Chapter 4

Geometric Modeling for RP&T

4.1 Introduction to Geometric Modeling

All physical objects in the world are three dimensional in nature. However, in order to convey one's own idea of a 3D object in his mind, it is not always possible and often not practicable to show by creating the physical shape in materials such as clay. Therefore, there was always an urge to depict 3D shapes as realistically as possible without actually creating their physical models. Since paper happens to be the easiest medium, techniques were developed by which it is possible to represent 3D objects by means of one or more 2D sketches on it. Some of these techniques used in depicting engineering objects are orthographic projections, isometric views etc. When very huge objects such as bridges, buildings etc. are to be depicted, perspective views are used. All these are 2D views which require some amount of imagination by the person to interpret them. The collection of these techniques used to represent 3D object in terms of 2D sketches and views is called *Engineering Graphics*. When use of computers is employed for the above Engineering Graphic work, Engineering Graphics comes to be known as *Computer Graphics*. (It may be noted that this definition of Computer Graphics no longer holds good since today Computer Graphics encompasses all techniques that are used for representation of objects, interaction with object models, preparation and presentation of the views).

Although initially only 2D details could be depicted in a computer, techniques to represent 3D details were developed which led to the possibility of representing objects in 3D. The schemes developed for this purpose are basically of three types:

1. Wire-frame representation
2. Surface representation
3. Solid representation.

In order to understand these representations, it is important to know the meaning of two terms, viz., *geometry* and *topology* since these two details are required to define an object. Geometry describes the location and size of entities whereas topology tells how the entities are connected among themselves. In other words, topology of an object describes how faces are bounded by edges, how edges are shared by faces, how vertices are shared by faces, how vertices are shared by edges and so on. For example, coordinates of a point, the radius of a circle etc. will be called geometry whereas the information as to whether a given arc or a corner is concave or convex will be called topology. Figure 4-1 has two shapes which have the same geometry but different topology.

Wire-frame, surface and solid representations are distinguished by the extent of topological information they have. Table 4-1 shows the geometric and topological information stored typically by these representations. From this table, it is obvious that the former model is a subset of the latter ones in that order. Because of this characteristic, all solid modeling systems make use of the wire-frame and surface models for quick intermediate displays.

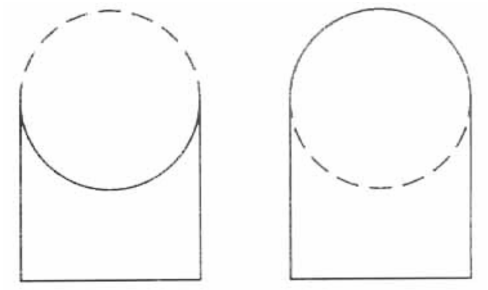

Figure 4-1 Same geometry but different topology

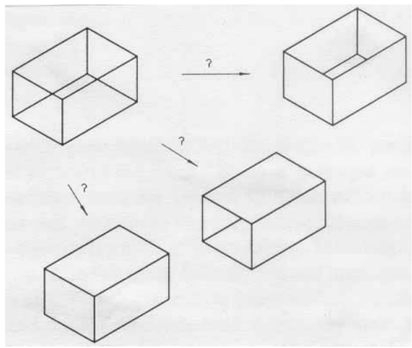

Figure 4-2 Different interpretations of a box in wire-frame modeling

Table 4-1 Geometric and Topological Information Stored in Various Types of Geometric Models

Geometric Model	Geometry	Topology
Wire-frame model	Vertices	Edges (Connectivity among vertices)
Surface model	Vertices	Edges Faces (Connectivity among edges)
Solid model	Vertices	Edges Faces Solids (Connectivity among faces)

4.2 Features of Wire-Frame Models

A wire-frame model is analogous to obtaining the shape of an object by welding the wires representing the edges together. It is the simplest of 3D representation schemes and hence the fastest. Therefore, surface and solid modelers make use of it for generating quick displays. However, the usefulness of wire-frame models is limited because of the following drawbacks:

- A wire-frame model is ambiguous since the same model can be interpreted in several ways. This is illustrated in Figure 4-2 where a block is shown in three different interpretations.
- It is possible to create nonsense objects in wire-frame representation. The examples shown in Figure 4-3 illustrate this.
- The topological information available in a wire-frame representation is inadequate to create hidden line removed views. Therefore, the objects looks messy making interpretation difficult.
- Calculation of mass properties such as surface area, volume, mass, center of gravity, moment of inertia etc. is not possible for a wire-frame model since only edges are available.

4.3 Features of Surface Models

A surface model can be of two types, viz.,

1. Faceted or tessellated surface model
3. Exact surface model.

A faceted surface model is analogous to obtaining the shape by pasting together the cardboard pieces that represent the faces. Since it is simple and a subset of any solid modeler, it is often used for generating quick hidden line removed displays. Till recently, it was possible to use only simpler shapes such as block, wedge, sphere, cylinder, cone and torus in solid modelers. Therefore, for applications that require freeform surface modeling as in styling and applications where accuracy is very important as in NC path generation, exact surface modelers were used in these areas. The exact surface modelers most commonly make use *of Non-Uniform Rational B-Spline (NURBS)* data structure. The advantage of NURBS is that it can represent all kinds of geometries right from straight line to more complicated higher order free-form surfaces such as the surface of a turbine blade using a unified data type. However, such surface modelers are almost extinct since solid modelers today can handle with greater ease freeform surfaces also. Some of the limitations of surface modelers are:

- Faceted surface models are inaccurate. If accuracy is to be improved, the size of the model becomes huge.
- Nonsense objects given in Figure 4-3 can be possible in a surface model also.
- Stitching of edges exactly is difficult.
- On a pure surface model, only surface area can be calculated and other mass properties are meaningless since details inside these surfaces are not defined.

4.4 Features of Solid Models

A solid model is said to be an unambiguous and informatically complete representation of a physical object. A solid model can be created in several ways but the resulting model can be interpreted only in one way. Since the inside of a solid is well defined, all mass properties can be calculated. Although the solid representation is the costliest of the geometric models discussed so far in terms compute space (RAM and disc space) and time, the rapidly falling the price to performance of computer hardware has overcome this difficulty. It may be noted that in the last one year, the cost of RAM alone has fallen by three-fourth.

Over the period, several solid model representations have been developed. Some of them are:

- Constructive Solid Geometry models (CSG)
- Boundary Representation models (B-Rep.)
- Spatial decomposition models
- Feature-based models.

Each one of them has its own advantages and limitations and accordingly they also find specific applications. Among these, CSG and feature-based models are called unevaluated models and the other two are in evaluated form. The details of these models are discussed in this section.

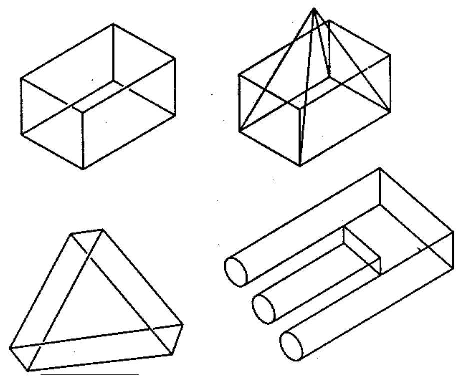

Figure 4-3 Some nonsense objects possible in wire-frame and surface models

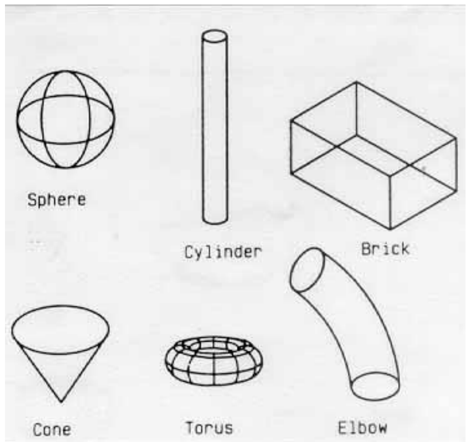

Figure 4-4 Primitives used in CSG models

4.4.1 Constructive Solid Geometry Models (CSG)

Constructive Solid Geometry Models (CSG) are the first solid models to be developed at University of Rochester. Certain geometries such as blocks and cylinders as shown in Figure 4-4 are very familiar to the users and the concept in modeling is to use these simpler known geometries called *primitives* to construct more complicated objects. The process of construction makes use of three operators called Boolean operators. These operators are:

1. Union
2. Subtraction
3. Intersection.

Just as one would add and subtract numbers, in CSG modeling, the geometries are manipulated using these operators. It is possible to represent the primitives and Boolean operators used in constructing any object in the form of a tree called *CSG tree*. The leaves of the CSG tree of any object will be the primitives used and the branches will be the Boolean operators. The root of the tree is the constructed object itself. A few other operators are also used for editing operations such as filletting and chamfering and transformations. However, they are not shown in CSG tree. The example of the CSG tree of a tappet valve is shown in Figure 4-5.

Advantages of CSG models

- Simple to understand and use.

- Low memory requirements.

Limitations of CSG models

- CSG is an unevaluated or raw model. Therefore, for any operation such as display or mass property calculations, it takes more time since all the necessary faces and edges are to be calculated every time.
- Data accession becomes more difficult as the complexity of the object increases.

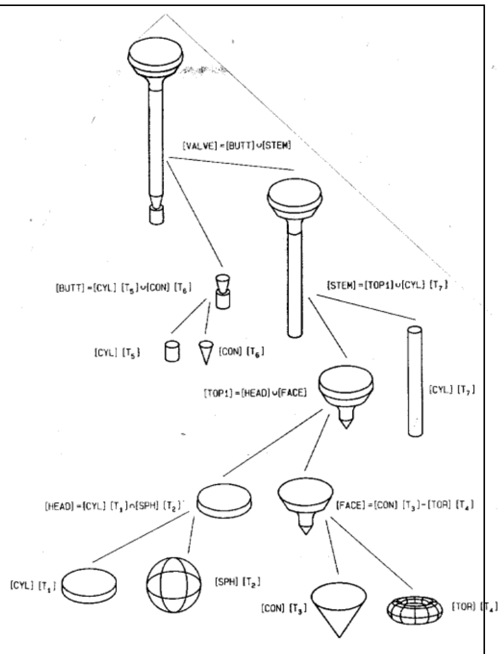

Figure 4-5 CSG of a tappet valve

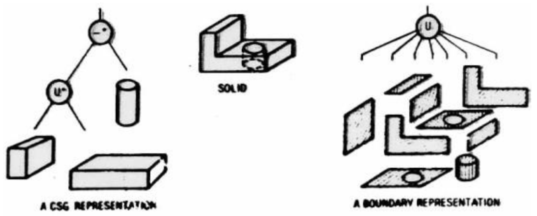

Figure 4-6 B-Rep. Vs. CSG representations

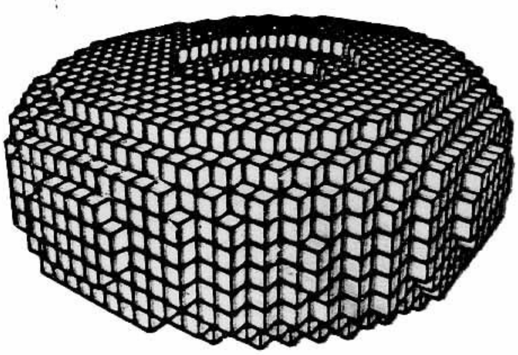

Figure 4-7 Solid representation by exhaustive enumeration

4.4.2 Boundary Representation models (B-Rep)

Boundary representation is a very powerful solid modeling scheme that makes use of the concept of half spaces. Any surface that can be mathematically represented, be it a plane, sphere etc., divides the universe into two halves. Any object boundary is constructed out of such surfaces each of which divide the universe into two halves. While the equation of these surfaces are equalities, the corresponding half spaces are represented by the respective inequalities. One of these inequalities (or half space) represent the side on which material is present. A boundary representation model (B-Rep.) is obtained as an intersection of all inequalities of the object boundaries corresponding to the material side. B-Rep. models are generally represented by a *winged data structure* with adequate redundancy for the benefit of speed. Figure 4-6 shows the details of an object in B-Rep. modeling.

Advantages of B-Rep models

- Algorithms work very fast due to its presence in evaluated form and data redundancy.

Limitations of B-Rep models

- B-Rep. models occupy large space.
- They are difficult for a user to create since he has to calculate the intersections of various surfaces. In other words, they are less user-friendly.
- B-Rep. models have large amount of data redundancy.

The topology of the object may be disturbed during manipulations if adequate care is not taken leading to nonsense objects.

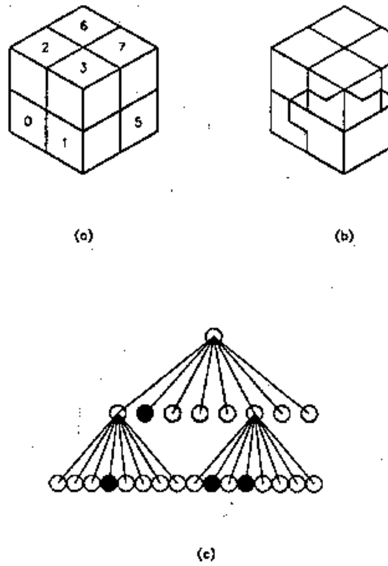

Figure 4-8 Octree representation of a part a) Universe b) Component c) Octree

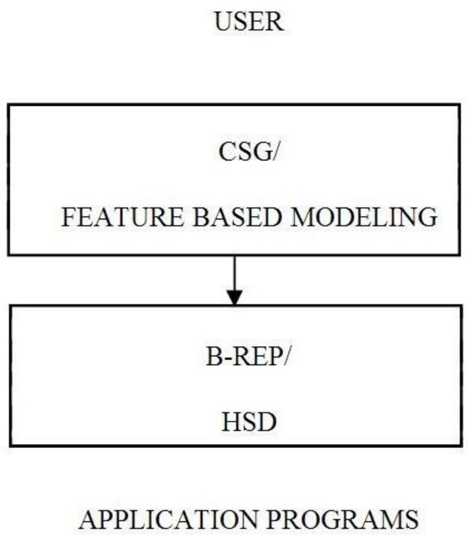

Figure 4-9 A practical solid modeling system

4.4.3 Spatial Decomposition Models

Spatial Decomposition Models are of two types, viz.,

1. Exhaustive enumeration models and
2. Hierarchical Space Decomposition (HSD).

The exhaustive enumeration techniques represent the object in terms of equally sized cubes or spheres or rods of equal diameter (See Figure 4-7). This technique, although very simple, is of no practical use since for the kind of resolution required, it becomes very huge. To overcome this difficulty, HSD models were introduced. It may be of many types such as octree, bintree, polytree etc. As shown in Figure 4-8, the universe (a cube that contains the object) is subdivided into 8 cubes by halving it across the principal directions. Each of these 8 cubes are categorized as FULL if they lie fully inside the object, EMPTY if they do not touch the object at all and PARTIAL otherwise. While EMPTY and FULL nodes become the leaves of the tree called *Octree*, the root will be the object itself. The PARTIAL cubes are the intermediate nodes of this tree and hence are further subdivided and categorized in the same manner recursively till the required resolution is reached. Because of the nature of the HSD models, they are commonly used in places where the number of primitives and Boolean operations are vast as in the case of NC simulation. They are also ideal to represent ill-defined lump geometries such as terrains, tomography etc. It may be noted that the size of the octree of the object depends on only its surface area for any given resolution and not on the number of primitives and Boolean operations.

Advantages of HSD models

- Memory required by HSD models is independent of the number of primitives and operations. For a given resolution, memory required depends only on the surface area of the object.
- Boolean operations, rendered display in isometric view, scaling up/down in powers of 2, orthogonal rotations etc. are trivially simple since these operations require only tree traversals with simple exchange of terms.
- One can choose any desired accuracy (of course, at the cost of memory and speed)
- It inherently lends itself to parallelization.

Limitations of HSD models

- Octree is an approximate representation.
- Memory required increases exponentially with increase in resolution.
  Feature details are irretrievably lost.

4.4.4 Feature-Based Models

The word *features* refers here to the manufacturing features such as holes, slots, bosses etc. The dimensions are always stored as parameters inside this model. If a hole is to be called a hole, the full circular cross-section of it must be within the object lest it becomes a slot. Therefore, the parameters of any feature cannot take any value; these values have to be within certain constraints. These constraints may be geometric relations such as perpendicularity, parallelism, concentricity etc. or they can be dimensional values such as lengths and angles. Therefore the term *Feature-Based Model* actually refers to *Feature-Based Constrained Parametric Model*. By changing the values of the parameters within the constraints, one can get a variety of shapes and sizes. Therefore, one can actually design a family of parts rather than a part in the same time. This leads to the possibility of productivity improvement of manyfold if proper planning is done by the user.

Procedure for constructing a Feature-Based model

- Choose a sketch plane.
- Sketch a rough 2D sketch on the sketch plane. This gives roughly the topology.
- Just constrain this sketch in the following three levels:
  - Use the rules defined internally
  - Add more relational constraints
  - Add dimensional constraints.

- Convert this 2D sketch into a 3D feature such as extrusion, sweep, cutout, slot, hole etc.
- The first feature created in this manner is called *base feature*. Use the above steps to create all other features.

Advantages of FBM models

- Unlike CSG, in feature-based modeling, the Boolean operators are not explicit. They depend on the feature characteristics.
- By changing a few parameters, the object can be changed unambiguously since all the dimensions are related to each other by these parameters. In this way, a family of parts can be designed with the same effort required to design a part in CSG modeling.
- It enforces just dimensioning.
- It can do automatic dimensioning.
- Due to the presence of constraints, even if the dimensions are changed, the topological relations are preserved. For instance, a through hole remains as a through hole even if the thickness of the plate is increased.
- Creation of 2D sketches and their conversion into 3D using familiar features makes this approach more elegant and natural.

Limitations of FBM models

- Comparatively higher intelligence is required for the users of feature-based modeling. Solving constraints requires considerable amount of geometric acumen.
- User should plan well in deciding the right parameters to exploit the benefits of this group design philosophy.

4.5 Conclusions

From the earlier discussions, it is clear that no single solid model scheme meets all the requirements. Modelers such as CSG and Feature-based modelers are very user-friendly but they are computationally inefficient, i.e., they are not system-friendly. On the other hand, B-Rep. and HSD schemes are highly system-friendly but are poor in user-friendliness. Therefore, any practical solid modeling system today invariably makes use of dual or multiple representations - one being user-friendly and the other being system-friendly. This is illustrated in Figure 4-9. Some examples are given in Table 4-2. In spite of all these developments, we are still not in a position to represent a physical object realistically. Because an object is not just its surface which we see but has different interior in terms of density, color, mechanical properties like hardness, smell, taste etc. Never ending pursuit for the same is on!

Table 4-2 Architecture of some practical solid modeling systems

Software	User-friendly Modeling	System-friendly Modeling
AutoCAD AME	CSG	B-Rep.
AutoCAD Designer	Feature-based	B-Rep.
EDS-Unigraphics	CSG (& Feature-based)	B-Rep.
Pro/Engineer	Feature-based	B-Rep.
I-DEAS	CSG	B-Rep.