IV	Polygon clipping-Sutherland Hodgeman algorithm, Weiler-Atherton algorithm, Three dimensional object representation-Polygon surfaces, Quadric surfaces – Basic 3D transformations

THREE DIMENSIONAL OBJECT REPRESENTATIONS

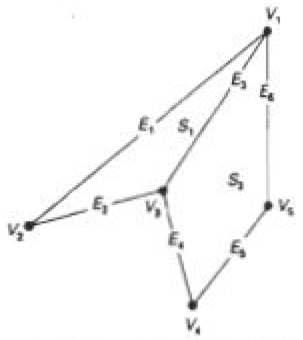
- Representation schemes for solid objects are often divided into two broad categories, although not all representations fall neatly into one or the other of these two categories. Boundary representations (B-reps) describe a three-dimensional object as a set of surfaces that separate the object interior from the environment.
- examples of boundary representations are polygon facets and spline patches.
- Space-partitioning representations are used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non overlapping, contiguous solids (usually cubes).
- A common space-partitioning description for a three-dimensional object is an octree representation

POLYGON SURFACES

- The most commonly used boundary =presentation for a three-dimensional graphics object is a set of surface polygons that enclose the object interior.
- Many graphics systems store all object descriptions as sets of surface polygons. This simplifies and speeds up the surface rendering and display of objects, since all surfaces are described with linear equations

Polygon Tables

- We specify a polygon surface with a set of vertex coordinates and associated attribute parameters.
- As information for each polygon is input, the data are placed into tables that are to be used in the subsequent' processing, display, and manipulation of the objects in a scene.
- Polygon data tables can be organized into two groups: **geometric tables and attribute tables**.
- **Geometric data tables** contain vertex coordinates and parameters to identify the spatial orientation of the polygon surfaces.
- **Attribute** information for an object includes parameters specifying the degree of transparency of the object and its surface reflectivity and texture characteristics.
- A convenient organization for storing geometric data is to create three lists: **a vertex table, an edge table, and a polygon table.**
- Coordinate values for each vertex in the object are stored in the vertex table.
- The edge table contains pointers back into the vertex table to identify the vertices for each polygon edge.
- And the polygon table contains pointers back into the edge table to identify the edges for each polygon.



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.

- Listing the geometric data in three tables, provides a convenient reference to the individual components (vertices, edges, and polygons) of each object. Also, the object can be displayed efficiently by using data from the edge table to draw the component lines.
- An alternative arrangement is to use just two tables: a vertex table and a polygon lable. But this scheme is less convenient, and some edges could get drawn twice. Another possibility is to use only a polygon table, but this duplicates coordinate information, since explicit coordinate values are listed for each vertex in each polygon. Also edge Information would have to be reconstructed from the vertex listings in the polygon table.
- We can add extra information to the data tables of for faster information extraction. For instance. we could expand the edge table to include forward pointers into the polygon table so that common edges between polygons could be identified more rapidly .

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.

- This is particularly useful for the rendering procedures that must vary surface shading smoothly across the edges from one polygon to the next. Similarly, the vertex table could be expanded so that vertices are cross-referenced to corresponding edge..
- Since the geometric data tables may contain extensive listings of vertices and edges for complex objects, it is important that the data be checked for consistency and completeness. When vertex, edge, and polygon definitions are specified, it is possible, particularly in interactive applications, that certain input errors could be made that would distort the display of the object. The more information included in the data tables, the easier it is to check for errors. Therefore, error checking is easier when three data tables (vertex, edge, and polygon) are used, since this scheme provides the most information. Some of the tests that could be performed by a graphics package are

(1) that every vertex is listed as an endpoint for at least two edges,
(2) that every edge is part of at least one polygon,
(3) that every polygon is closed,
(4) that each polygon has at least one shared edge,
(5) that if the edge table contains pointers to polygons, every edge referenced by a polygon pointer has a reciprocal pointer back to the polygon.

Plane Equations

- To produce a display of a three-dimensional object, we must process the input data representation for the object through several procedures. These processing steps include transformation of the modeling and world-coordinate descriptions to viewing coordinates, then to device coordinates; identification of visible surfaces; and the application of surface-rendering procedures.
- The equation for a plane surface can be expressed In the form

$$\mathbf{Ax+By+Cz=0}$$

where (x, y, z) is any point on the plane, and the coefficients A, B, C, and D are constants describing the, spatial properties of the plane.
we can select three successive polygon vertices, (x1, y1, z1,), (x2, y2, z2), and (x3,, y3, z3,), and solve the following set of simultaneous linear plane equations for the ratios A/D, B/D, and C/D:

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

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

Since we are usually dealing with polygon surfaces that enclose an object interior, we need to distinguish between the two sides of the surface. The side of the plain that faces the object interior is called the "inside" face, and the visible or outward side is the "outside" face. If polygon vertices are specified in a counterclockwise direction then viewing the outer side of the plane in a right-handed coordinate system, the direction of the normal vector will be from inside to outside.

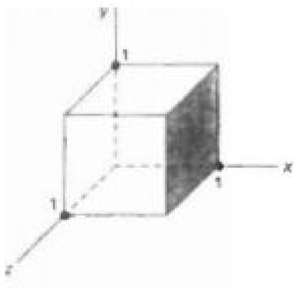


Figure 10-5
The shaded polygon surface of the unit cube has plane equation $x - 1 = 0$ and normal vector $\mathbf{N} = (1, 0, 0)$.

Plane equations are used also to identify the position of spatial points relative to the plane surfaces of an object. For any point (x, y, z) not on a plane with parameters A, B, C, D, we have

$Ax+By+Cz+D \neq 0$

We can identify the point as either inside or outside the plane surface according to the sign (negative or positive) of Ax + By + Cz + D:

- if Ax + By + Cz + D < 0, the point (x, y, z) is inside the surface
- if Ax + By + Cz + D > 0, the point (x, y, z) is outside the surface

QUADRIC SURFACES

- A frequently used class of objects are the quadric surfaces, which are described with second-degree equations (quadratics). They include spheres, ellipsoids, tori, paraboloids, and hyperboloids.
- Quadric surfaces, particularly spheres and ellip- 10-3 soids, are common elements of graphics scenes, and they are often available in Quadric Surfaces graphics packages as primitives horn which more complex objects can be constructed.

Sphere

In Cartesian coordinates, a spherical surface with radius r centered on the coordinate origin is defined as the set of points (x, y, z) that satisfy the equation

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

We can also describe the spherical surface in parametric form, using latitude and longitude angles

$$x = r\cos\phi\cos\theta, \qquad -\pi/2 \leq \phi \leq \pi/2$$
$$y = r\cos\phi\sin\theta, \qquad -\pi \leq \theta \leq \pi$$
$$z = r\sin\phi$$

we could write the parametric equations using standard spherical coordinates, where angle ***Φ*** is specified as the colatitude.). Then, ***Φ*** is defined over the range 0 <= ***Φ***<=***Π***, and ***P*** is often taken in the range 0 <= ***P*** <=2***Π***.

We could also set up the representation using parameters u and v, defined over the range from 0 to 1 by substituting:

Φ=***Π***u.

P =2***Π***v.

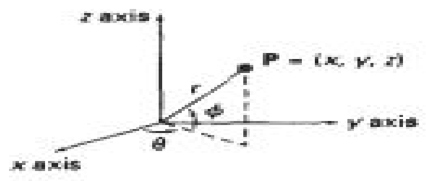


Figure 10-8
Parametric coordinate position (r, θ, ϕ) on the surface of a sphere with radius r.

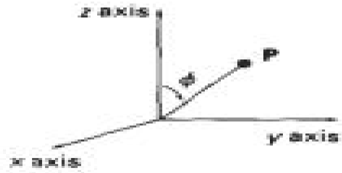


Figure 10-9
Spherical coordinate parameters (r, θ, ϕ), using colatitude for angle ϕ.

Ellipsoid

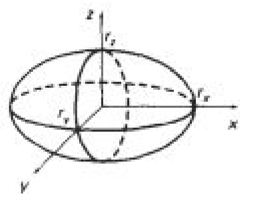


Figure 10-10
An ellipsoid with radii r_x, r_y, and r_z centered on the coordinate origin.

An ellipsoidal surface can be described as an extension of a 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 centered on the origin is :

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

And a parametric representation for the ellipsoid in terms of the latitude angle and the longitude angle is:

$$x = r_x \cos\phi \cos\theta, \qquad -\pi/2 \le \phi \le \pi/2$$
$$y = r_y \cos\phi \sin\theta, \qquad -\pi \le \theta \le \pi$$
$$z = r_z \sin\phi$$

Torus

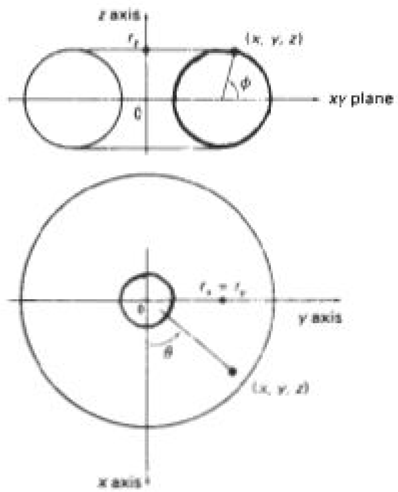


Figure 10-11
A torus with a circular cross section centered on the coordinate origin.

- A torus is a doughnut-shaped object.
- It can be generated by rotating a circle or other conic about a specified axis. The Cartesian representation for points over the surface of a torus can be written in the form :
-

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

- 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 degree. Using latitude and longitude angles ***Φ*** and ***P*** , we can describe the toms surface as the set of points that satisfy

$$x = r_x(r + \cos\phi)\cos\theta, \qquad -\pi \le \phi \le \pi$$
$$y = r_y(r + \cos\phi)\sin\theta, \qquad -\pi \le \theta \le \pi$$
$$z = r_z \sin\phi$$