Computer Graphics Course

Three-Dimensional Modeling

Lecture 13

"Representations of Solids"

References

D. Hearn, M.P. Baker "Computer Graphics", Prentice Hall, Second Edition, 1994, pp.355-362.

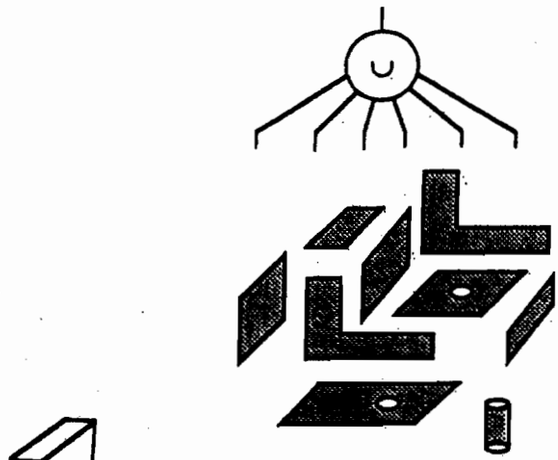
J.D.Foley et al. "Computer graphics: Principles and Practice", Addison-Wesley, Second Edition, 1992, pp.533-562.

Shape Modeling and Computer Graphics with Real Functions
http://www.u-aizu.ac.jp/public/www/labs/sw-sm/FrepWWW/F-rep.html

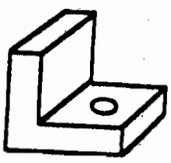
Representations of Solids

- Primitive instancing
- Boundary representation (B-rep)
- Constructive Solid Geometry (CSG)
- Spatial-partitioning representations
- Sweep representations
- Function representation (F-rep)

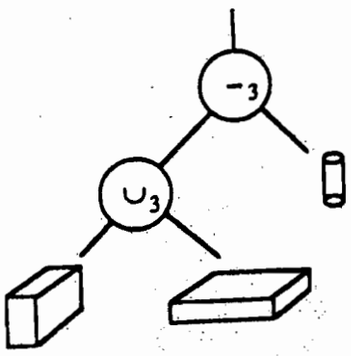
Styles of representation



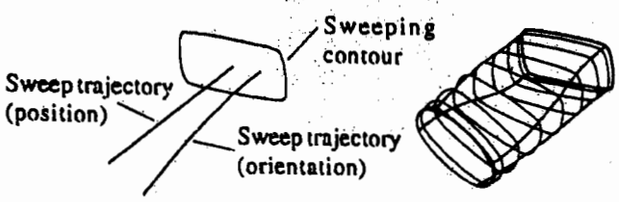
Boundary representation
B-rep



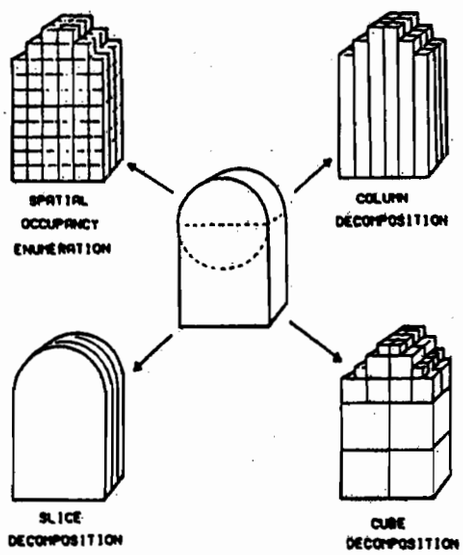
SOLID



Constructive Solid Geometry
CSG



Sweeping



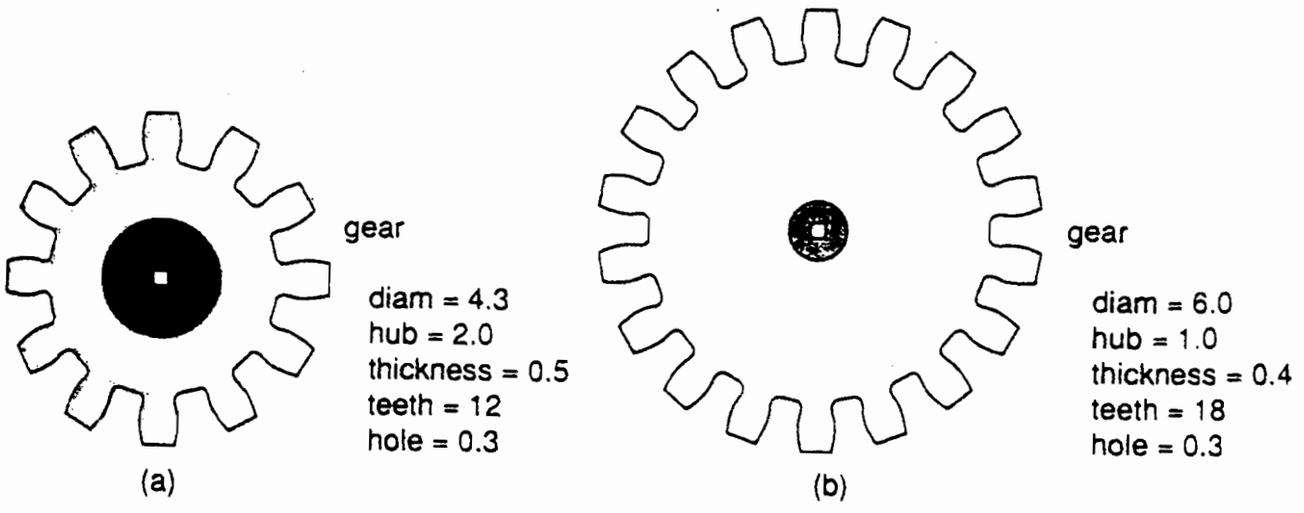
Spatial decomposition

Primitive instancing

The modeling system defines a set of primitive 3D solid shapes for the specific application area:

$$\textbf{primitive}_i\ (a_1, a_2, ..., a_k)$$

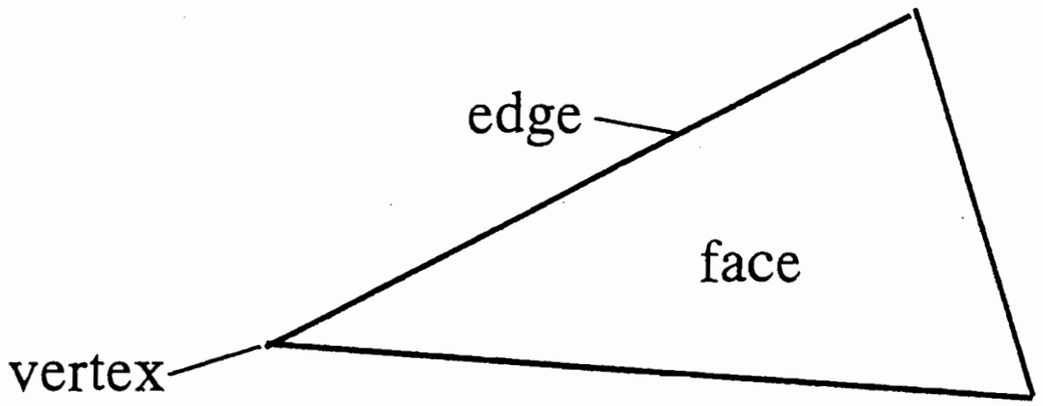
- primitive + parameters define a family of parts;
- may include complex objects (gears, bolts, etc.);
- no operations to form a new more complex object;
- only one way to create a new kind of object - to write the code that defines it;
- programs to draw or to calculate mass properties must be written individually for every primitive;



Two gears defined by primitive instancing.

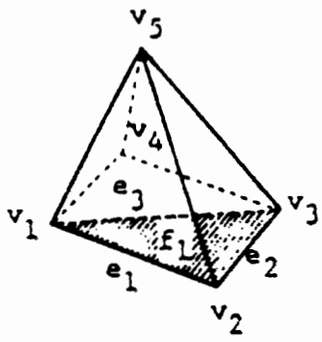
Boundary representation (B-rep)

- B-rep describes a solid in terms of its surface boundaries: *vertices*, *edges*, and *faces*.

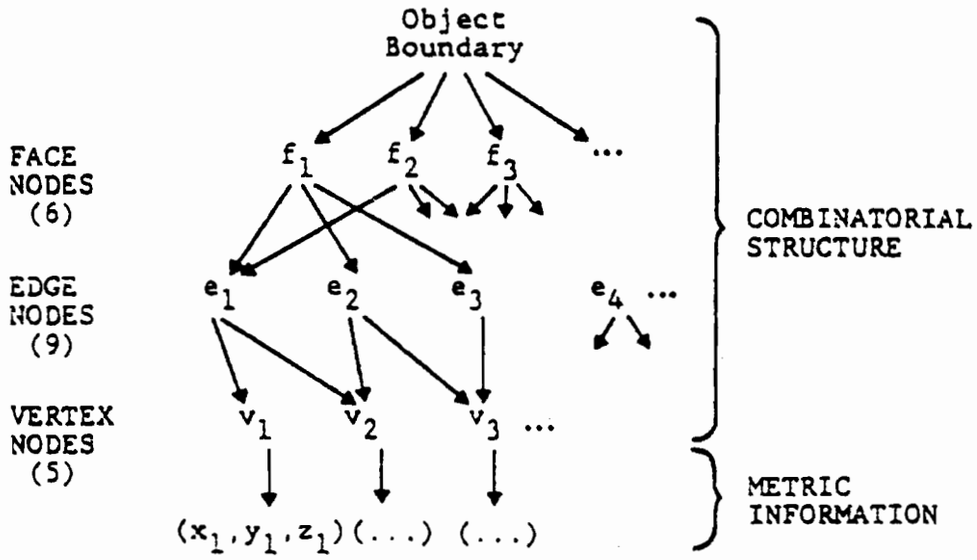


- Types: polygonal and curved faces;
- Curved faces can be approximated by polygons or represented by parametric (implicit) surfaces;
- Many B-rep systems support only solids whose boundaries are *2-manifolds*.

Boundary representation
Example



(a)



(b)

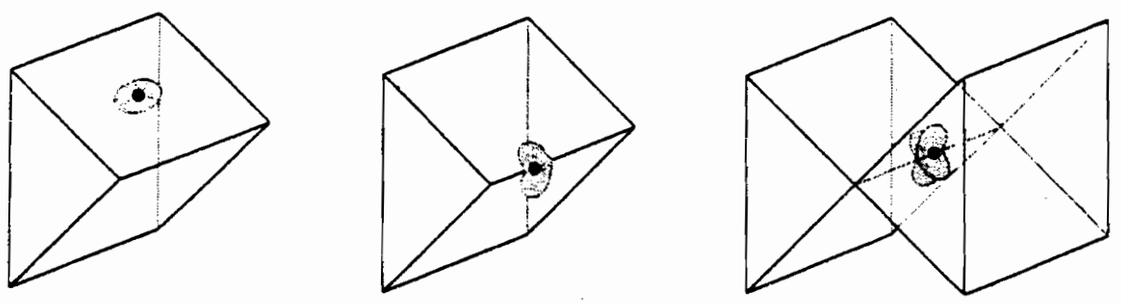
A boundary representation for a rectangular pyramid.

Boundary representation
2-manifolds

• Every point on a 2-manifold has a neighborhood of points around it that is topologically the same as a disk in the plane.

• There is a continuous one-to-one correspondence between the neighborhood and the disk (Figures a and b).

• Example: if more than two faces share an edge (Figure c), any neighborhood contains points from each of those faces. Thus, the surface is not a 2-manifold.



On a 2-manifold, each point, shown as a black dot, has a neighborhood of surrounding points that is a topological disk, shown in gray in (a) and (b). (c) If an object is not a 2-manifold, then it has points that do not have a neighborhood that is a topological disk.

Boundary representation

Polyhedra and Euler's Formula

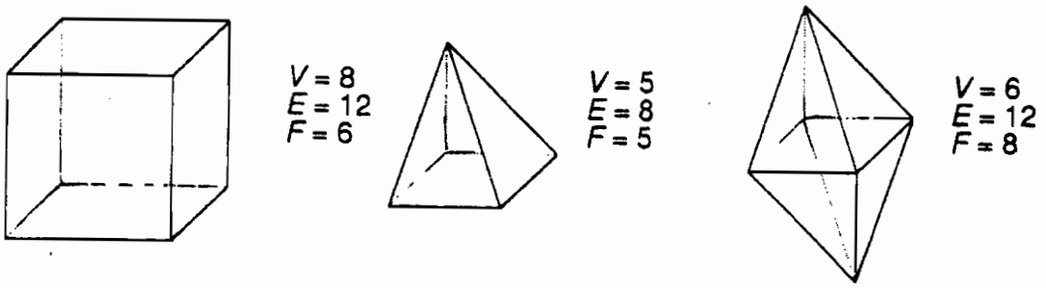
- A 3D *polyhedron* is a solid that is bounded by a set of polygons:
  - each edge connects two vertices and is shared by exactly two faces;
  - at least three edges meet at each vertex;
  - faces do not interpenetrate.

- A *simple polyhedron* can be deformed into a sphere (no holes).

- The B-rep of a simple polyhedron satisfies Euler's formula:

$$V - E + F = 2$$

where V is the number of vertices, E is the number of edges, F is the number of faces.



Some simple polyhedra with their *V*, *E*, and *F* values.

Boundary representation

Generalized Euler's Formula

• The B-rep of 2-manifolds that have faces with holes satisfies the generalized Euler's formula:

$$V - E + F - H = 2\,(C - G)$$

where V is the number of vertices,

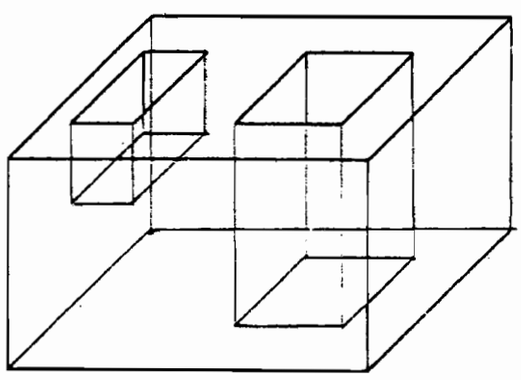
E is the number of edges,

F is the number of faces,

H is the number of holes in the faces,

C is the number of separate components (parts),

G is the *genus* (for a torus G = 1).



$V - E + F - H = 2(C - G)$

24 36 15 3 1 1

A polyhedron with two holes in its top face and one hole in its bottom face.

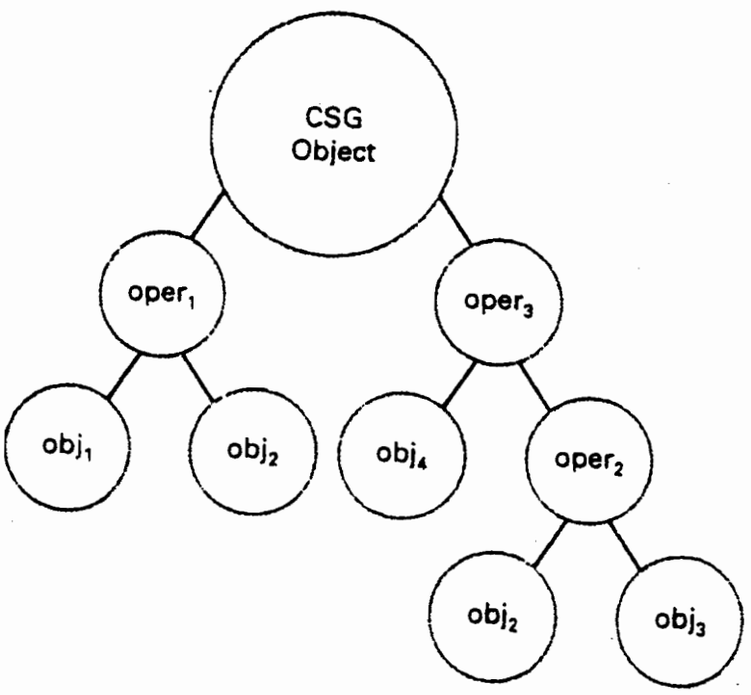
Constructive Solid Geometry (CSG)

• CSG is based on a set of 3D solid primitives and *regularized* set-theoretic operations.

• Traditional primitives: block, cylinder, cone, sphere, torus.

• Operations: union, intersection, difference + translation and rotation.

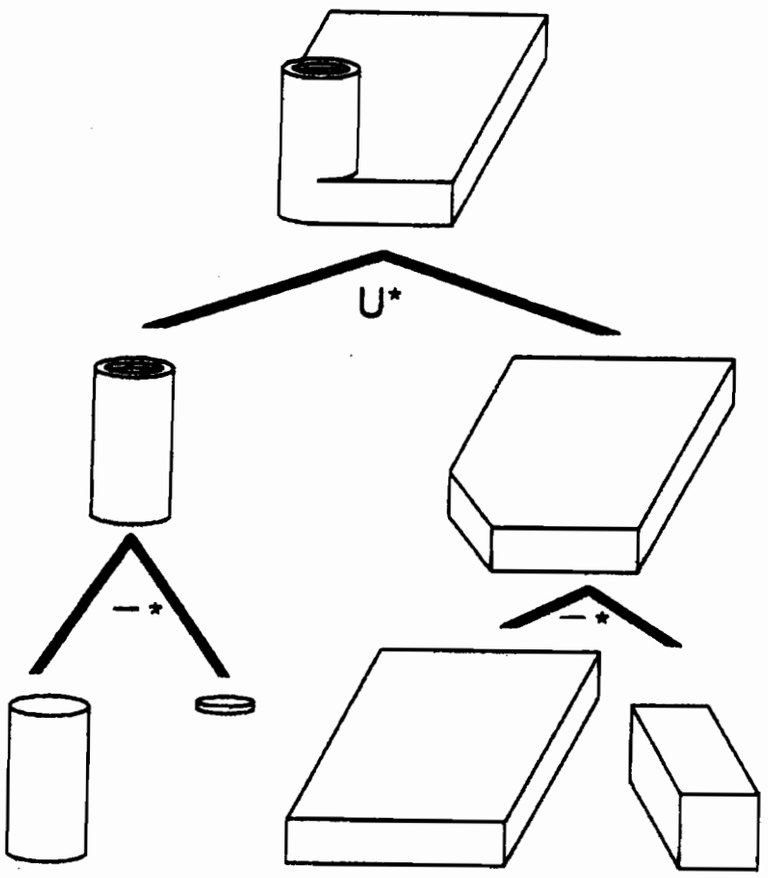
• A complex solid is represented with a binary tree usually called *CSG tree*.



A CSG tree representation for an object.

Constructive Solid Geometry

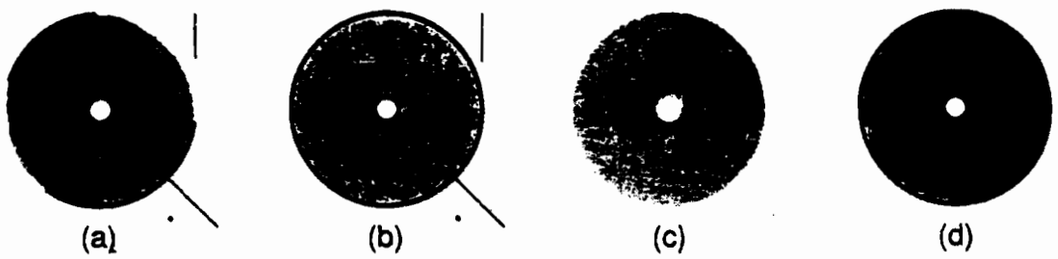
Example of the CSG tree



An object defined by CSG and its tree.

Constructive Solid Geometry
Regular solids

- Consider a solid and different types of its points:
  - *boundary points* are points whose distance from the object and the object's complement is zero;
  - *interior points* belong to the solid and are not boundary points;
  - *exterior points* do not belong to the solid.
- A *closed set* contains all its boundary points, an *open set* contains no any one boundary point.
- The union of a set with its boundary is a *closure* of a set.
- The *regularization* of a set is defined as the closure of the set's interior points.

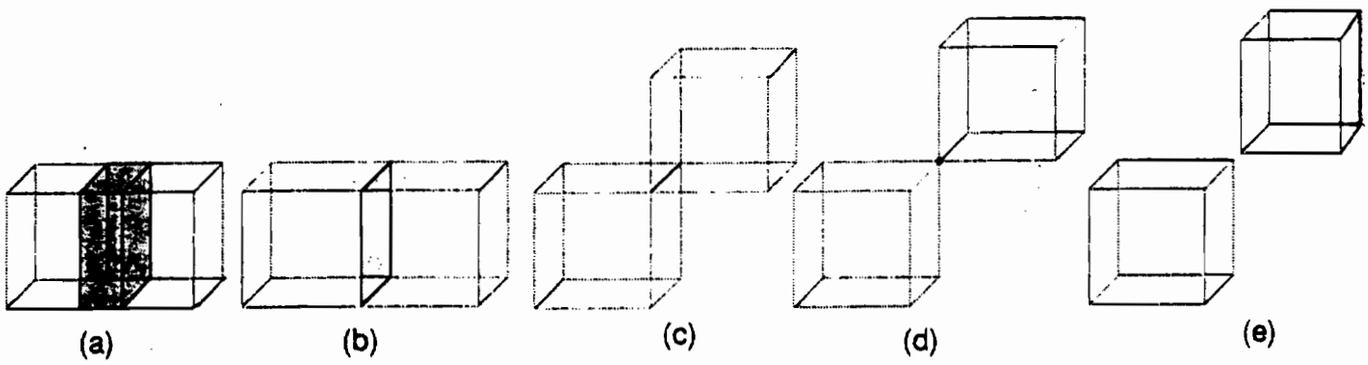


Regularizing an object. (a) The object is defined by interior points, shown in light gray, and boundary points. Boundary points that are part of the object are shown in black; the rest of the boundary points are shown in dark gray. The object has dangling and unattached points and lines, and there is a boundary point in the interior that is not part of the object. (b) Closure of the object. All boundary points are part of the object. The boundary point embedded in the interior of (a) is now part of the interior. (c) Interior of the object. Dangling and unattached points and lines have been eliminated. (d) Regularization of the object is the closure of its interior.

Constructive Solid Geometry

Regularized set-theoretic operations

• Applying an ordinary set operation to two solid objects does not always results in a solid object



The ordinary Boolean intersection of two cubes may produce (a) a solid, (b) a plane, (c) a line, (d) a point, or (e) the null set.

• Regularized set operations are defined such that operations on solids always result in solids:

$$A\ op^*\ B = closure\ (\ interior\ (\ A\ op\ B\ )\ ),$$

where **op** is one of the ordinary set operations.

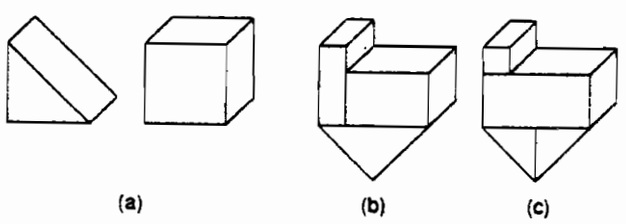
Spatial-partitioning representations

In spatial-partitioning representations, a solid is decomposed into a collection of adjoining, nonintersecting solids that are more primitive than the original solid. Primitives may vary in type, size, position, parametrization, and orientation.

Forms of spatial-partitioning representations:
- Cell decomposition
- Spatial-occupancy enumeration
- Octrees
- Binary space-partitioning trees

Cell decomposition

- Parametrized set of primitive cells that are often curved.
- Constructing complex objects by "gluing" primitive cells together.
- Restrictions on "glue" operation often require that two cells share a single point, edge, or face.
- Representation is not unique:



The cells shown in (a) may be transformed to construct the same object shown in (b) and (c) in different ways.

- Cell decomposition is used in 3D finite element methods for the numerical solution of differential equations.

Spatial-partitioning representations

Spatial-occupancy enumeration

• Special case of cell decomposition with identical cells arranged in a fixed, regular grid. The cells are often called *voxels*.

• The most common cell type is the cube, and the representation of space as a regular array of cubes is called a *cuberrile*.

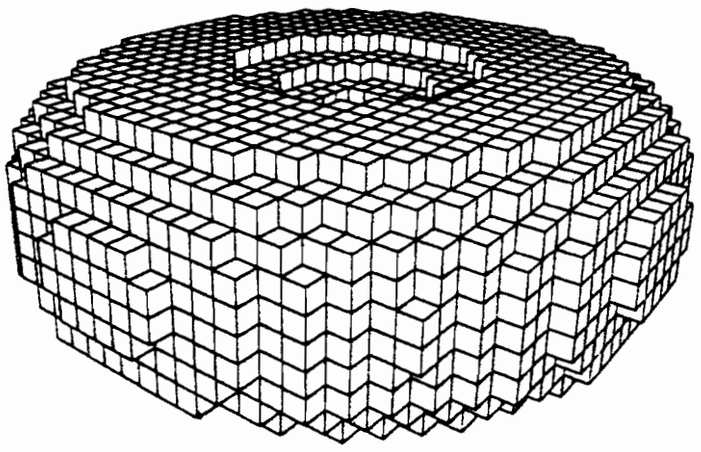
• For every cell only it presence or absence in the grid is defined. A cell is presented in the grid if it is *occupied* by the object.

• Set-theoretic operations can be easily implemented.

• Disadvantages:

- approximate model, no concept of "partial occupancy",

- memory consuming (up to n^3 cells).

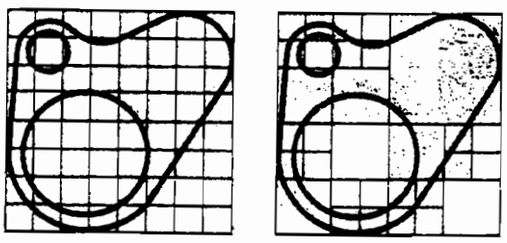


Torus represented by spatial-occupancy enumeration.

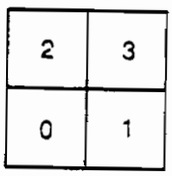
Spatial-partitioning representations
Quadtrees and octrees

• *Octrees* are a hierarchical variant of spatial occupancy enumeration. Octrees are derived from *quadtrees*, a 2D image representation.

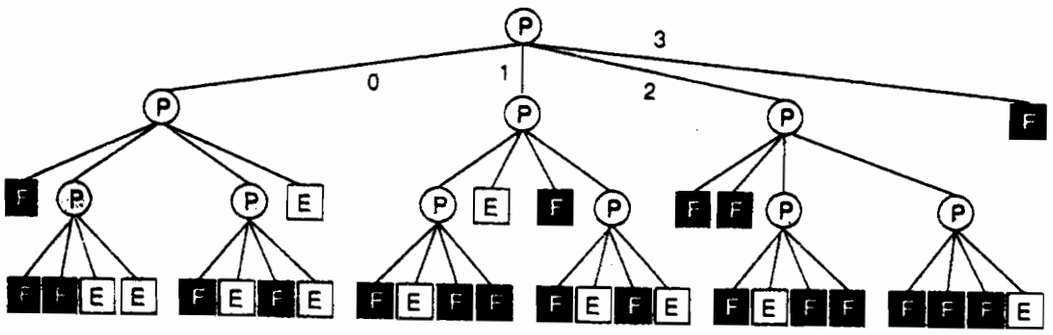
• A quadtree is derived by recurrently subdividing a 2D plane in both directions to form quadrants.



An object represented using (a) spatial-occupancy enumeration (b) a quadtree.



Quadrant numbering

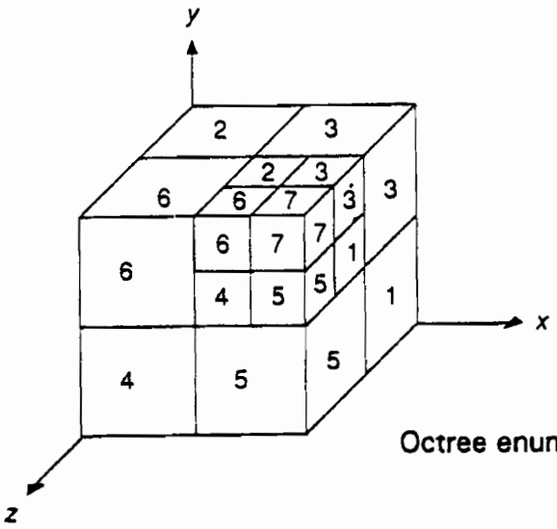


Quadtree data structure for the object

F = full, P = partially full, E = empty.

Spatia. partitioning representations

Octrees



Octree enumeration. Octant 0 is not visible.

• An octree scheme divides regions of 3D space (usually cubes) into *octants* and stores 8 data elements in each node of the tree.

• Procedure for generating octree: each octant is tested, and octant subdivisions continue until octants are homogeneous (full or empty).

• Number of nodes in an octree is proportional to the object's surface.

• Operations: Boolean, rotation by 90 degrees, scaling by powers of 2, translations.

• Problem of aliasing under general transformations.

Spatial-partitioning representations

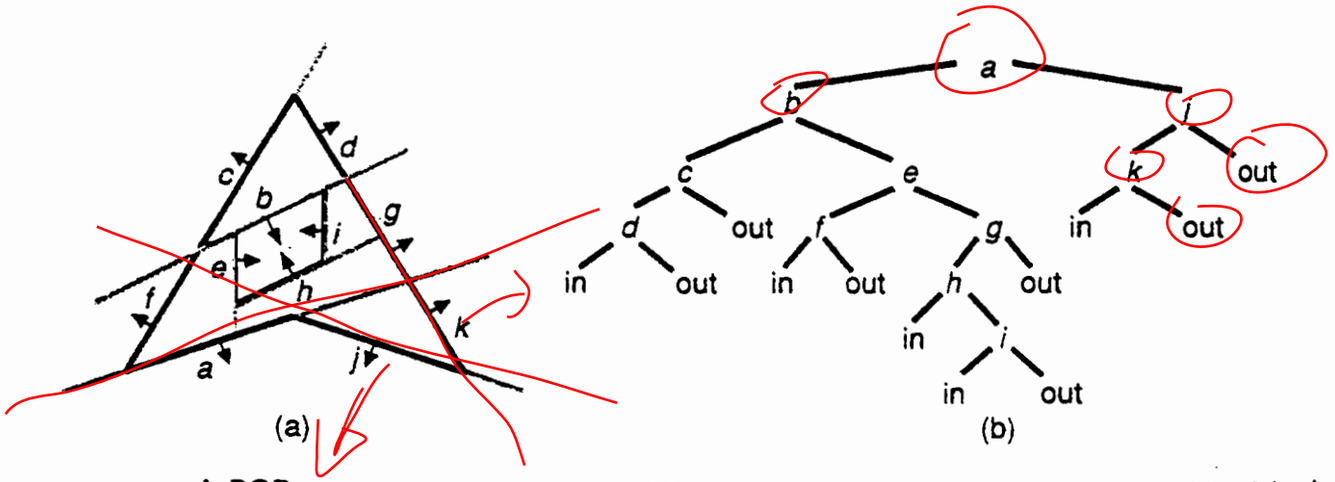
Binary space-partitioning (BSP) trees

• *BSP trees* recursively divide space into pairs of subspaces, each separated by a plane of arbitrary orientation and position.

• Each internal node of the BSP tree is associated with a plane and has two child pointers, one for each side of the plane.

• If the halfspace on on a side of the plane is homogeneous, then its child is a leaf and represents a region either inside ("in") or outside ("out") the object.

• A BSP tree can represent an arbitrary concave polyhedral solid with holes as a union of convex "in" regions.



A BSP tree representation in 2D. (a) A concave polygon bounded by black ines. Lines defining the half-spaces are dark gray, and "in" cells are light gray. (b) The 3SP tree.

Sweep representations

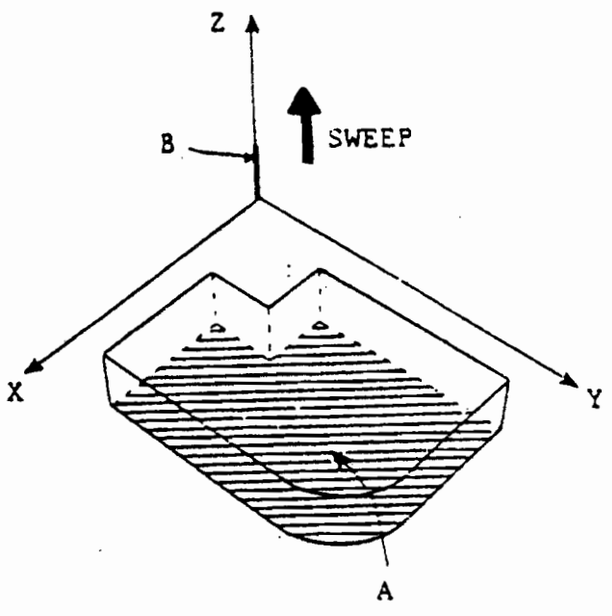
• A set of all points visited by an object moving along a trajectory is a new object, called a *sweep*.

• *Translational sweeping (extrusion)*: 2D area moves along a line normal to the plane of the area.

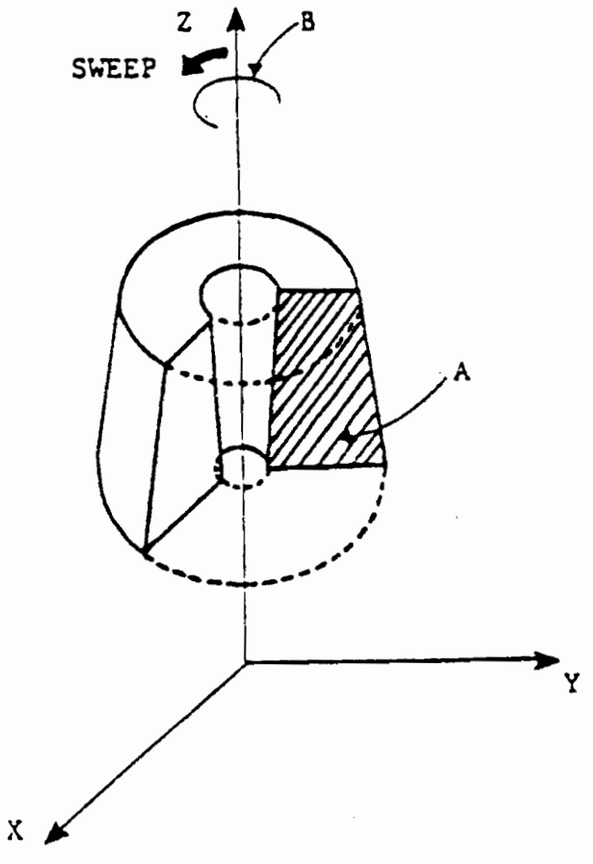
• *Rotational sweeping* is defined by rotating an area about an axis.

• Sweeps whose generating area changes in size, shape or orientation and follows an arbitrary curved trajectory are called *generalized cylinders*.

• Problems: sweeping by a moving solid, self-intersections, CSG operations on sweeps.



Translational sweeping.



Rotational sweeping.

Function representation (F-rep)

• A geometric object is defined by a single real function of several variables $f(x_1, x_2, ..., x_n) \geq 0$.

• A function can be defined analytically, with an evaluation algorithm, or with tabulated values and an appropriate interpolation procedure.

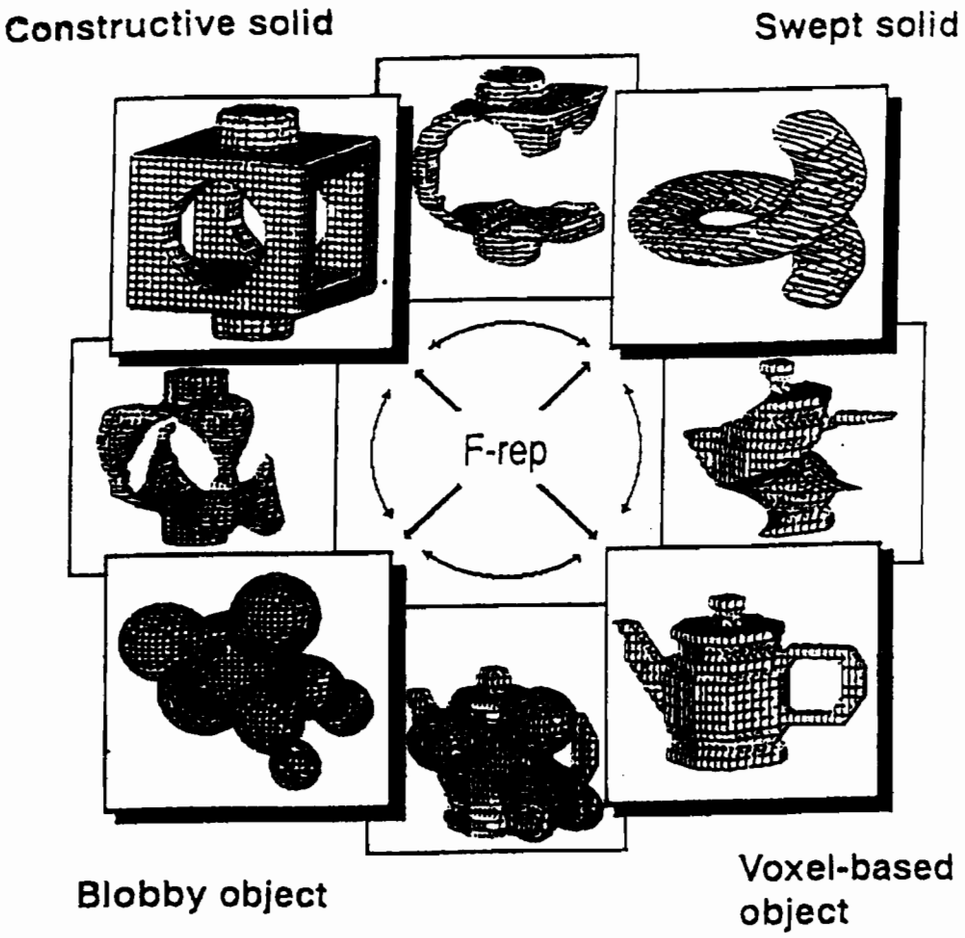
• F-rep is closed under the following operations:

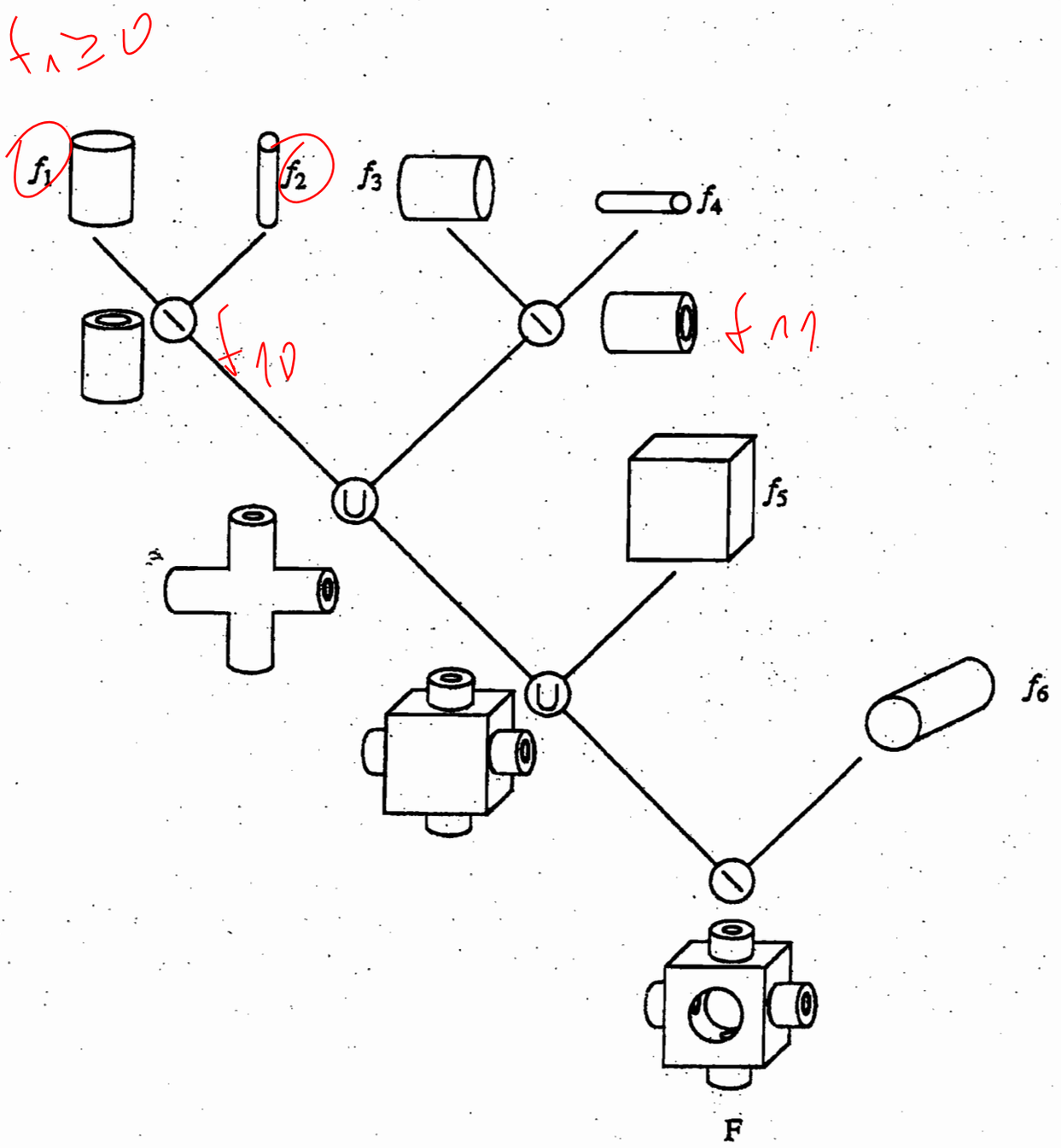
- set-theoretic operations with R-functions;
- blending set-theoretic operations;
- offsetting;
- Cartesian product;
- bijective mapping;
- projection;
- deformations;
- metamorphosis.

• An abstraction level higher than those of other known representations is provided. Combinations of the following modeling styles are supported: CSG, sweeping, blobby objects, volumetric objects. Dimension independent representation.

• Not closely related with B-rep.

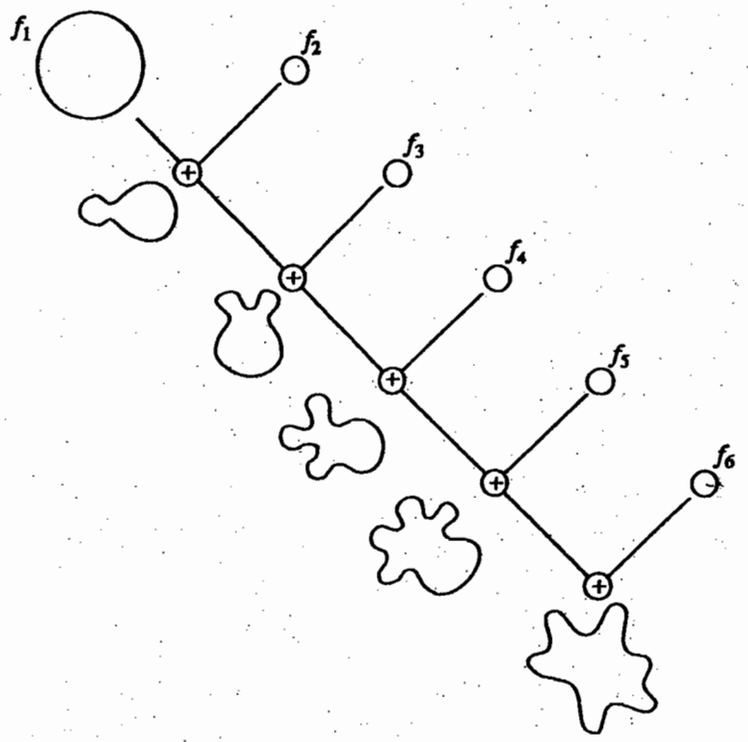
• Time-consuming and requires parallel/distributed processing or hardware solutions.


Constructive solid
Swept solid
F-rep
Blobby object
Voxel-based object



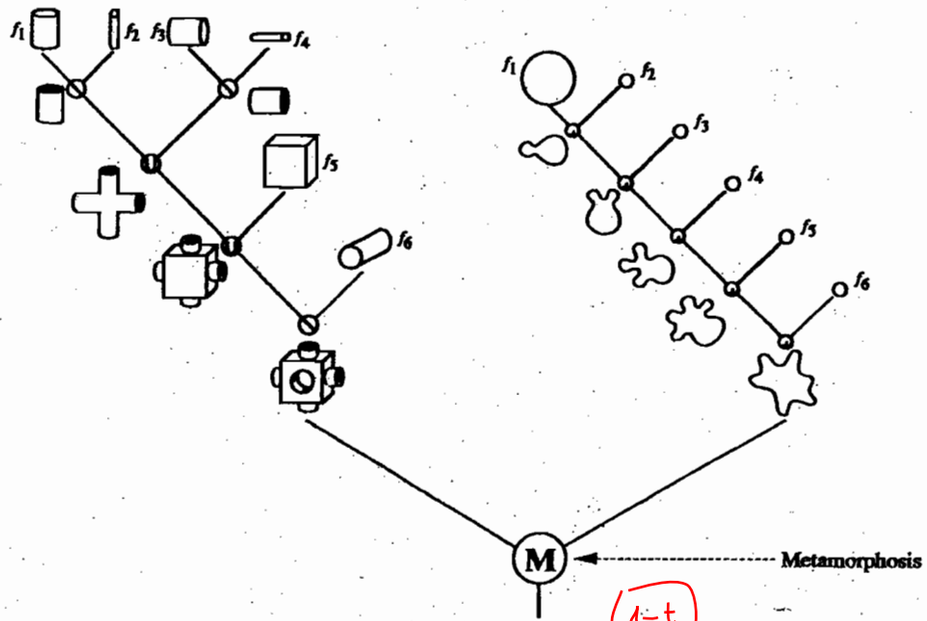
$$F = ((f_1 \backslash f_2) \cup (f_3 \backslash f_4)) \cup f_5) \backslash f_6$$

$\backslash, \cup$: R function （操作）



$$f_i = b_i e^{-a_i r_i^2}$$

$$F_2 = ((((f_1 + f_2) + f_3) + f_4) + f_5) + f_6 = \sum_{i=1}^{6} f_i$$



$$M(t) = F_1 \cdot t + F_2 (1 + t) \geq 0$$