# Lecture notes: Object modeling

One of the classic problems in computer vision is to construct a model of an object from an image of the object. An object model has the following general principles:

- **Compact representation** – The data necessary to describe the model should be much smaller (orders of magnitude) less than the original image data.

For a model created from image data, the model encodes a large but finite point set (the pixels), but for a model created from human imagination, the model encodes an essentially infinite point set.

- **Retains important features** – The model should convey the interesting features of the object such as its shape and color.

The relative importance of features depends upon the task for which the model will be used. Visualization tasks can require very detailed models, while navigation tasks can use coarse models to avoid obstacles.

- **Conductive to answering questions** – The model should allow for the calculation of properties about the object, such as its size and position.

Again, this can be task dependent. Consider for example the Arabic and Roman number models. We can model the number fifty-four as 54 in Arabic and LIV in Roman. Both are useful for providing a compact representation of fifty-four things, but one is much easier to use for subtraction and addition. What is LIX + CVII? It is easier to calculate if we write it as 59 + 107.

Just as with numbers, there are several different types of modeling systems for objects. These notes cover 6 different methods.

(1) **Mesh**. A mesh model represents an object using a large number of triangles. Each triangle represents a small piece of a surface.

- Triangles can each be given a different color or texture.
- The triangles are all planar and so can be easily rotated, translated, and rendered using ray tracing.
- A model is easily fit to range image data by making a triangle for every 3 neighboring pixels.
- Multiple views (images) are relatively easy to combine. The union of 3D point clouds from each image is found, and then a nearest-neighbor method is used to connect points and make triangles.
- Meshes are good for visualization but poor for answering geometric questions. They do not generally provide information on the number of surfaces of an object or relations between object parts. Although they can be more compact than original range data, they are not nearly as compact as some other models.

Figures 1 and 2 show some example meshes.

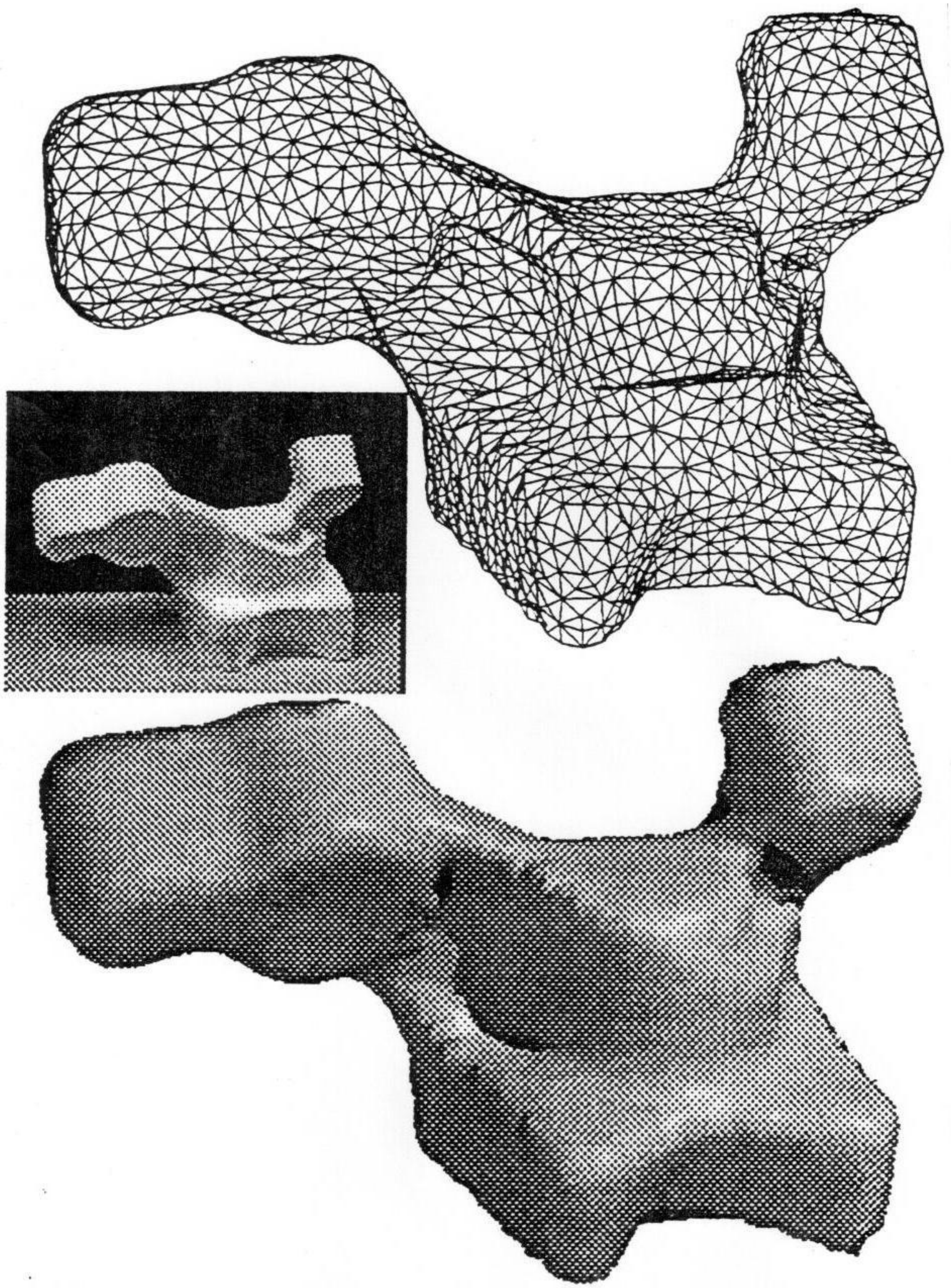

Figure 1:  A mesh model of a car part.

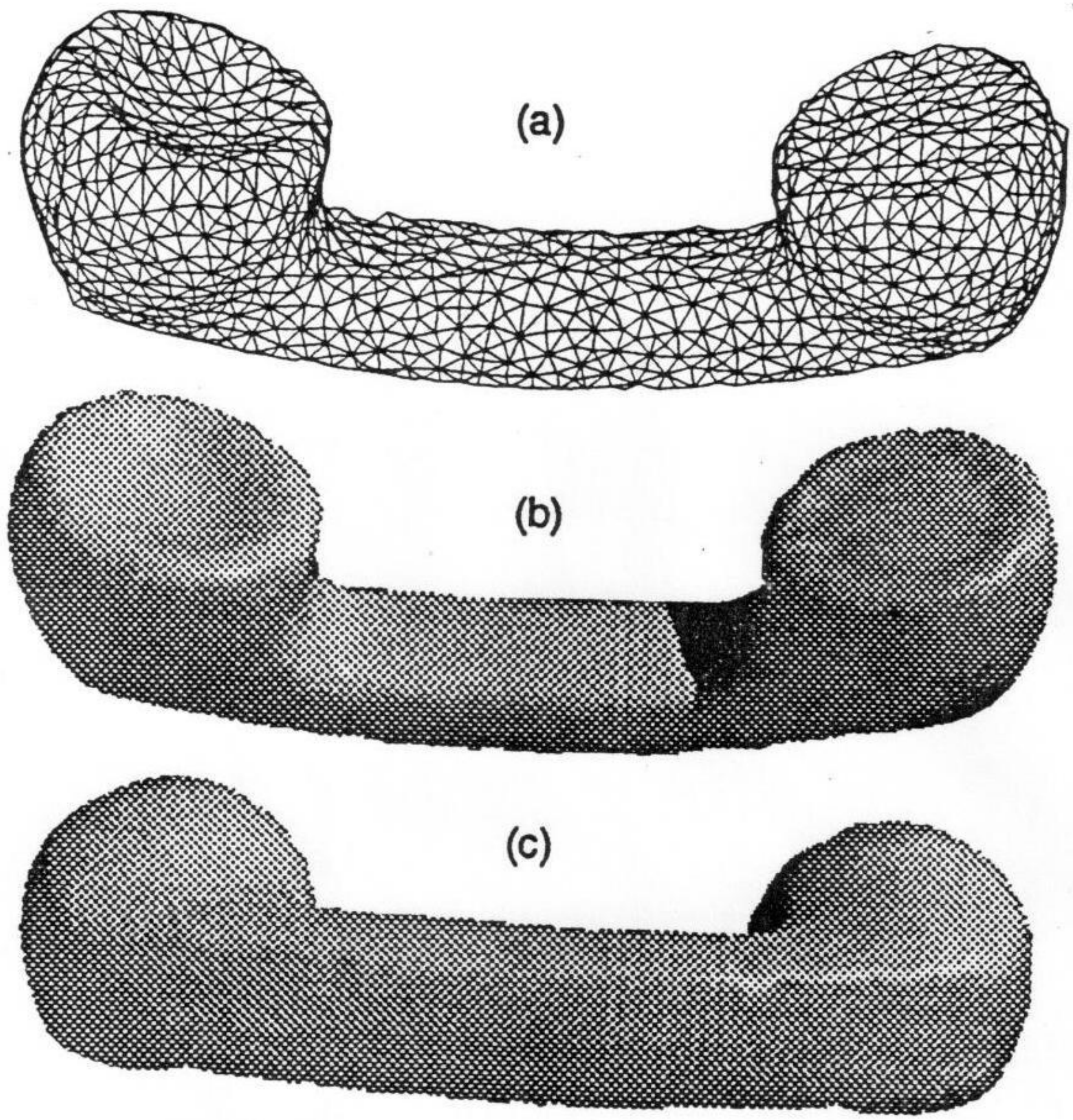


Figure 2: A mesh model of a telephone handset.

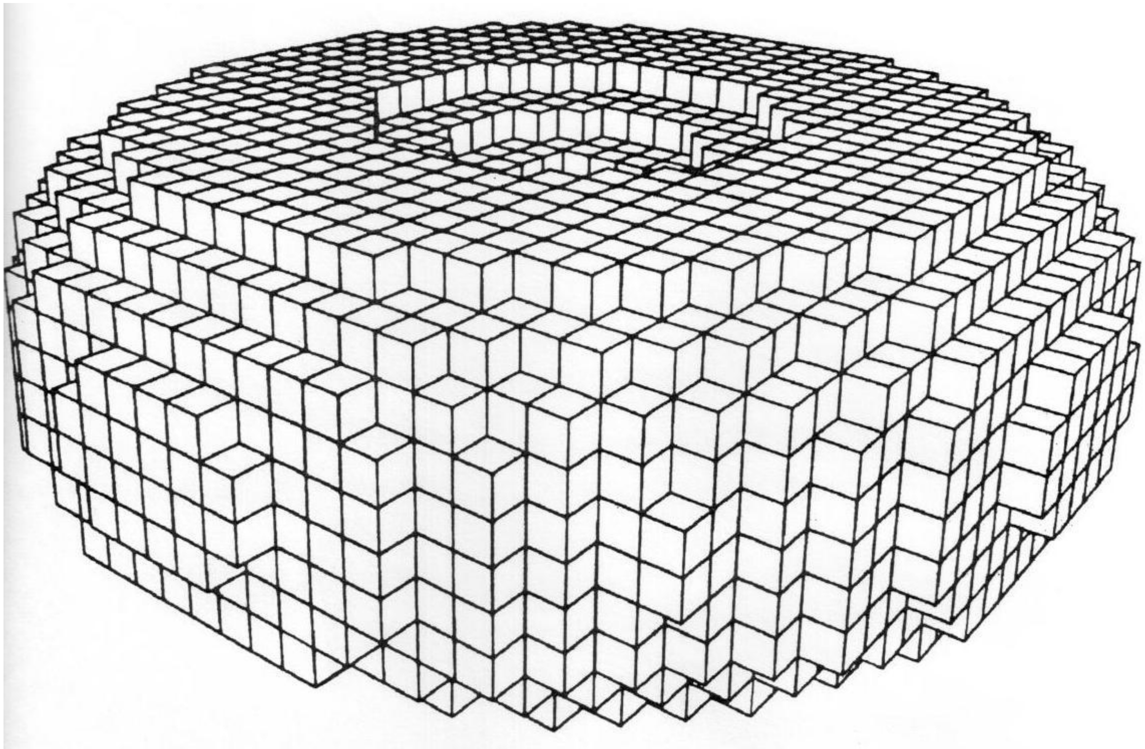

Figure 3: A voxel grid model of a donut.

(2) **Voxel grid**. A voxel grid represents an object using a 3D grid of cells. Each cell is called a volume element (voxel) and models a small cube of space.

- Voxels can each be given a different color or texture.
- The triangles are all composed of simple planes and so can be easily rotated, translated, and rendered using ray tracing.
- A model is easily fit to range image data by marking a voxel as full if a range data point lies within it, and empty it no range data points lie within it.
- Multiple views (images) are easy to combine. A new view of additional 3D data can simply be projected into the voxel space.
- Voxel grids are okay for visualization but not quite as good as meshes because they tend to yield a rasterization jagged effect unless the density is very high. They can answer some geometric questions such as volume, but do not provide explicit surface primitives. Although they can be more compact than original range data, they are not nearly as compact as some other models.

Figure 3 shows an example voxel grid.

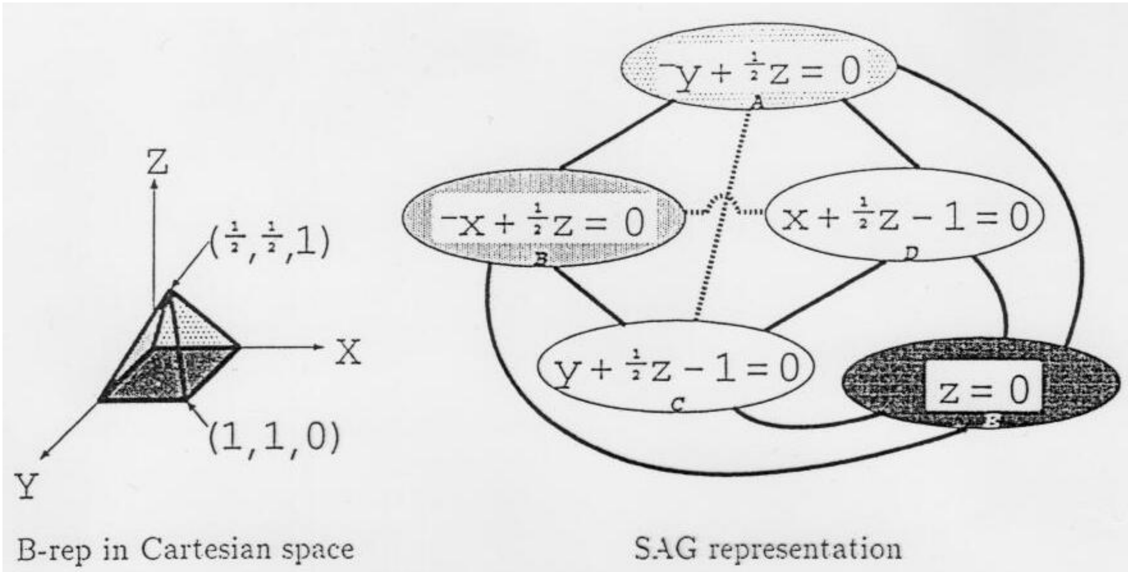


Figure 4: An example boundary representation (brep) model. The surface adjacency graph (SAG) represents the topology of the surfaces (how they are connected), and the nodes represent the geometry (the equations of the surfaces).

(3) **Boundary representation (b-rep).** A boundary model represents an object using a set of surface equations and a topology. The surface equations describe planes of curved surfaces. The topology describes the connections of the surfaces.

- Surfaces can each be given a different color or texture, or a functional pattern that varies across the surface.
- Surface equations can be easily rotated, translated, and rendered using ray tracing, but the ray tracing typically devolves curved surfaces into planar elements for simpler computations and consistency.
- A model is difficult to fit to range image data because the data must be segmented into surfaces. Finding the topology is equivalent to finding edges which is also not trivial.
- Multiple views (images) are difficult to combine. A new view of another angle of the object must be matched against the existing topology of the model.
- B-reps are poor for visualization because they tend to have little detail. They are far more compact than meshes and voxel grids. They tend to be good for some geometric questions because surface, edge and volume computations are all simple. They are also useful for questions involving surface and part relationships, e.g. topology.

Figure 4 shows an example b-rep model.

Figure 5 highlights some of the similarities and differences between mesh and b-rep models. While the b-rep car model looks poor, it can more easily be used to answer questions about pieces of the car.

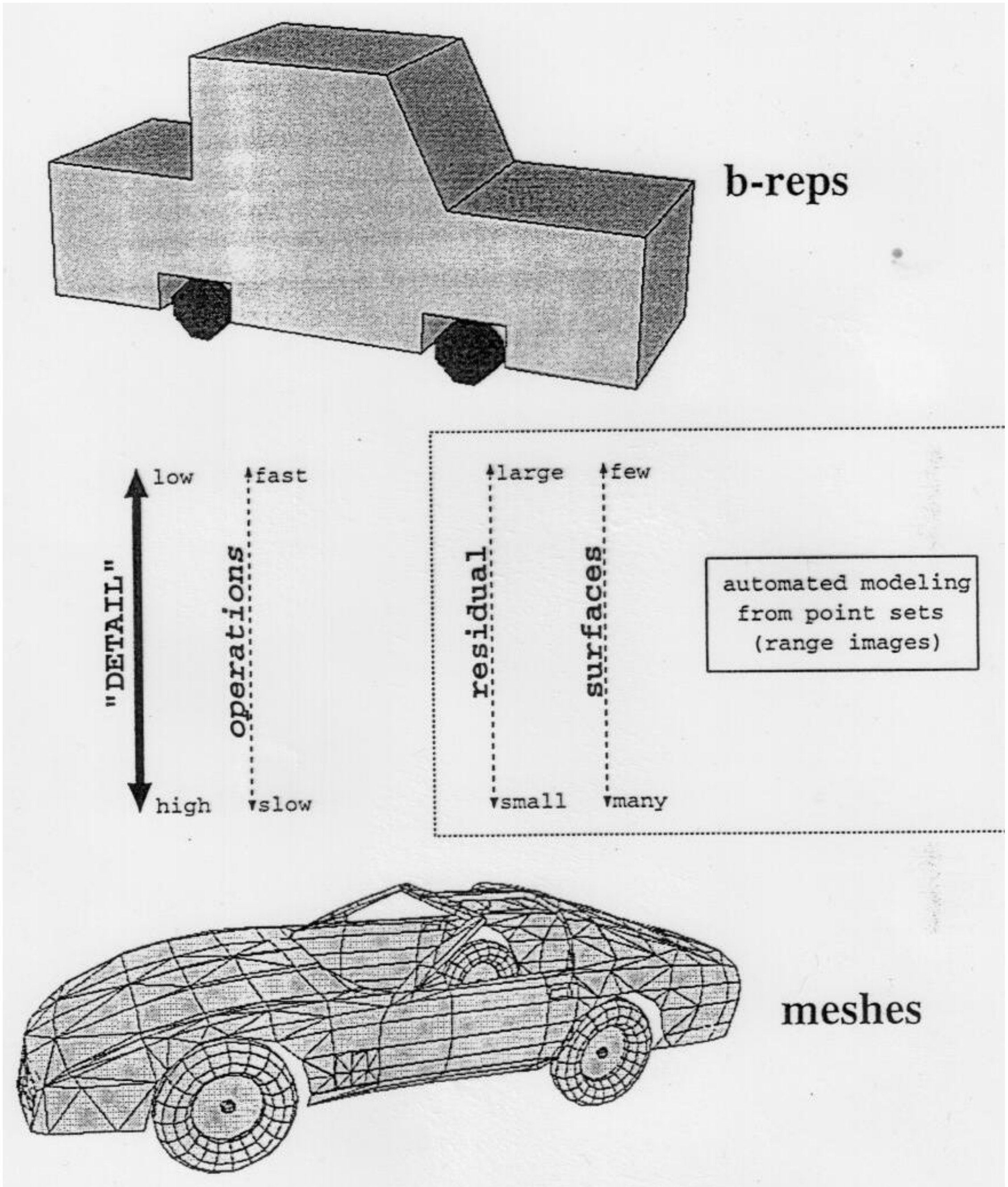


Figure 5: Continuum of b-rep to mesh modeling.

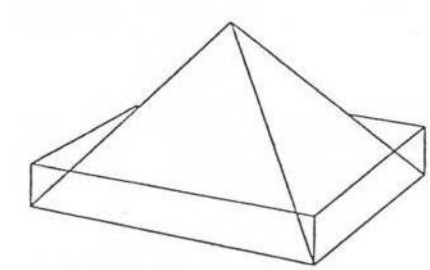

Figure 6: Invalid b-rep model.

Figure 6 shows an example of an invalid b-rep. Invalidity means that although the surface primitives are valid and the topology is valid, their combination describes an object that cannot be physically realized.

Many mesh models are invalid. This is not a problem if the model will only be used for visualization. B-reps however are generally used for answering geometric/physical questions, so that model validity is critical.

Rules for b-rep validity are well established. For a polyhedron b-rep to be valid, it must satisfy the following criteria:

1. Each face must have at least 3 edges.
2. Each edge must have precisely 2 vertices.
3. Each edge must belong to precisely two faces.
4. Each vertex in a face must belong to precisely two of the face’s edges.
5. Each vertex must be unique.
6. All vertices belonging to a face must solve the same plane equation.
7. Edges must be either disjoint or intersect at a common vertex.
8. Faces must be either disjoint or intersect at a common edge or vertex.

Meshes can be manipulated or constructed to force validity. This is necessary for example for models that will have physics applied to them, such as in video games.

The next three modeling types are all known as “parts” models, because they use primitives to represent different parts of objects.

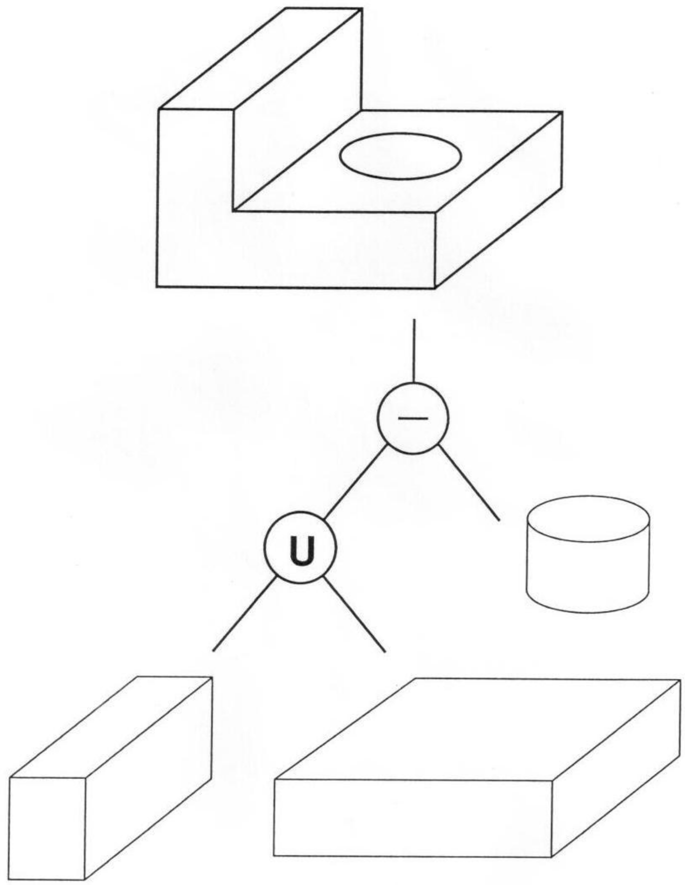

Figure 7:  Constructive solid geometry model.

(4) **Constructive solid geometry.**  A CSG model uses a set of primitives and combinatorial rules to represent an object.  Figure 7 shows an example.  This modeling scheme can be useful when working with objects composed of similar parts, but has not proven as generally practical as a mesh model.

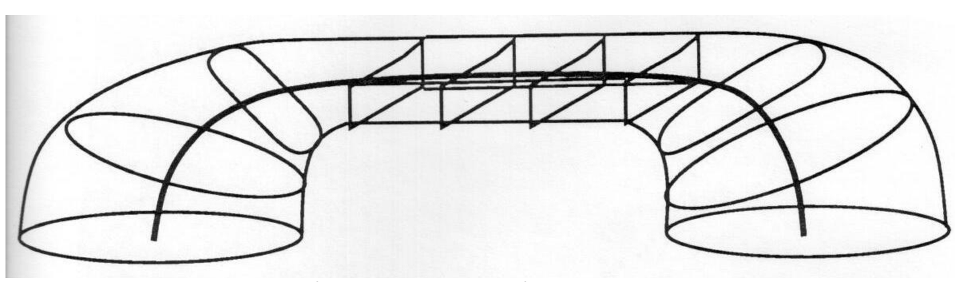
Figure 8:  Spline model.

(5) **Spline model.**  A spline model uses a contour through 3D space along with a cross-sectional function to describe a volume.  It provides a very compact way to represent complex curved surfaces.  However, there is no easy way to render it so it is typically converted to a mesh for visualization.  There is also no easy way to represent holes, topology, or combinations of parts.

(6) **Superquadrics and Hyperquadrics.** A superquadric or hyperquadric model uses a set of volumetric primitives of tunable shape. Each primitive represents a different part of an object. The shape of each primitive can be controlled by adjusting parameters that enclose the volume. At the course website, a document giving details on these models provides details.

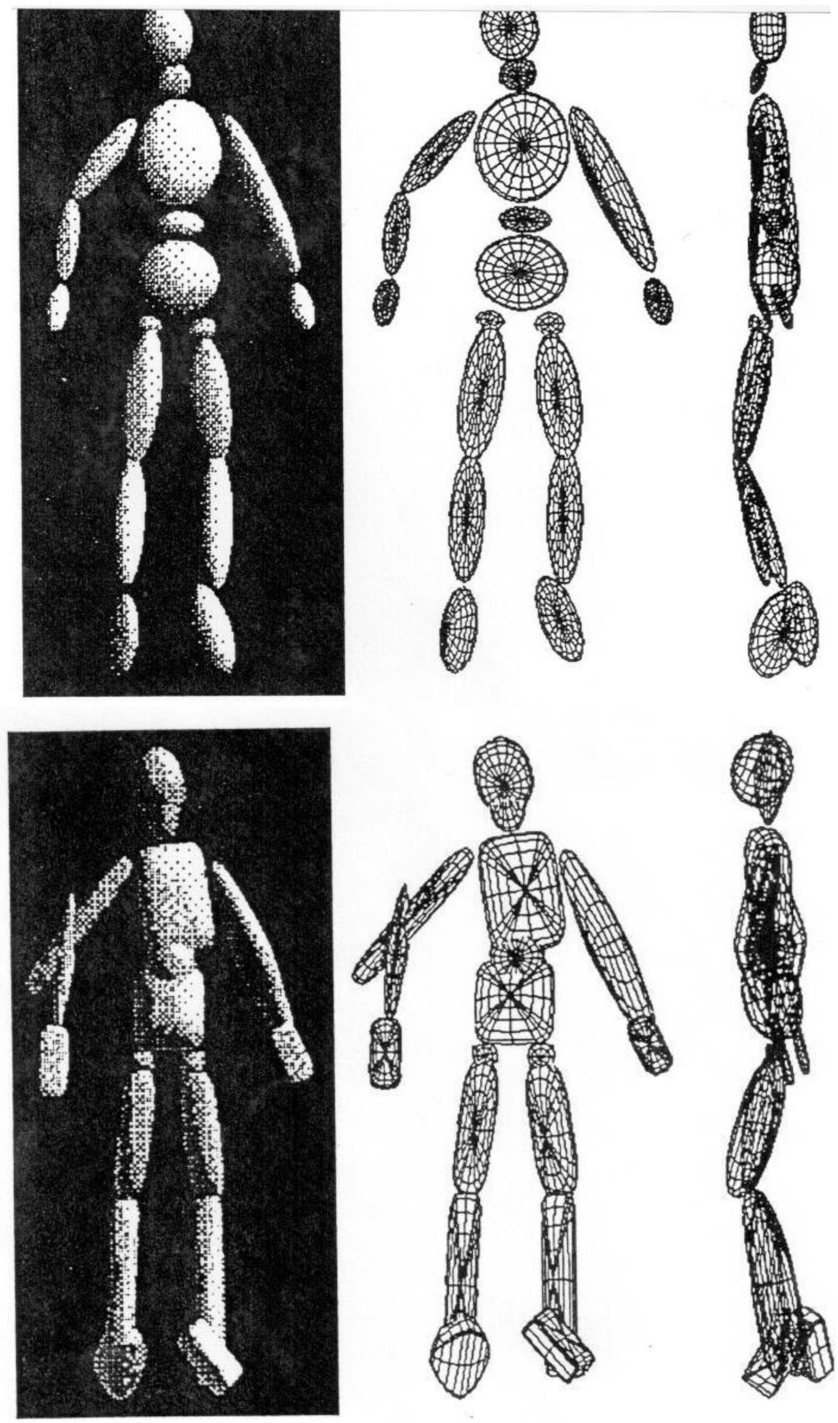

Figure 9: Superquadric (top) and hyperquadric (bottom) model.