9. 3D Object Representations

Methods:

- Polygon and Quadric surfaces: For simple Euclidean objects
- Spline surfaces and construction: For curved surfaces
- Procedural methods: Eg. Fractals, Particle systems
- Physically based modeling methods
- Octree Encoding
- Isosurface displays, Volume rendering, etc.

Classification:

Boundary Representations (B-reps) eg. Polygon facets and spline patches
Space-partitioning representations eg. Octree Representation

Objects may also associate with other properties such as mass, volume, so as to determine their response to stress and temperature etc.

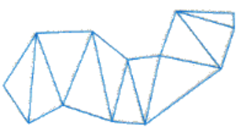
9.1 Polygon Surfaces

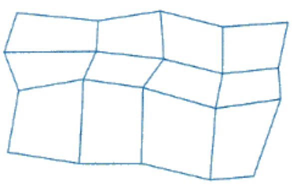
This method simplifies and speeds up the surface rendering and display of objects.

For other 3D objection representations, they are often converted into polygon surfaces before rendering.

<u>Polygon Mesh</u>

- Using a set of connected polygonally bounded planar surfaces to represent an object, which may have curved surfaces or curved edges.
- The wireframe display of such object can be displayed quickly to give general indication of the surface structure.
- Realistic renderings can be produced by interpolating shading patterns across the polygon surfaces to eliminate or reduce the presence of polygon edge boundaries.
- Common types of polygon meshes are triangle strip and quadrilateral mesh.





- Fast hardware-implemented polygon renderers are capable of displaying up to 1,000,000 or more shaded triangles per second, including the application of surface texture and special lighting effects.

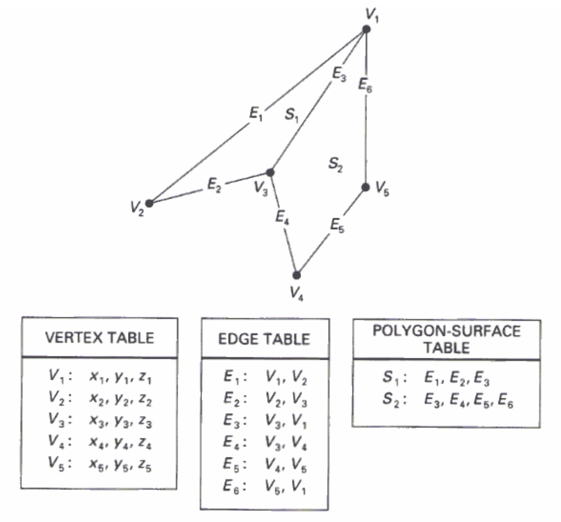
Polygon Tables

This is the specification of polygon surfaces using vertex coordinates and other attributes:

1. Geometric data table: vertices, edges, and polygon surfaces.
2. Attribute table: eg. Degree of transparency and surface reflectivity etc.

Some consistency checks of the geometric data table:

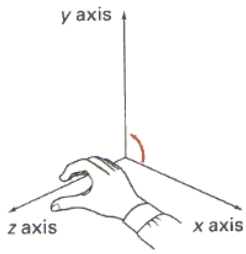
- Every vertex is listed as an endpoint for at least 2 edges
- Every edge is part of at least one polygon
- Every polygon is closed



Plane equation and visible points

Consider a cube, each of the 6 planes has 2 sides: inside face and outside face.

For each plane (in a right-handed coordinate system), if we look at its surface and take 3 points in counter-clockwise direction: (x_1,y_1), (x_2,y_2), and (x_3,y_3), we can compute 4 values: A,B,C,D as



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

Then, the plane equation at the form: Ax+By+Cz+D=0 has the property that:

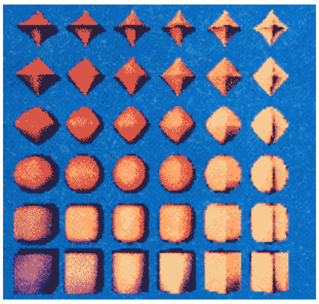
If we substitute any arbitrary point (x,y) into this equation, then,

$Ax + By + Cz + D < 0$ implies that the point (x,y) is inside the surface, and

$Ax + By + Cz + D < 1$ implies that the point (x,y) is outside the surface.

9.2 Curved Surfaces

1. Regular curved surfaces can be generated as
- Quadric Surfaces, eg. Sphere, Ellipsoid, or
- Superquadrics, eg. Superellipsoids

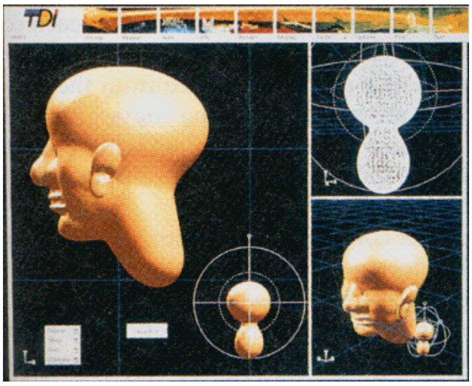


These surfaces can be represented by some simple parametric equations, eg, for ellipsoid:

$x = r_x \cos^{s1}\phi \cos^{s2}\theta, \quad -\pi/2 <= \phi <= \pi/2$
$y = r_y \cos^{s1}\phi \sin^{s2}\theta, \quad -\pi <= \theta <= \pi$
$z = r_z \sin^{s1}\phi$

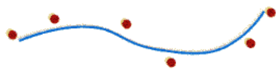
Where s1, r_x, r_y, and r_x are constants. By varying the values of ϕ and θ, points on the surface can be computed.

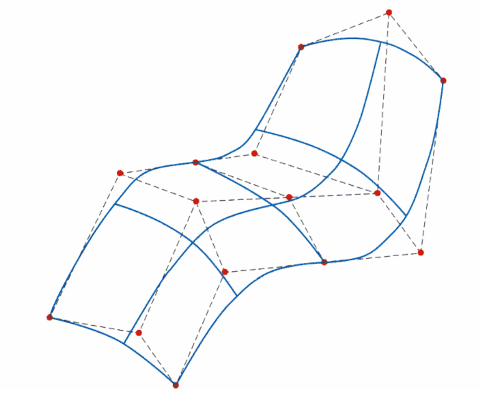
2. Irregular surfaces can also be generated using some special formulating approach, to form a kind of **blobby objects** -- The shapes showing a certain degree of fluidity.



3. Spline Representations

Spline means a flexible strip used to produce a smooth curve through a designated set of points. Several small weights are distributed along the length of the strip to hold it in position on the drafting table as the curve is drawn.
We can mathematically describe such a curve with a piecewise cubic polynomial function => spline curves. Then a spline surface can be described with 2 sets of orthogonal spline curves.

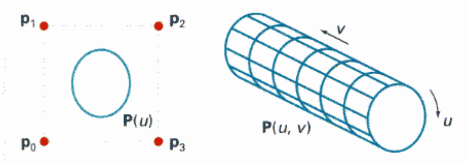




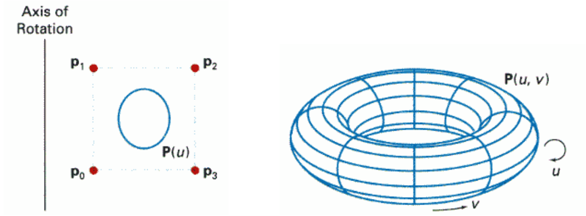
9.3 Sweep Representations

Sweep representations mean sweeping a 2D surface in 3D space to create an object. However, the objects created by this method are usually converted into polygon meshes and/or parametric surfaces before storing.

A Translational Sweep:



A Rotational Sweep:



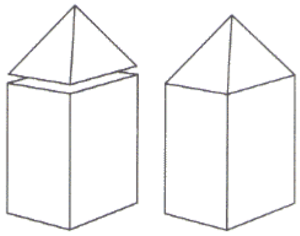
Other variations:
- We can specify a special path for the sweep as some curve function.
- We can vary the shape or size of the cross section along the sweep path.
- We can also vary the orientation of the cross section relative to the sweep path.

9.4 Constructive Solid-Geometry Methods

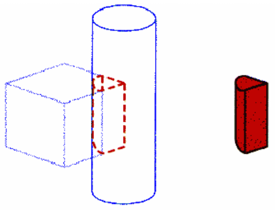
The Constructive Solid-Geometry Method (CSG) combines the volumes occupied by overlapping 3D objects using set operations:

- Union
- Intersection
- Difference

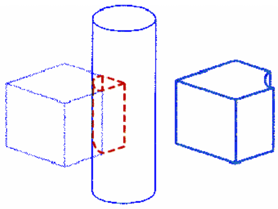
Object created by a union operation:



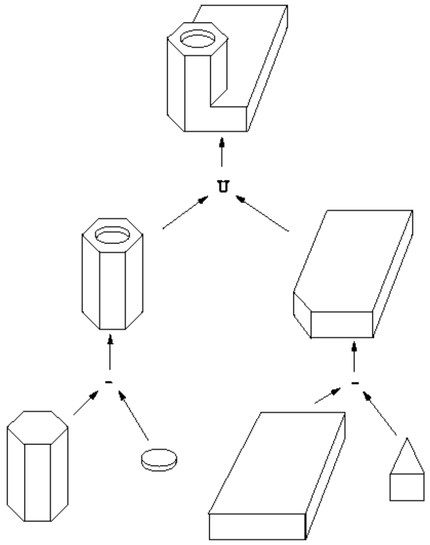
Object created by an intersection operation:



Object created by a difference operation:



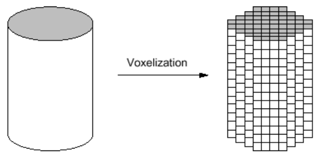
A CSG object can be represented with a binary tree:



9.5 Voxel Representation

In voxel representation, an object is decomposed into identical cells arranged in a fixed regular grid. These cells are called voxels (volume elements), in analogy to pixels.

Eg. A cylinder can be represented as follows by voxels. A '1' may represent inside the cylinder while a '0' may represent outside of the cylinder. Alternatively we may use 8 bits to represent the transparency value. If a voxel has a value of '0', it is a fully-transparent cell. If a voxel has a value of '255', it is a non-transparent cell.

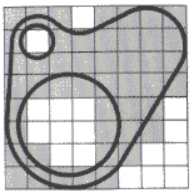


9.6 Octrees

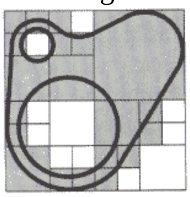
Octrees are hierarchical tree structures that describe each region of 3D space as nodes. When compared with the basic voxel representation, octrees reduce storage requirements for 3D objects. It also provides a convenient representation for storing information about object interiors.

Octree encoding procedure is an extension of the quadtree encoding of 2D images:

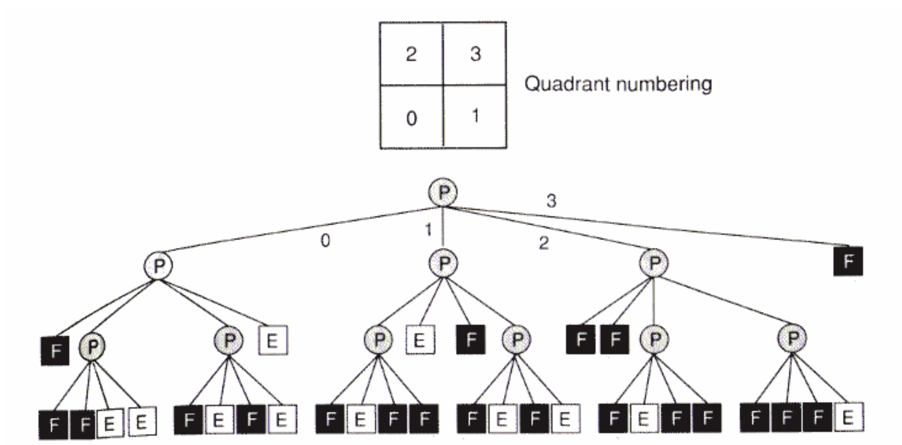
Bitmap representation of a 2D object:



Quadtree encoding of a 2D object:

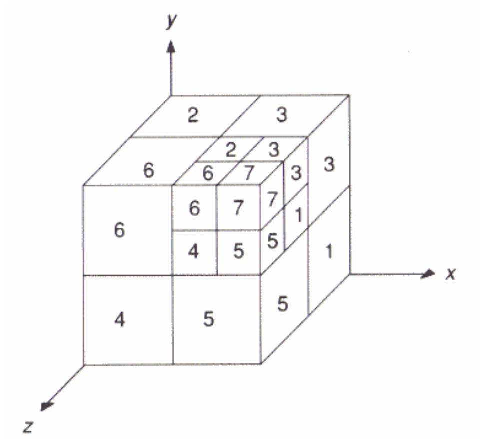


Quadtree data structure for the above object: F = full, P = partially full, E = empty.



The code of this image in quadtree representation is: P PPPF FPPE PEFP FFPP FFEE FEFE FEFF FEFE FEFF FFFE.

For octree representation of 3D data, the numbering method is as follows:



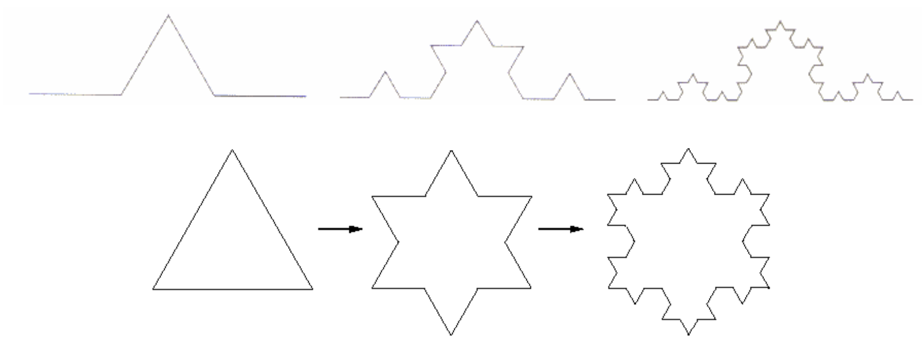
9.7 Fractals

Fractal objects refer to those objects which are self-similar at all resolutions.

Most of the natural objects such as trees, mountains and coastlines are considered as fractal objects because no matter how far or how close one looks at them, they always appear to be somewhat similar.

Fractal objects can also be generated recursively by applying the same transformation function to an object, eg. Scale down + rotate + translate.

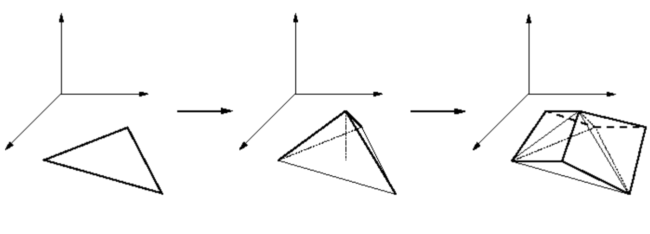
For example, a Fractal Snowflake:



The name "Fractal" comes from its property: fractional dimension. Unlike Euclidean dimension, the dimensions of fractal objects may not be integers, and we call these dimensions as 'fractal dimension'.

Euclidean dimensions and Fractal dimensions:
- A line segment is 1D. If we divide a line into N equal parts, the parts each look like the original line scaled down by a factor of $N = N^{1/1}$.
- A square is 2D. If we divide it into N parts, each part looks like the original scaled down by a factor of $N^{1/2}$.
- For the fractal snowflake, when it is divided into 4 pieces, each resulting piece looks like the original scaled down by a factor of 3, so it has the dimension ***d*** such that $4^{1/d}=3$. That is, d = 1.26.

We may also create 3D objects in a similar way by adding a third dimension. The following example replaces each triangle of the object with a pyramid, by inserting a single point, at each step (except for the bottom face). This will result in a mountain like object, with a regular appearance.



To give a more natural appearance to the created object, we usually allow some limited random variations at each level of recursion: