**VISION INSTITUTE OF MANAGEMENT**

**COMPUTER GRAPHICS & MULTIMEDIA APPLICATION**

**BCA 2nd YEAR/4th SEM**

**UNIT 4(Representing Curves & Surfaces)**

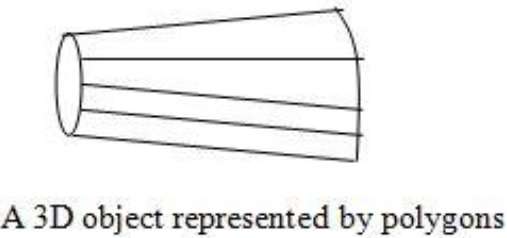
## **Polygon Surfaces**

Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

* **Boundary Representations (B−reps) -** It describes a 3D object as a set of surfaces that separates the object interior from the environment.
* **Space–partitioning representations −** It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids usually cubes

The most commonly used boundary representation for a 3D graphics object is a set of surface polygons that enclose the object interior. Many graphics system use this method. Set of polygons are stored for object description. This simplifies and speeds up the surface rendering and display of object since all surfaces can be described with linear equations.

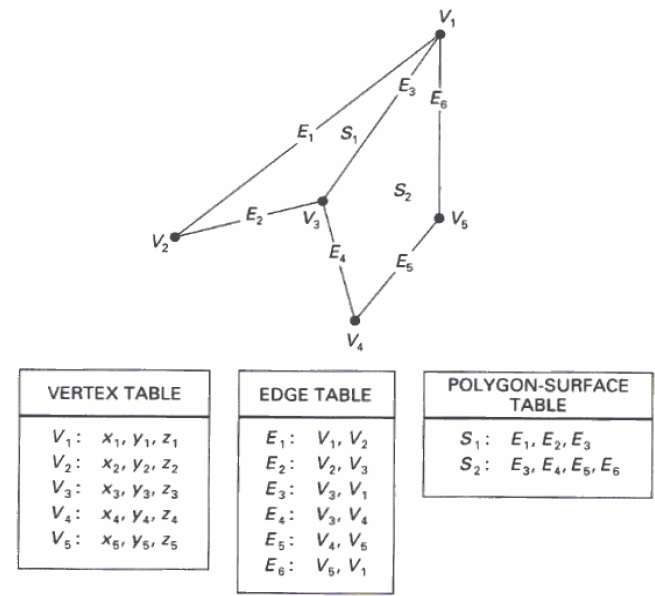
The polygon surfaces are common in design and solid-modeling applications, since their **wireframe display** can be done quickly to give general indication of surface structure. Then realistic scenes are produced by interpolating shading patterns across polygon surface to illuminate.

****

## **Polygon Tables**

In this method, the surface is specified by the set of vertex coordinates and associated attributes. As shown in the following figure, there are five vertices, from v1 to v5.

* Each vertex stores x, y, and z coordinate information which is represented in the table as v1: x1, y1, z1.
* The Edge table is used to store the edge information of polygon. In the following figure, edge E1 lies between vertex v1 and v2 which is represented in the table as E1: v1, v2.
* Polygon surface table stores the number of surfaces present in the polygon. From the following figure, surface S1 is covered by edges E1, E2 and E3 which can be represented in the polygon surface table as S1: E1, E2, and E3.

****

## **Plane Equations**

The equation for plane surface can be expressed as −

Ax + By + Cz + D = 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 obtain the values of A, B, C, and D by solving a set of three plane equations using the coordinate values for three non collinear points in the plane. Let us assume that three vertices of the plane are (x1, y1, z1), (x2, y2, z2) and (x3, y3, z3).

Let us solve the following simultaneous equations for ratios A/D, B/D, and C/D. You get the values of A, B, C, and D.

A/D x1 + B/D y1 + C/D z1 = -1

A/D x2 + B/D y2 + C/D z2 = -1

A/D x3 + B/D y3 + C/D z3 = -1

For any point x, y, z with parameters A, B, C, and D, we can say that −

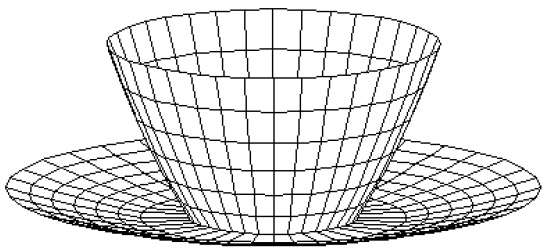
* Ax + By + Cz + D ≠ 0 means the point is not on the plane.
* Ax + By + Cz + D < 0 means the point is inside the surface.
* Ax + By + Cz + D > 0 means the point is outside the surface.

## **Polygon Meshes**

3D surfaces and solids can be approximated by a set of polygonal and line elements. Such surfaces are called **polygonal meshes**. In polygon mesh, each edge is shared by at most two polygons. The set of polygons or faces, together form the “skin” of the object.

This method can be used to represent a broad class of solids/surfaces in graphics. A polygonal mesh can be rendered using hidden surface removal algorithms. The polygon mesh can be represented by three ways −

* Explicit representation
* Pointers to a vertex list
* Pointers to an edge list



### Advantages

* It can be used to model almost any object.
* They are easy to represent as a collection of vertices.
* They are easy to transform.
* They are easy to draw on computer screen.

### Disadvantages

* Curved surfaces can only be approximately described.
* It is difficult to simulate some type of objects like hair or liquid.

# **Computer Graphics Curves**

In computer graphics, we often need to draw different types of objects onto the screen. Objects are not flat all the time and we need to draw curves many times to draw an object.

## **Types of Curves**

A curve is an infinitely large set of points. Each point has two neighbours except endpoints. Curves can be broadly classified into three categories − **explicit, implicit,** and **parametric curves**.

### Implicit Curves

Implicit curve representations define the set of points on a curve by employing a procedure that can test to see if a point in on the curve. Usually, an implicit curve is defined by an implicit function of the form −

f *[Math Processing Error]x, y* = 0

It can represent multivalued curves *[Math Processing Error] multipleyvaluesforanxvalue*. A common example is the circle, whose implicit representation is

**x2 + y2 - R2 = 0**

### Explicit Curves

A mathematical function y = f*[Math Processing Error]x* can be plotted as a curve. Such a function is the explicit representation of the curve. The explicit representation is not general, since it cannot represent vertical lines and is also single-valued. For each value of x, only a single value of y is normally computed by the function.

### Parametric Curves

Curves having parametric form are called parametric curves. The explicit and implicit curve representations can be used only when the function is known. In practice the parametric curves are used. A two-dimensional parametric curve has the following form −

P*[Math Processing Error]t* = f*[Math Processing Error]t*, g*[Math Processing Error]t* or P*[Math Processing Error]t* = x*[Math Processing Error]t*, y*[Math Processing Error]t*

The functions f and g become the *[Math Processing Error]x, y* coordinates of any point on the curve, and the points are obtained when the parameter t is varied over a certain interval [a, b], normally [0, 1].

## **Bezier Curves**

Bezier curve is discovered by the French engineer **Pierre Bezier**. These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve. The Bezier curve can be represented mathematically as −

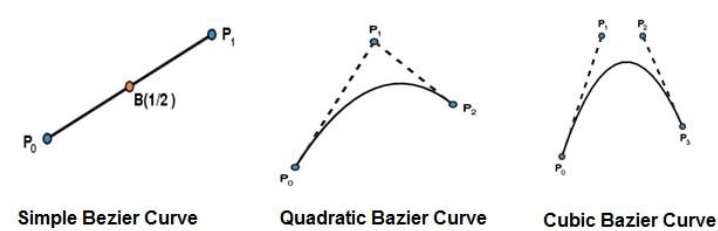
*[Math Processing Error]∑k=0nPiBin (t)*

Where *[Math Processing Error]pi* is the set of points and *[Math Processing Error]Bin(t)* represents the Bernstein polynomials which are given by −

*[Math Processing Error]Bin(t)=(ni)(1−t)n−iti*

Where **n** is the polynomial degree, **i** is the index, and **t** is the variable.

The simplest Bezier curve is the straight line from the point *[Math Processing Error]P0* to *[Math Processing Error]P1*. A quadratic Bezier curve is determined by three control points. A cubic Bezier curve is determined by four control points.



## **Properties of Bezier Curves**

Bezier curves have the following properties −

* They generally follow the shape of the control polygon, which consists of the segments joining the control points.
* They always pass through the first and last control points.
* They are contained in the convex hull of their defining control points.
* The degree of the polynomial defining the curve segment is one less that the number of defining polygon point. Therefore, for 4 control points, the degree of the polynomial is 3, i.e. cubic polynomial.
* A Bezier curve generally follows the shape of the defining polygon.
* The direction of the tangent vector at the end points is same as that of the vector determined by first and last segments.
* The convex hull property for a Bezier curve ensures that the polynomial smoothly follows the control points.
* No straight line intersects a Bezier curve more times than it intersects its control polygon.
* They are invariant under an affine transformation.
* Bezier curves exhibit global control means moving a control point alters the shape of the whole curve.
* A given Bezier curve can be subdivided at a point t=t0 into two Bezier segments which join together at the point corresponding to the parameter value t=t0.

## **What is Solid Modeling?**

Solid modeling is the most advanced method of geometric modeling in three dimensions. Solid modeling is the representation of the solid parts of the object on your computer. The typical geometric model is made up of wire frames that show the object in the form of wires. This wire frame structure can be two dimensional, two and half dimensional or three dimensional. Providing surface representation to the wire three dimensional views of geometric models makes the object appear solid on the computer screen and this is what is called as solid modeling.

## **Advantages of Solid Modeling**

Solid modeling is one of the most important applications of the CAD software and it has been becoming increasingly popular of late. The solid modeling CAD software helps the designer to see the designed object as if it were the real manufactured product. It can be seen from various directions and in various views. This helps the designer to be sure that the object looks exactly as they wanted it to be. It also gives additional vision to the designer as to what more changes can be done in the object.

## **Process of Making the Solid Models**

To make the solid models you have to first make the wire frame model of the object and convert it into 3D view. Thereafter the surfaces are added to the 3D wire model to convert it into 3D solid model. For creating the solid models you need to have special CAD software that can create solid models. One of the most popular CAD software for solid modeling is SolidWorks. Its latest version is SolidWorks 2009. A number of other CAD software like AutoCAD and others also have features of creating the solid models.

## **Applications of Solid Modeling**

Solid modeling is used not only for creating solid models of machine parts, but also the buildings, electric circuits and even of the human beings. The solid modeling software are being used for a large variety of applications, here are some of them:

* 1. **Engineering**: The engineering design professionals use solid modeling to see how the designed product will actually look like. The architects and civil engineers use it to use the layout of the designed building.
  2. **Entertainment industry**: The animation industry has been using solid modeling to create various characters and the movies out of them.
  3. **Medical industry**: Modern imaging scanners are being used to create the solid models of the internal parts of the body. This helps the doctors to visualize specific tissues of the body, designing various medical devices etc.

# **Boundary Representations**

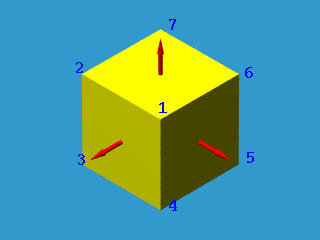
*Boundary Representation*, or *B-rep* for short, can be considered as an extension to the wireframe model. The merit of a B-rep is that a solid is bounded by its surface and has its *interior* and *exterior*. The surface of a solid consists of a set of well-organized faces, each of which is a piece of some surface (*.e.g.*, a surface patch). Faces may share vertices and edges that are curve segments. Therefore, a B-rep is an extension to the wireframe model by adding face information to the latter.

**There are two types of information in a B-rep:**

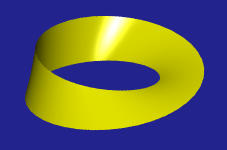
1. *Topological*
2. *Geometric*

*Topological* information provide the relationships among vertices, edges and faces similar to that used in a wireframe model. In addition to connectivity, topological information also include *orientation* of edges and faces. *Geometric* information are usually equations of the edges and faces.

The orientation of each face is important. Normally, a face is surrounded by a set of vertices. Using the right-handed rule, the ordering of these vertices for describing a particular face must guarantee that the normal vector of that face is pointing to the exterior of the solid. Normally, the order is *counter clockwise*. If that face is given by an equation, the equation must be rewritten so that the normal vector at every point on the part that is being used as a face points to the exterior of the solid. Therefore, by inspecting normal vectors one can immediately tell the inside and outside of a solid under B-rep. This orientation must be done for *all* faces. The following shows three faces and their outward pointing normal vectors. To describe the top surface, the vertices should be 6, 7, 2, 1 or 7, 2, 1, 6 or 2, 1, 6, 7 or 1, 6, 7, 2. To describe the left face, the order should be 1, 2, 3, 4 or 2, 3, 4, 1 or 3, 4, 1, 2 or 4, 1, 2, 3.



Unfortunately, not all surfaces can be oriented this way. If the surface of a solid can be oriented this way, it is called *orientable*; otherwise, it is *non-orientable*. The following shows the well-known Mobius band which is one-sided and non-orientable.



**Spatial Partitioning**

## **Introduction: -**

In spatial partitioning representations, a solid is decomposed into a collection of adjoining, nonintersecting solids. A quadtree is a representation format used to encode images. The fundamental idea behind the quadtree is that any image can be split into four quadrants. Each quadrant may again be split into four sub quadrants, etc. In the quadtree, a parent node represents the image, while four child nodes, in a predetermined order, represent the four quadrants.

The following are different spatial partitioning techniques:

1. Cell Decomposition
2. Spatial Occupancy Enumeration
3. Quadtrees and Octrees
4. Binary space partitioning Trees

## 

## **Objective: -**

To demonstrate the spatial partitioning technique using quadtrees.

## **Description: -**

The project can be described as follows-

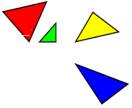
Given a set of polygons obtain the spatial organization of the polygons.

**Input: -**

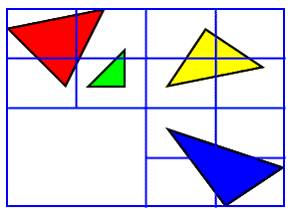
Set of vertex coordinates of the polygons.

**Output: -**

1. Display the input polygons.



2. Display the partitioned space.



This is just an example not the exact output of the partitioned space, still more partitioning required.

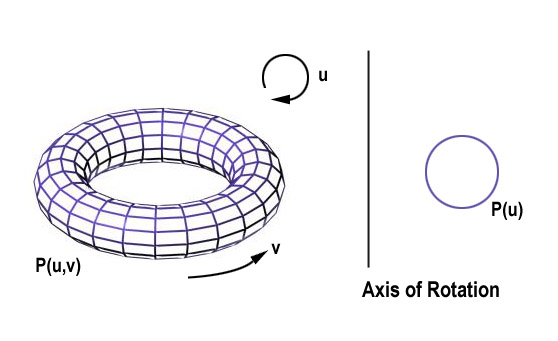
# **Constructive Solid-Geometry Methods**

### Constructive models represent a solid as a combination of primitive solids. (CSG)

## **Sweep Representation**

### Specifying a 2D shape and a sweep that moves the shape through a region of space.

We perform a sweep by moving the shape along a path. At intervals along this path, we replicate the shape and draw a set of connecting line in the direction of the sweep to obtain the wireframe representation.

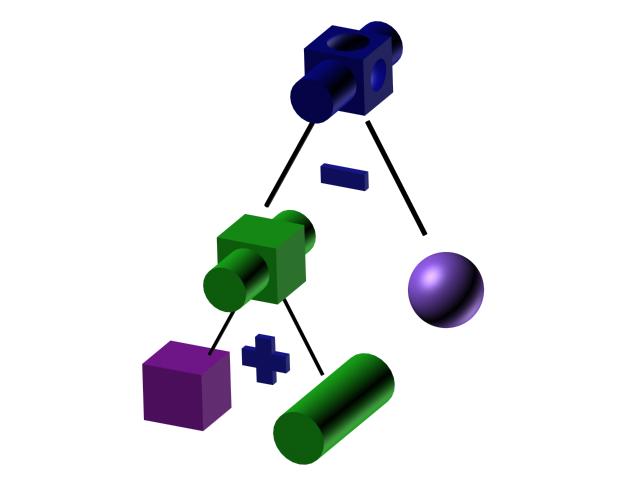


**Example of torus designed using a rotational sweep. The periodic spline cross section is rotated about an axis of rotation specified in the plane of the cross section.**

## **Constructive Solid Geometry (CSG)**

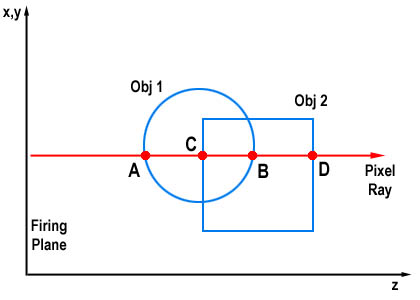
### 1. Definition

* Combine volume occupied by overlapping 3D objects using set boolean operations
* Each primitive is defined as a combination of half-spaces.
* Typical standard primitives are:   
  cone, cylinder, sphere, torus, block, closed spline surface, right angular wedge. Swept solids (a revolution or linear sweep of a planar face which may contain holes.
* Operations are union, intersection and difference.



### 2. Implementation with ray casting

Ray Casting is often used to implement CSG operation when objects are described with boundary representation. We fire a ray from the plane xy (which represent the screen). The surface limits for the composite object are determined by the specified set operation (see Table).

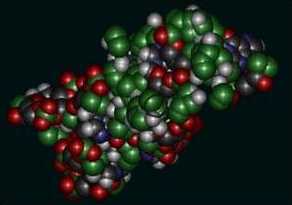


|  |  |
| --- | --- |
| **Operation** | **Surface Limit** |
| Union | A, D |
| Intersection | C, B |
| Difference (Obj2 - Obj1) | B, D |

  This method can also be used for physical simulation.

## **Visualization**

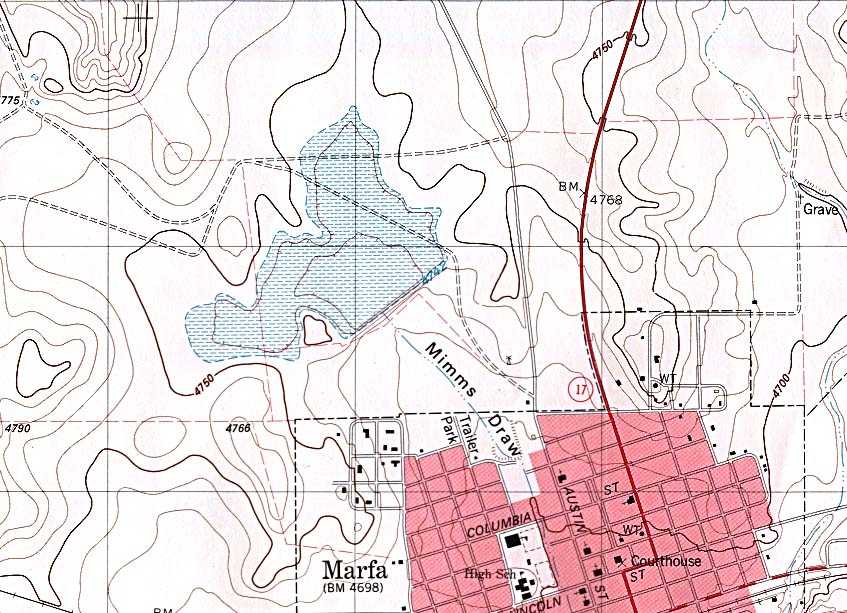
The use of graphical methods to aid in analysis, often of datasets so large or complex they cannot be conceptualized otherwise. The analysed data can be scalars, vectors, tensors, or a combination of these types.



**An example of visualization. Proteins are too complex to describe verbally**

**Scalar Fields**

* Graphs/charts
* Pseudo-color
* Contour plots, isoclines



**A topographic map of Marfa, Texas. Topographical maps are the most common utilization of isolinear plots.**

**Volume Rendering**

Project the data into 2D using ray casting, but the 2D value can be some function of the values along that ray.

**Examples:**

* Max/min values
* Accumulate the opacity of points until it reaches 1 or the ray passes through (X-rays).

**Spatial Data Structures**

We will use hierarchical tree structures to store object information.

Why?

* Reduce Storage Space
* Speed up visibility computations

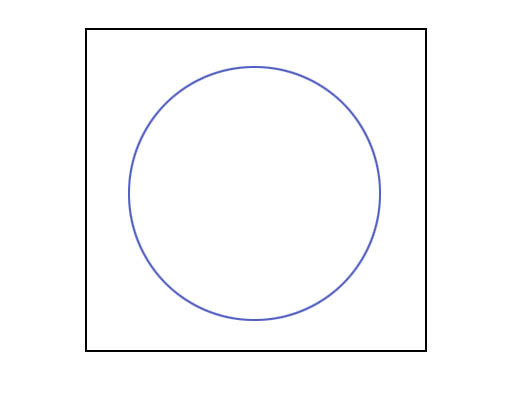
Let's start with 2D.

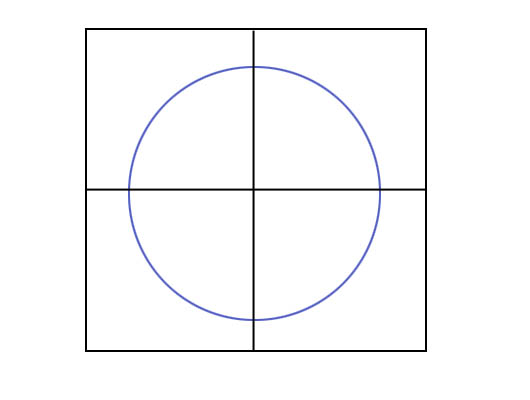
**Quadtrees**

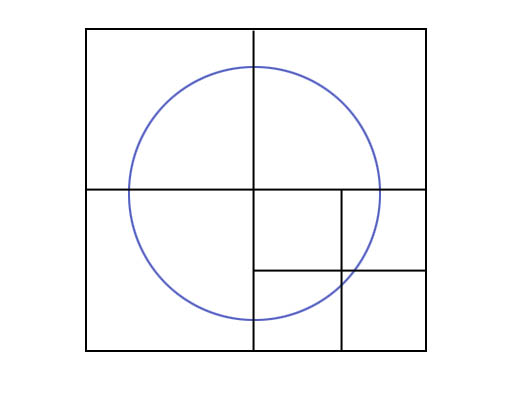
**Definition**

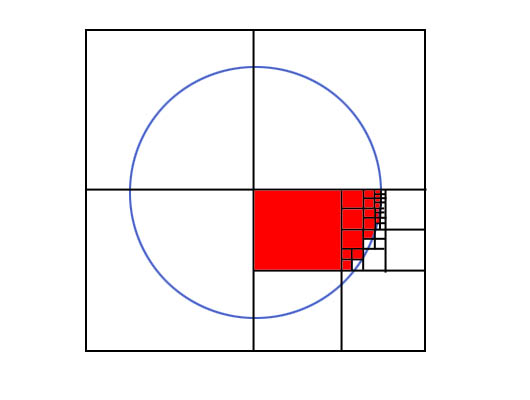
* Based on the divided-and-conquer principle, a quadtree is a data structure that recursively subdivides a plane into 4 quadrants.
* The decision to subdivide is based on an attribute of the current quadrant. If the quadrant is heterogeneous with respect to this attribute, further subdivision occurs. If it is homogeneous, or we have reached the desired level of detail, we stop subdividing.

We will apply quadtrees to storing planar polygons. Our attribute will be whether current quadrant is filled by the interior of a polygon:

****

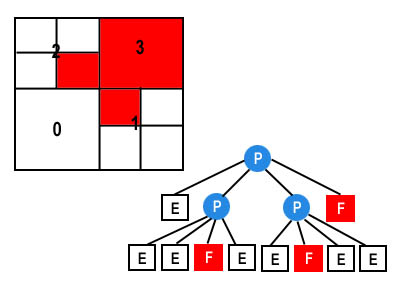
****

****

****

The successive subdivisions can be represented as a tree with heterogeneous quadrants as nodes and homogeneous quadrants as leaves.

Here, we are grouping by color:

****

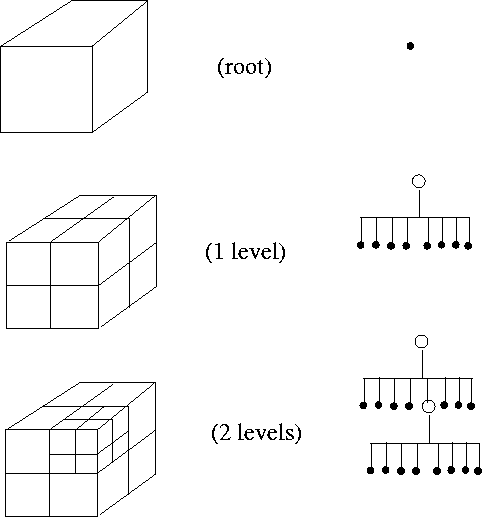
**F: Full; E: Empty; P: Partially Full**

For an area containing 2^n by 2^n pixels, a quadtree representation contains at most n levels.

**Octrees**

* Octrees are based on the same principle, but divide regions of 3D space (usually into cubes).
* The scene is subdivided at each step with three mutually perpendicular planes, aligned with the Cartesian coordinate planes.
* Individual partitions of 3D space are called Voxels (Volume Elements).
* Applications: ray casting, shadow casting.

Octrees, like Quadtrees, use a node structure to store the volume elements:



In this figure, the octree has been refined twice. First, the root is refined into eight cells, each representing an octant of the root's domain. Then, one of the root's children is refined yet again.

Here is an example of a fully subdivided torus:

