plane A,B,C,D are constants describing spatial properties of the plane.
- To find A,B,C,D solve sets of plane eqns.
- $(x1,y1,z1)$ $(x2,y2,z2)$, $(x3,y3,z3)$
- Solve set of simultaneous linear plane equations for the ratios

$$(A/D)xk + (B/D)\,yk + (C/D)zk = -1 \qquad\qquad k=1,2,3$$

The solution for this set of equations can be obtained in determinant form, using Cramer's rule, as

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \qquad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}$$

$$C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \qquad D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \tag{10-3}$$

Expanding the determinants, we can write the calculations for the plane coefficients in the form

$$\begin{aligned} A &= y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \\ B &= z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \\ C &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \\ D &= -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1) \end{aligned} \tag{10-4}$$

Orientation of a plane surface

- The orientation of the plane surface can be described with the normal vector, which has Cartesian components(A,B,C) that are calculated with above equations.

- Need to distinguish between two sides of the polygon surface (inside and outside)
  - Inside: Plane faces the object interior
  - Outside: Outward face.

- Normal vector will be from inside to outside if
  - polygon vertices are specified in counterclockwise direction &
  - Viewing from the outer side of the plane in a right handed coordinate system.

Normal Vector N Calculations using unit cube

- Determine the components of normal vector by two methods
- Method 1:
  - Select 3 vertices in counterclockwise direction
  - Coordinates are substituted to obtain plane coefficients: A=1,B=0,C=0,D=-1 by substituting these vertices in determinant eqns.
  - Normal vector obtained is in the positive x axis
- Method 2:
  - Normal vector can be obtained using vector cross product
  - V1,V2,V3 are vertex positions from outside to inside
    - N = (V2-V1) × (V3-V1)
    - Generate value for A,B,C and obtain D using plane equations

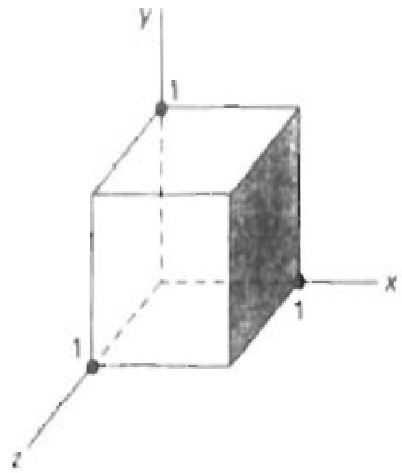


Figure 10-5
The shaded polygon surface of the unit cube has plane equation $x - 1 = 0$ and normal vector $\mathbf{N} = (1, 0, 0)$.

Inequalities of Plane Eqns.

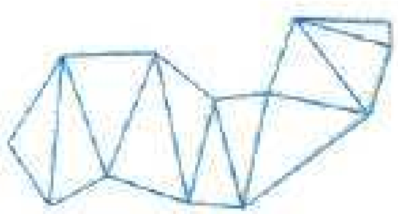
- Plane eqns. are also used to find the position of the spatial points relative to the plane surfaces.

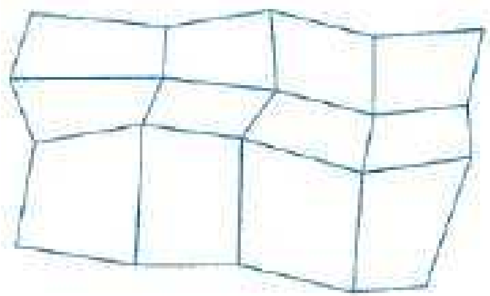
- For any point (x,y,z) not on plane surface,
  - $Ax + By + Cz + D \neq 0$

  - $Ax + By + Cz + D < 0$ point lies inside the surface
  - $Ax + By + Cz + D > 0$ point lies outside the surface

Polygon Mesh

- When object surfaces are to be tiled, it is more convenient to specify surface faces with a polygon mesh.
- Common types of polygon meshes are triangular strip and quadrilateral meshes.
- Triangle strip
    - Produce n-2 connected triangles, given n-vertices
- Quadrilateral mesh
    - Generate (n-1)(m-1) quadrilaterals, given n by m array of vertices





Curved Lines and Surfaces

- Display of 3D curved lines and surfaces are generated using set of mathematical functions defining the objects or from a set of user specified data points.
- When functions are specified, the package can project the defining equations of a curve to the display plane and plot pixel along the path of Projection plane.
  - Eg: Quadrics and super quadrics
- When of set discrete coordinate points is used to specify an object shape, a functional description is obtained that best fits the designated points according to the constraints of the application.
  - Eg: spline representations

Quadric - Sphere

- A frequently used class of objects are quadric surfaces
- These are 3D surfaces described using quadratic equations
- Quadric surfaces include:
    - Spheres
    - Ellipsoids
    - Tori

Quadric - Sphere

- A spherical surface with radius r centred on the origin is defined as the set of points (x, y, z) that satisfy the equation

$$x^2 + y^2 + z^2 = r^2$$

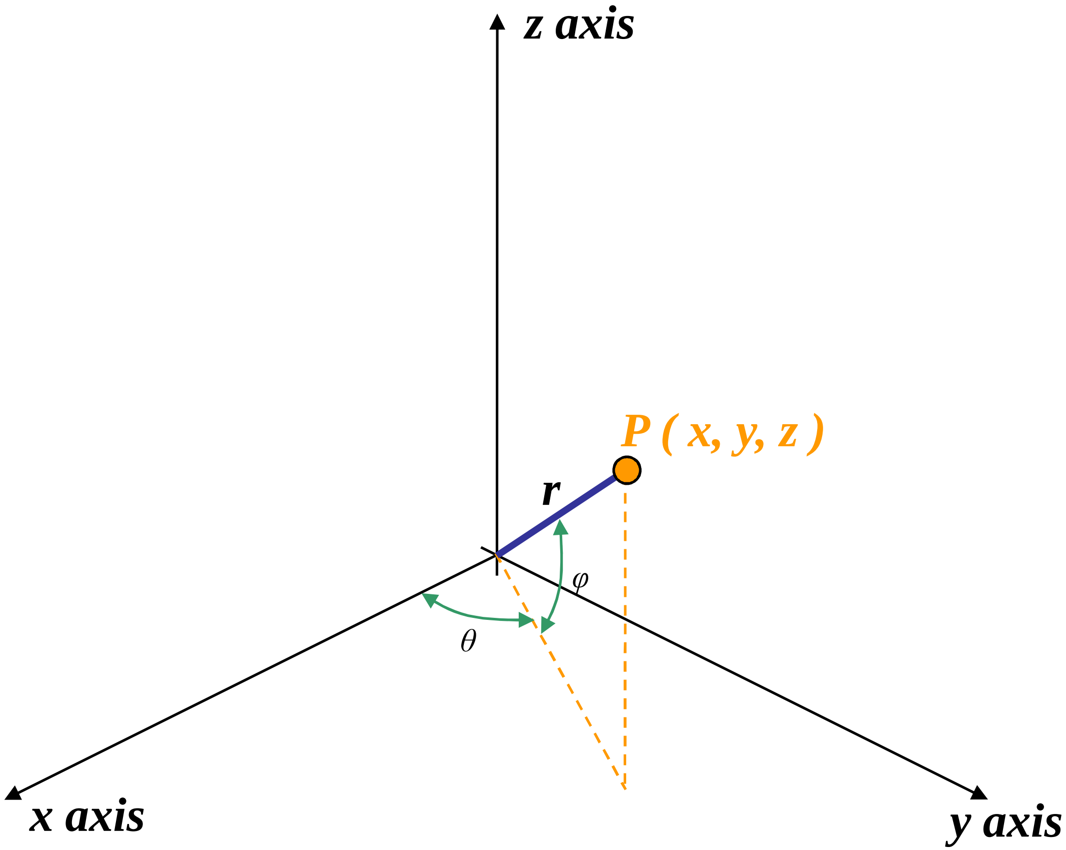
- This can also be done in parametric form using latitude and longitude angles

$$x = r\cos\varphi\cos\theta \qquad -\pi/2 \le \varphi \le \pi/2$$

$$y = r\cos\varphi\sin\theta$$

$$z = r\sin\varphi \qquad -\pi \le \theta \le \pi$$

Quadric - Sphere



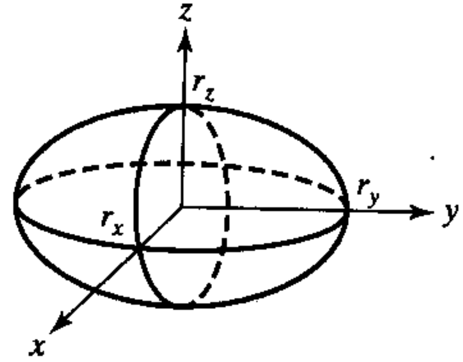
Quadric - ellipsoid

- An Ellipsoid is an extension of spherical surface where the radii in three mutually perpendicular directions can have different values.
- The Cartesian representation for points over the surface of an ellipsoid is centered on the origin is

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 = 1$$

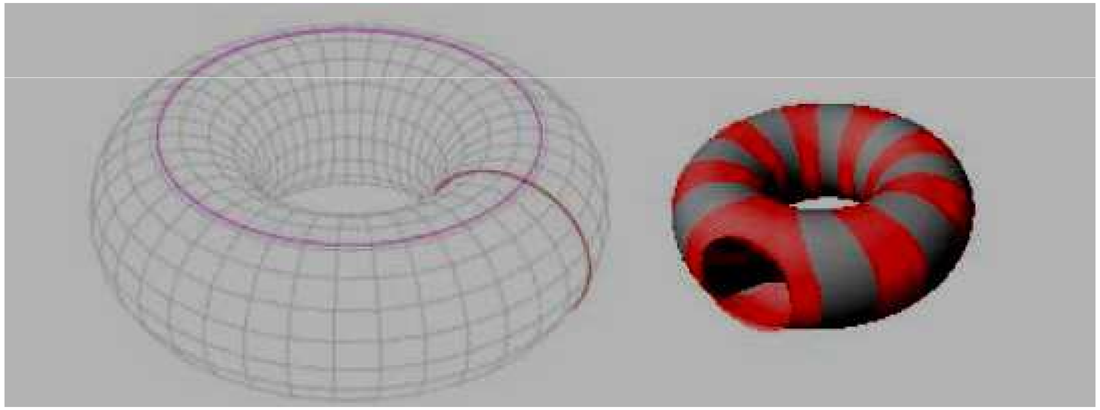
- The parametric representation for the ellipsoid in terms of latitude and longitude angle is

$$\begin{aligned} x &= r_x \cos\phi \cos\theta, & -\pi/2 \le \phi \le \pi/2 \\ y &= r_y \cos\phi \sin\theta, & -\pi \le \theta \le \pi \\ z &= r_z \sin\phi \end{aligned}$$

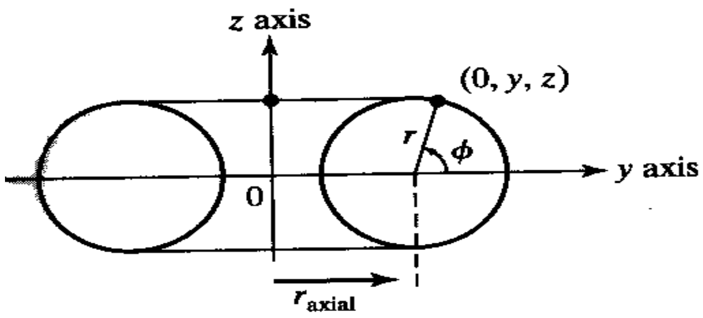


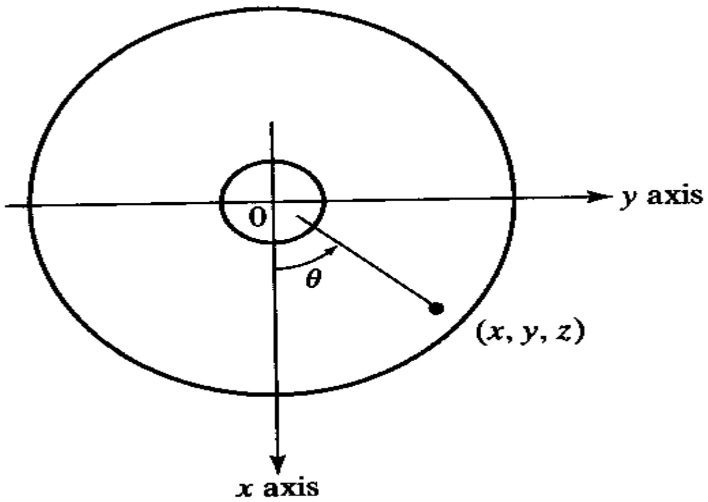
Quadric - Torus

- The torus is doughnut shaped object .
- Obtained by rotating circle or ellipse about a specified axis.



Quadric - torus





$$\left[r - \sqrt{\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2} \right]^2 + \left(\frac{z}{r_z}\right)^2 = 1 \qquad (10\text{-}11)$$

where r is any given offset value. Parametric representations for a torus are similar to those for an ellipse, except that angle ϕ extends over 360°. Using latitude and longitude angles ϕ and θ, we can describe the torus surface as the set of points that satisfy

$$\begin{aligned} x &= r_x(r + \cos\phi)\cos\theta, && -\pi \le \phi \le \pi \\ y &= r_y(r + \cos\phi)\sin\theta, && -\pi \le \theta \le \pi \qquad (10\text{-}12) \\ z &= r_z \sin\phi \end{aligned}$$

SuperQuadrics

- This class of objects are a generalization of quadric representations.
- Super quadrics are obtained by incorporating additional parameters to the quadric equations to provide increased flexibility for adjusting object shapes.

Superquadric - superellipse

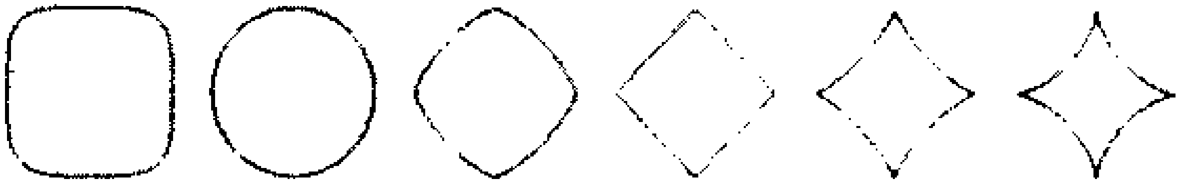
- The super ellipse is obtained from the equation of ellipse by allowing the exponent on the x and y terms to be variable
- The Cartesian super ellipse equation is

$$\left(\frac{x}{r_x}\right)^{2/s} + \left(\frac{y}{r_y}\right)^{2/s} = 1$$

- Parametric Equations

$$x = r_x \cos^s \theta, \qquad -\pi \le \theta \le \pi$$

$$y = r_y \sin^s \theta$$



rx=ry and different s

0.5 1.0 1.5 2.0 2.5 3.0

Superquadric - Superellipsoid

- It is obtained from the equation of ellipsoid by incorporating two exponent parameters

$$\left[\left(\frac{x}{r_x}\right)^{2/s_2} + \left(\frac{y}{r_y}\right)^{2/s_2}\right]^{s_2/s_1} + \left(\frac{z}{r_z}\right)^{2/s_1} = 1$$

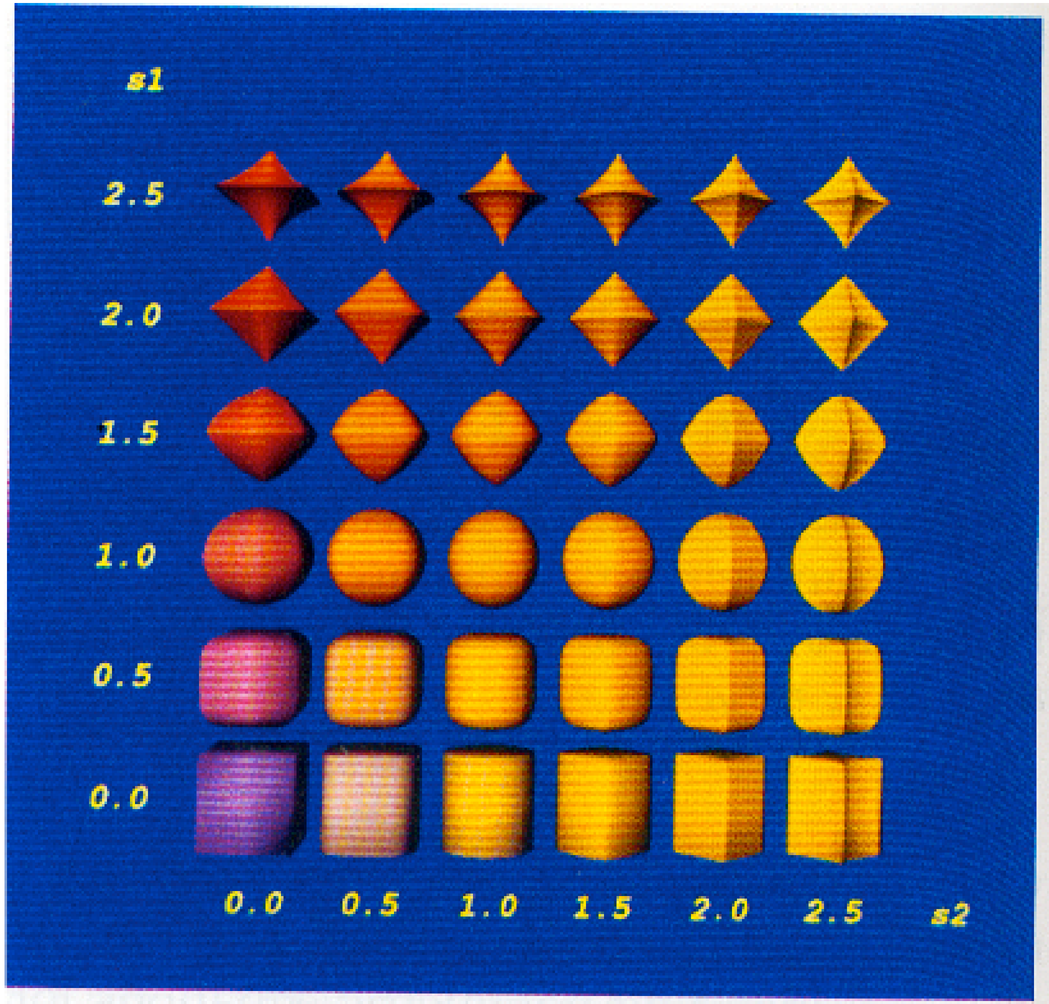
$$x = r_x \cos^{s_1}\phi \cos^{s_2}\theta, \qquad -\pi/2 \leq \phi \leq \pi/2$$

$$y = r_y \cos^{s_1}\phi \sin^{s_2}\theta, \qquad -\pi \leq \theta \leq \pi$$

$$z = r_z \sin^{s_1}\phi$$

Superellipsoids

Super ellipsoids plotted with different values for s1 and s2 with rx=ry=rz.



Blobby objects

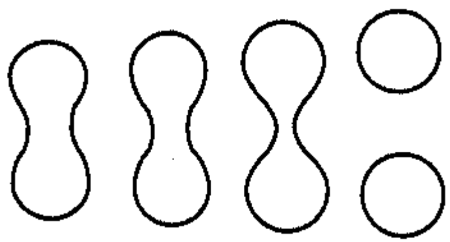
Some objects do not maintain fixed shape, but change their surface properties

- in certain motions
- in contact with other objects
- Shape is not fixed
  - water droplets
  - melting objects
  - Molecular shapes
  - muscle shape in human body
- These objects exhibit their "blobbiness".
- Also, various bumps and dents are often used to describe

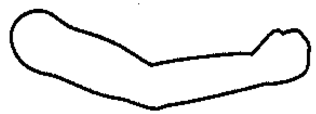
the object

Usual principle

- Fixed volume while shape is changed, e.g. molecules moving apart from each other and human muscles
- Molecular bonding : As two molcules move away from each other, the surface shapes stretch,snap and finally contract into spheres.



(a)



(b)

Gaussian functions

- To represent blobby objects several density functions are used. One such function is Gaussian density functions or bumps.

$$f(x,y,z) = \sum_k b_k \cdot e^{-a_k \cdot r_k^2} - T = 0$$

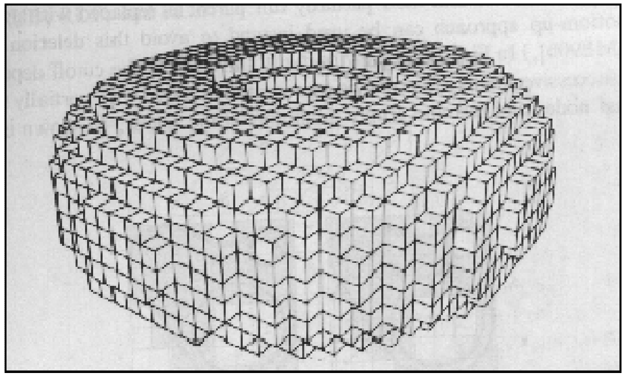
$$\text{where } r_k = \sqrt{x_k^2 + y_k^2 + z_k^2}, \text{T is a treshold value}$$

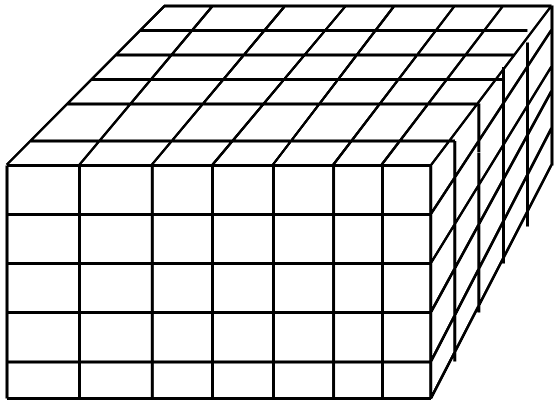
• Parameters a and b are used to adjust the amount of blobbiness.

• If b < 0 – dents instead of bumps

Spatial Partitioning

- Volume data
  - Use identical cells (voxels)
  - Expensive storage but simple data structure
  - Useful for medical imaging: volume visualization





Spatial Partitioning

- Octrees
  - Partition space into 8 cubes, recursively

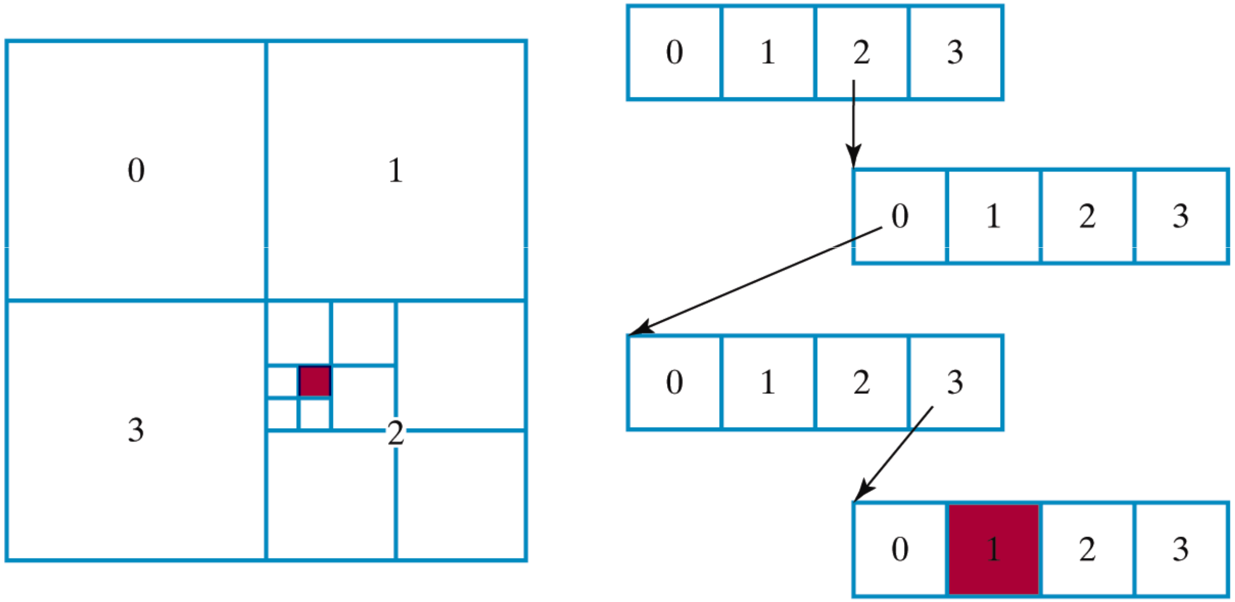
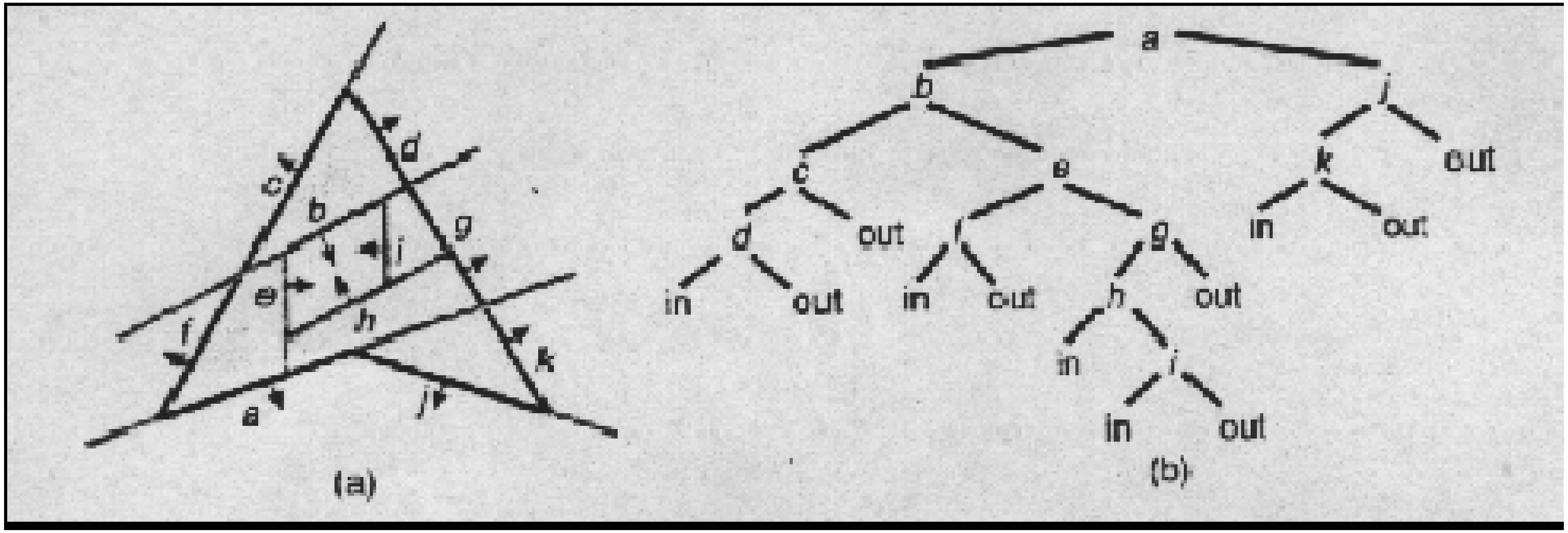


Figure 8-65

Quadtree representation for a square region of the *xy* plane that contains a single foreground-color area on a solid-color background.

Spatial Partitioning

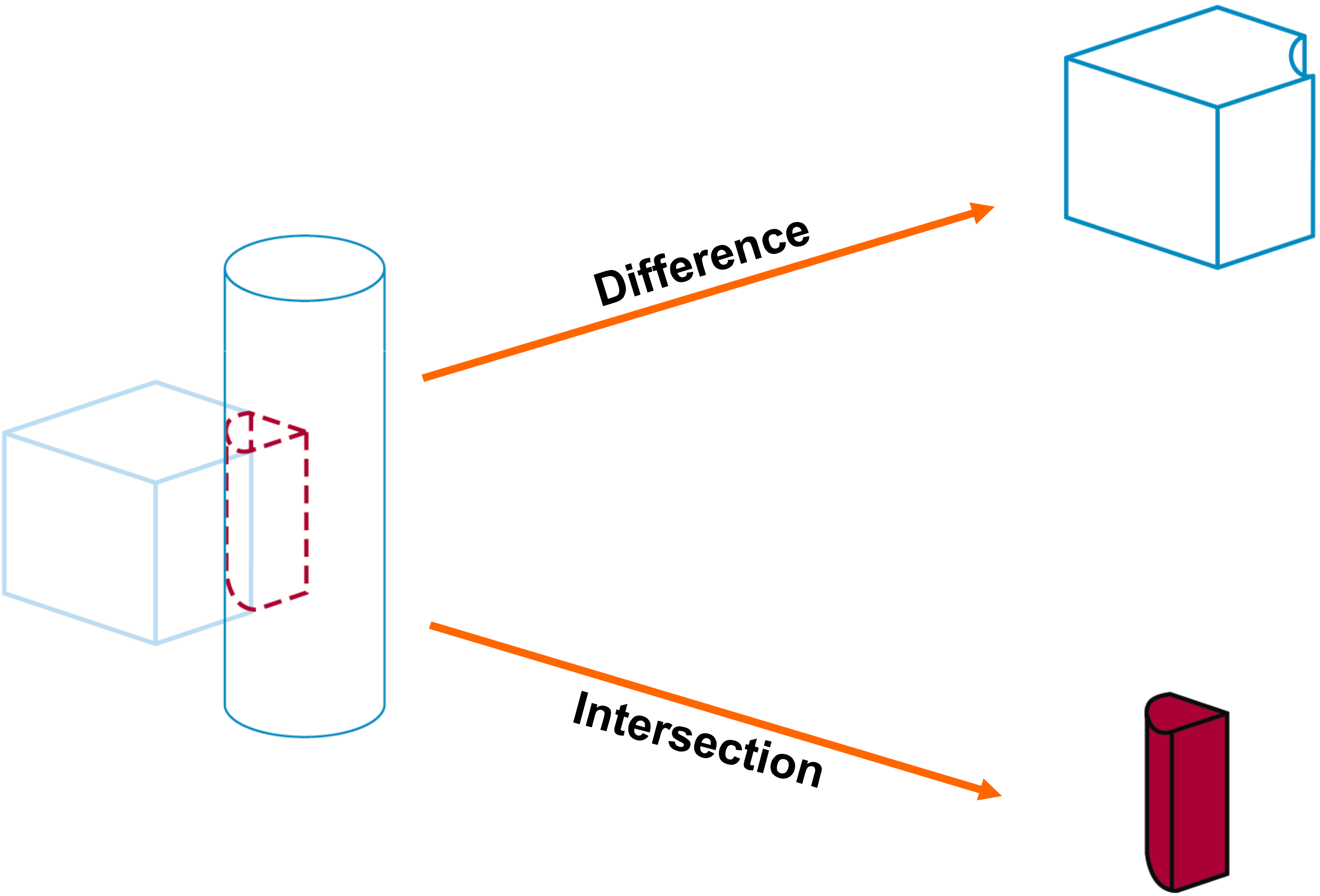
- Binary Space Partitioning (BSP) trees
  - Subdivide a scene into two sections at each step with a plane that can be at any position and orientation



Constructive Solid Geometry Methods

- Constructive Solid Geometry (CSG) performs solid modelling by generating a new object from two three dimensional objects using a set operation
- Valid set operations include
  - Union
  - Intersection
  - Difference

Constructive Solid Geometry Methods



Constructive Solid Geometry Methods

- CSG usually starts with a small set of primitives such as blocks, pyramids, spheres and cones

- Two objects re initially created and combined using some set operation to create a new object

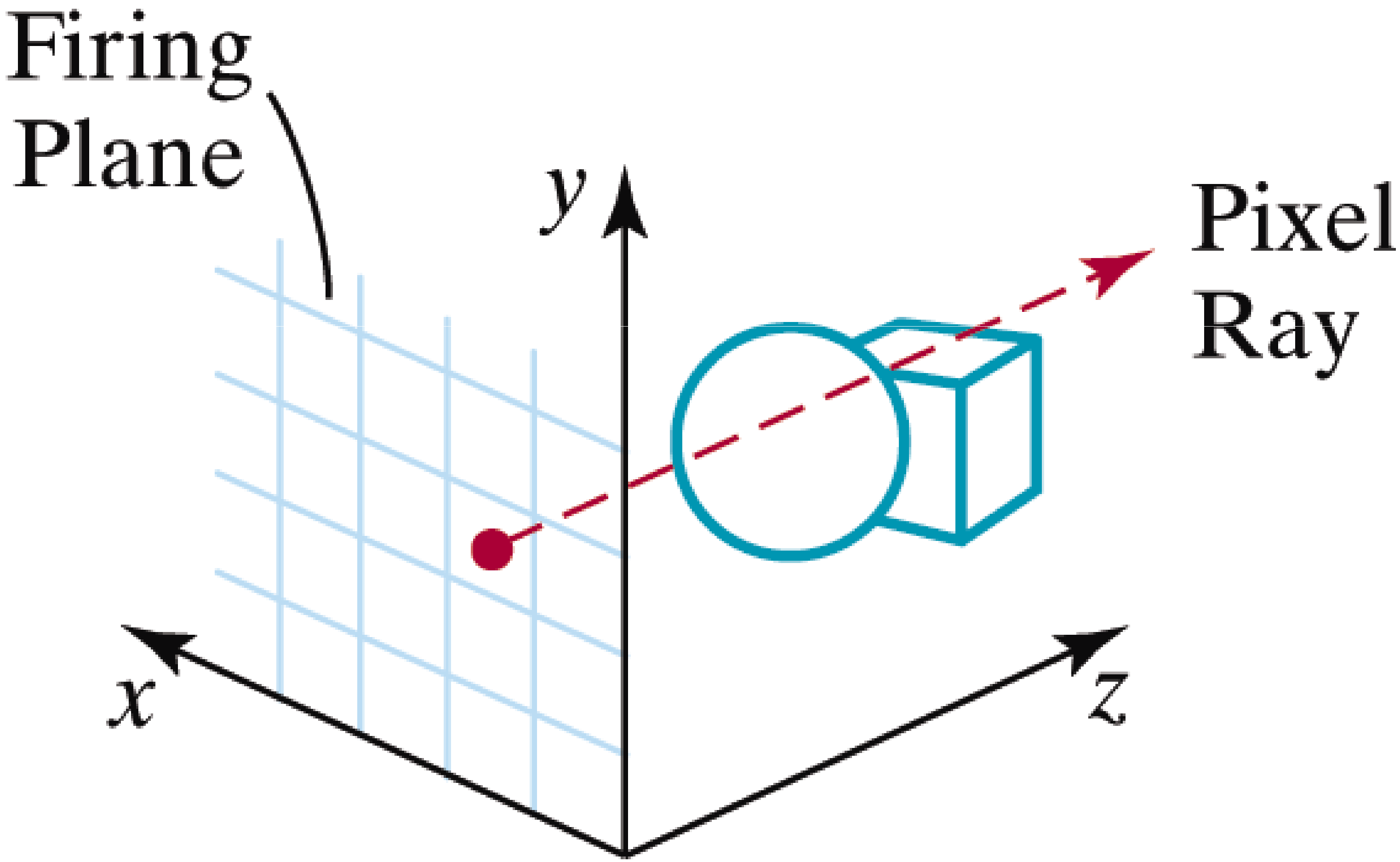
- This object can then be combined with another primitive to make another new object

- This process continues until modelling complete

Ray-Casting

- **Ray-casting** is typically used to implement CSG operators when objects are described with boundary representations.
- Ray casting is applied by determining the objects that are intersected by a set of parallel lines emanating from the xy plane along the z axis.
- The xy plane is referred to as the **firing plane**

Ray-Casting

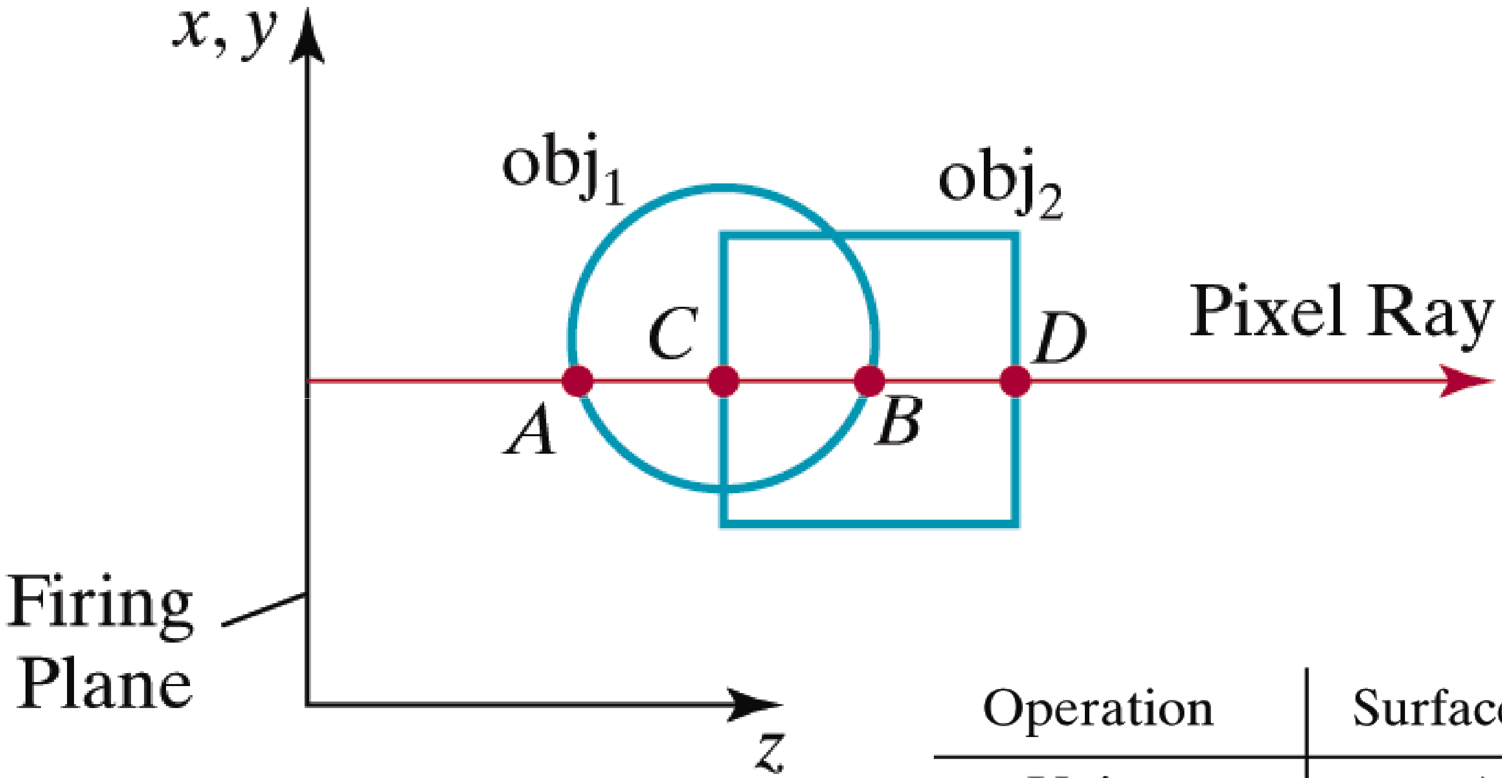


Ray-Casting

- Surface intersections along each ray are calculated and these are sorted according to distance from the firing plane.

- The surface limits for the composite object are then determined by the specified set operation

Ray-Casting



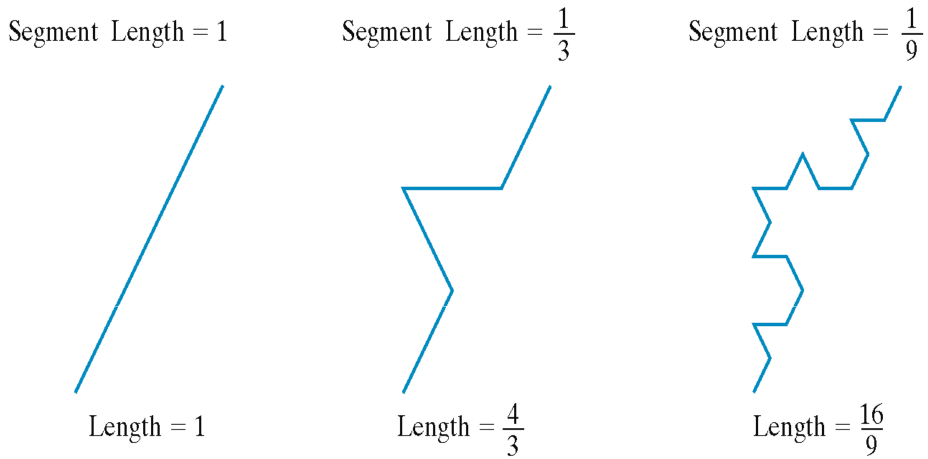
Operation	Surface Limits
Union	A, D
Intersection	C, B
Difference ($obj_2 - obj_1$)	B, D

Images taken from Hearn & Baker, "Computer Graphics with OpenGL" (2004)

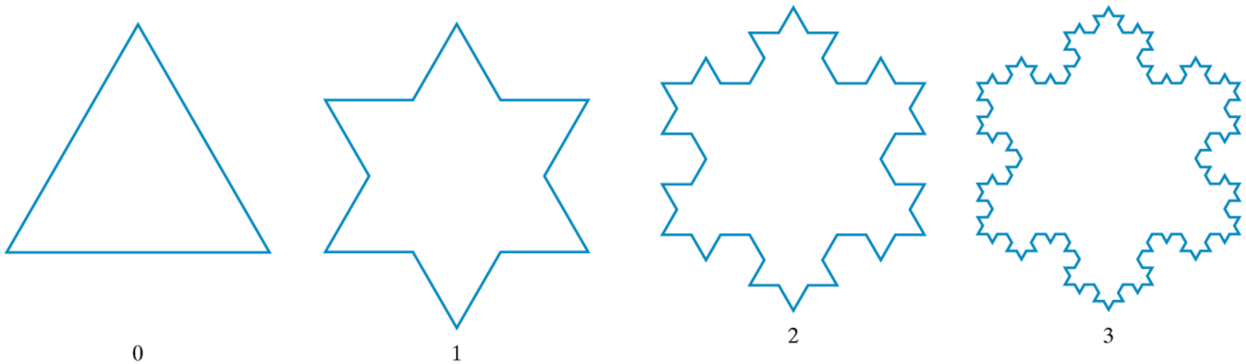
Procedural Modeling

Self-similar fractals

Substitution

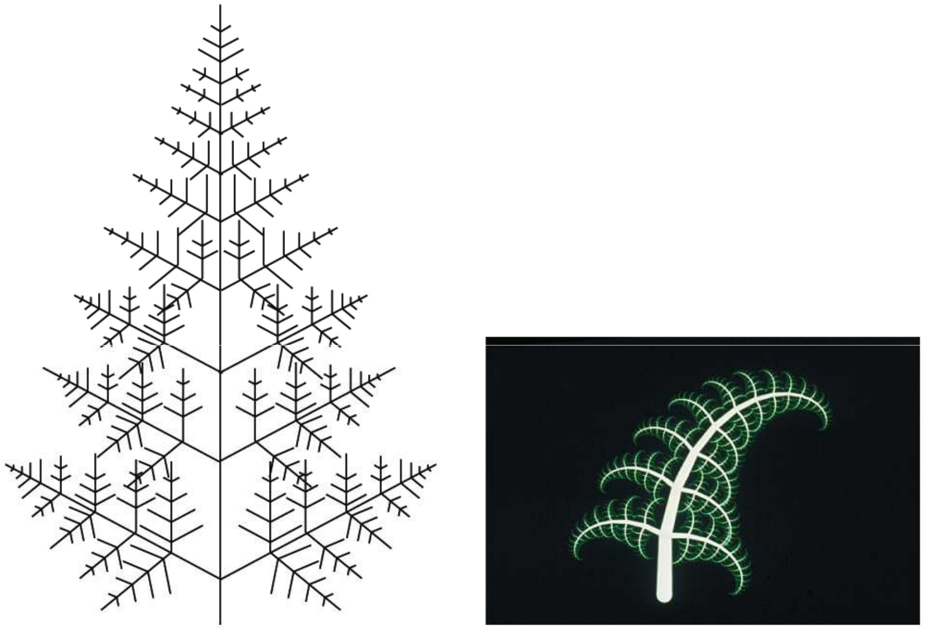


Example: Koch curve



Procedural Modeling

Substitution rules



(a) (b)

Figure 8-76

Self-similar constructions for a fern. (*Courtesy of Peter Oppenheimer, Computer Graphics Lab, New York Institute of Technology*.)

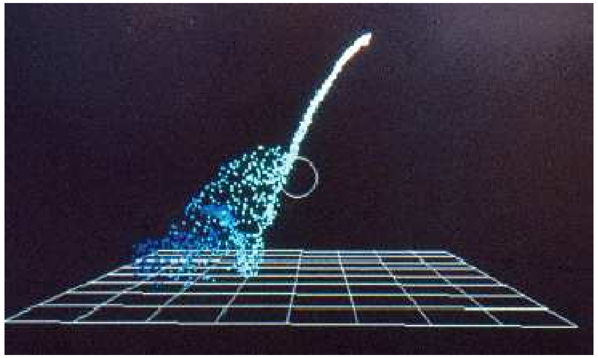
Procedural Modeling

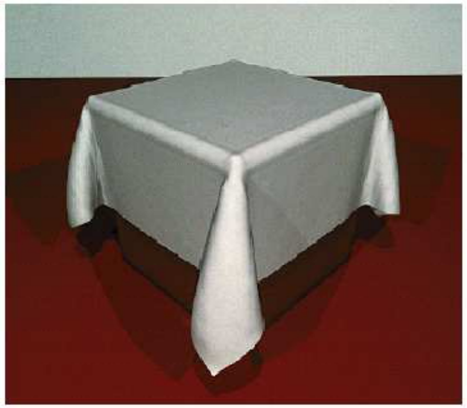
Natural scenes with trees, flowers, and grass

Physically Based Modeling

Particle systems

Shape description is combined with physical simulation





Physically Based Modeling

Procedural modeling + physically based simulation