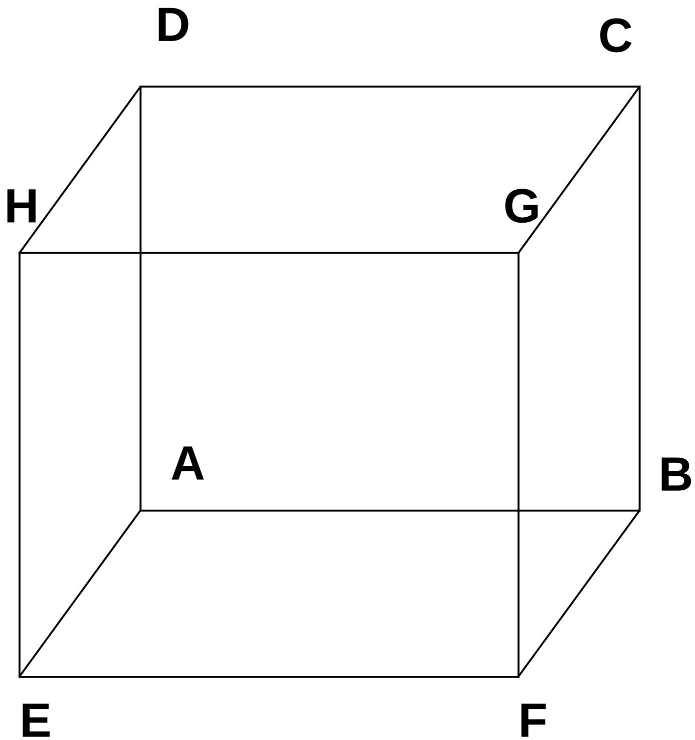
Solid Modeling; Constructive Solid geometry (CSG)

Various methods:

- **Regularized Boolean Set operations**
- **Sweep Representation**
- **Octrees**
- **CSG**
- **B-reps**
- **Primitive Instancing**
- **Fractal Dimension**

Solid Representation

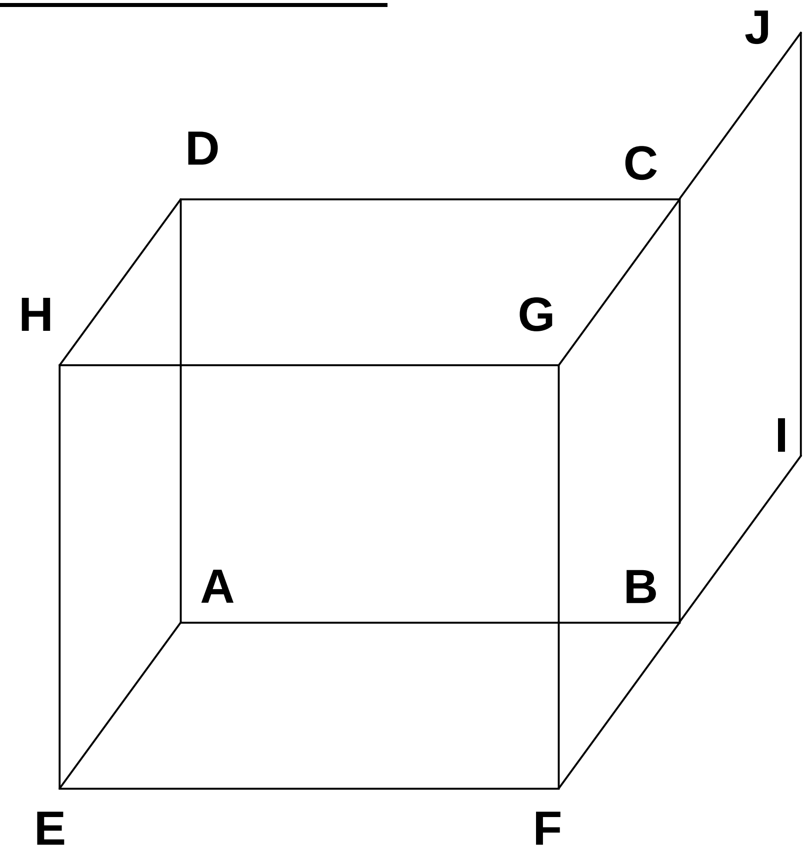


Vertices:

A	(0, 0, 0)
B	(1, 0, 0)
C	(1, 1, 0)
D	(0, 1, 0)
E	(0, 0, 1)
F	(1, 0, 1)
G	(1, 1, 1)
H	(0, 1, 1)

Lines:

AB
BC
CD
DA
EF
FG
GH
HE
AE
BF
CG
DH



Additional Vertices:

I	(1, 0, -1)
J	(1, 1, -1)

Additional Lines:

BI
IJ
JC

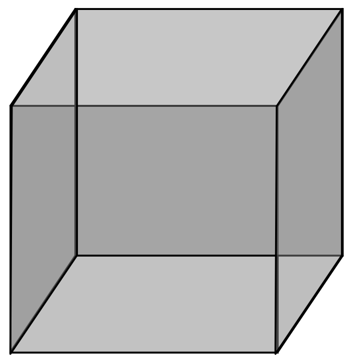
Solid is bound by surfaces. So need to also define the polygons of vertices, which form the solid. It must also be a valid representation.

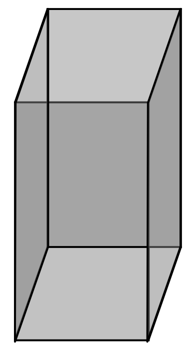
Regularized Boolean Set Operations

Operators (*op*):
Union: $\bigcup$
Intersection: $\bigcap$
Difference: $-$

Boolean intersections of, say cubes, may produce solids, planes, lines, points and null object. Two examples:

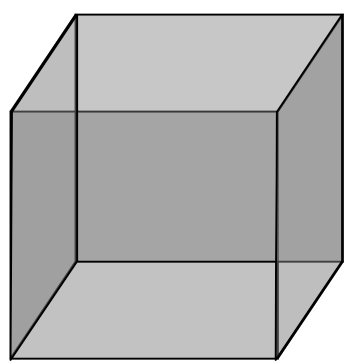
A

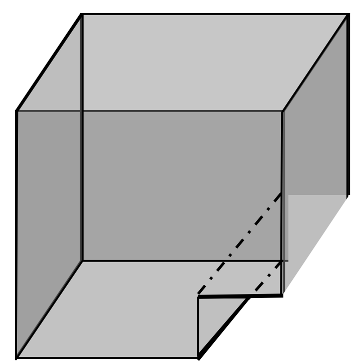




$A \bigcap B$

B

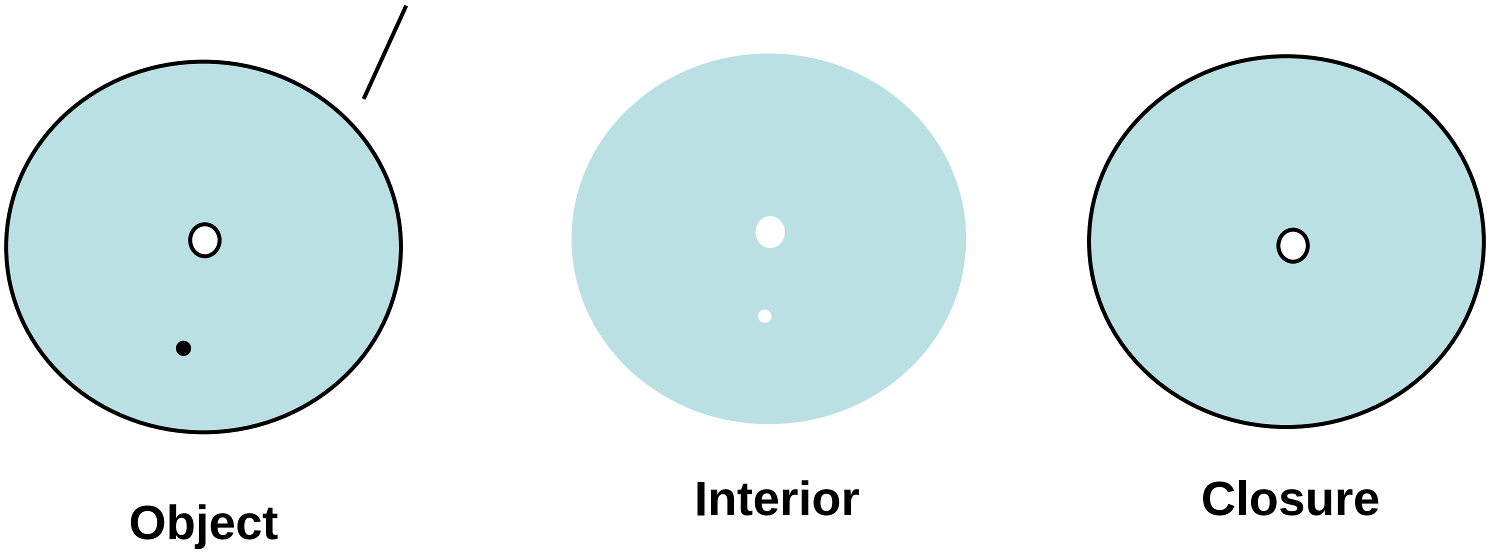




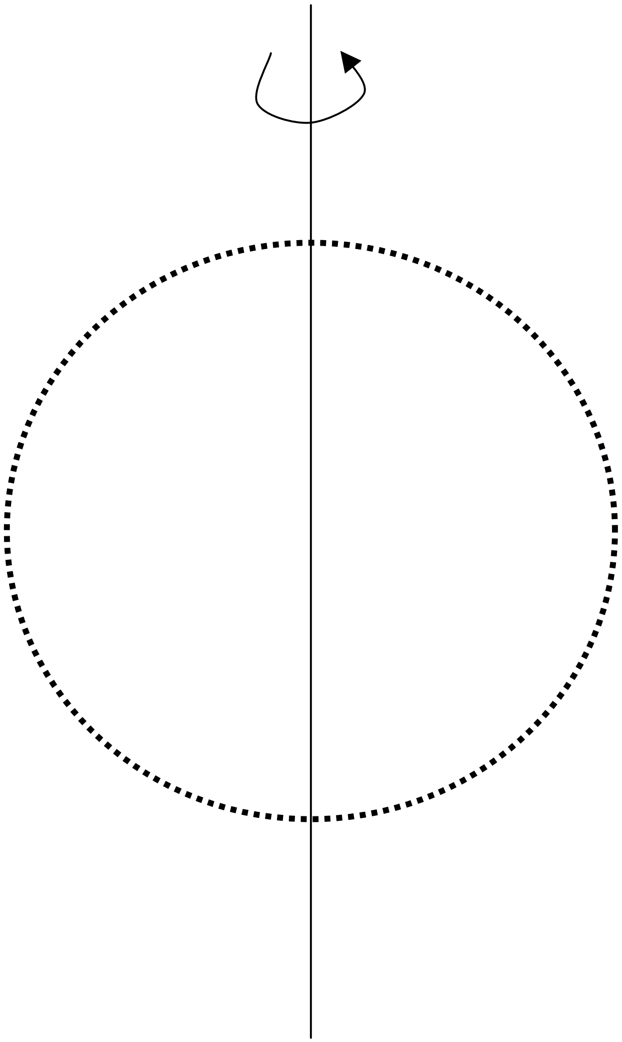
A - B

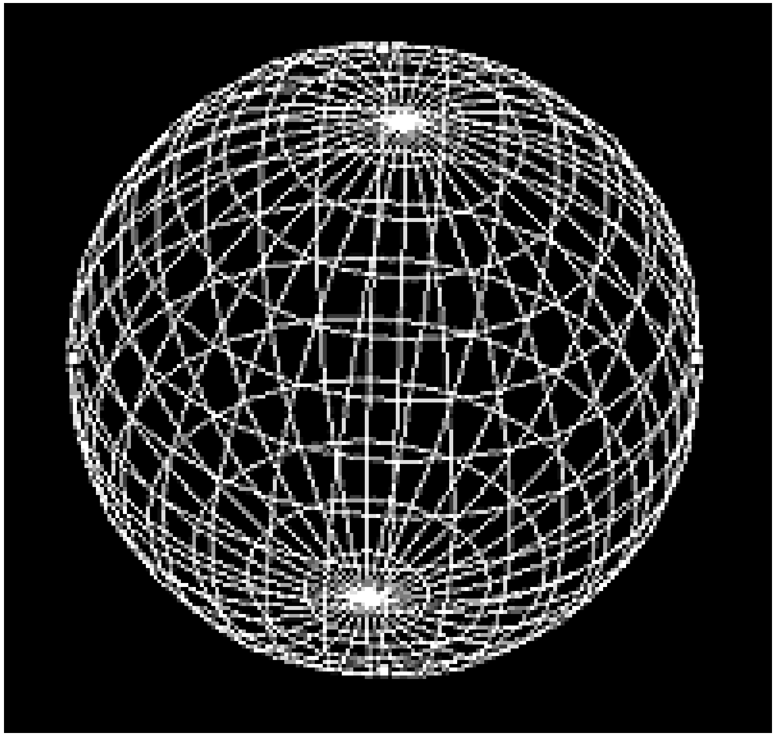
Hanging points, lines are eliminated by regularizing:

$$A\ \boldsymbol{op}^{*}\ B = \text{closure}\ (\ \text{interior}\ (A\ \boldsymbol{op}\ B))$$

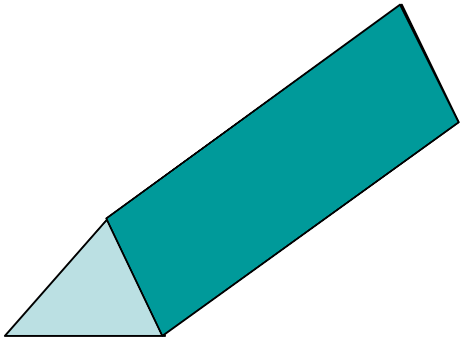


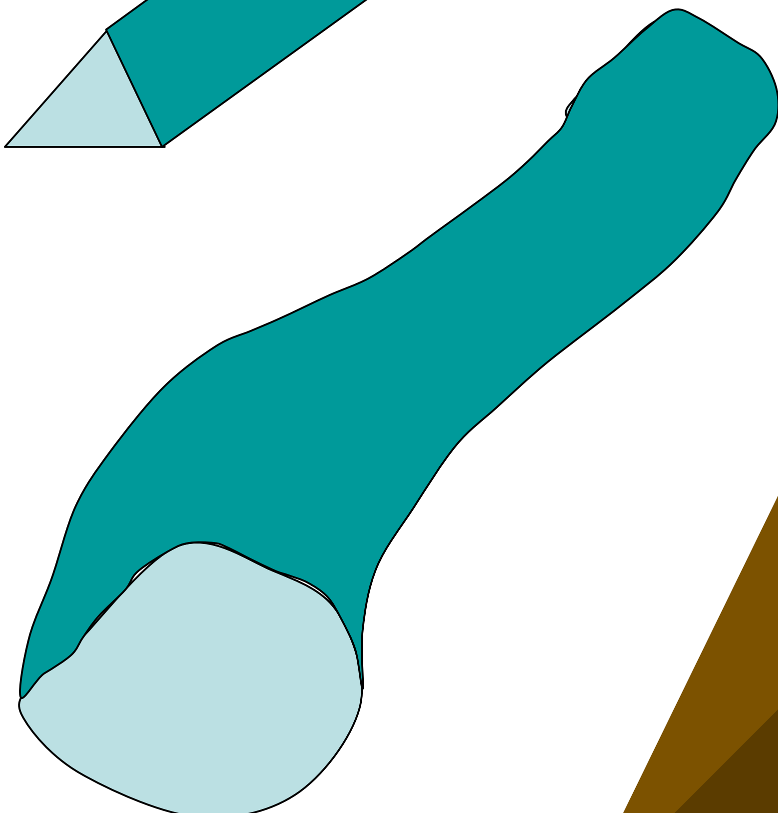
Sweep Representation

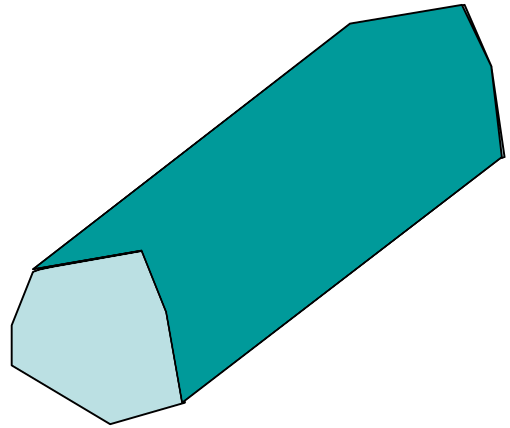


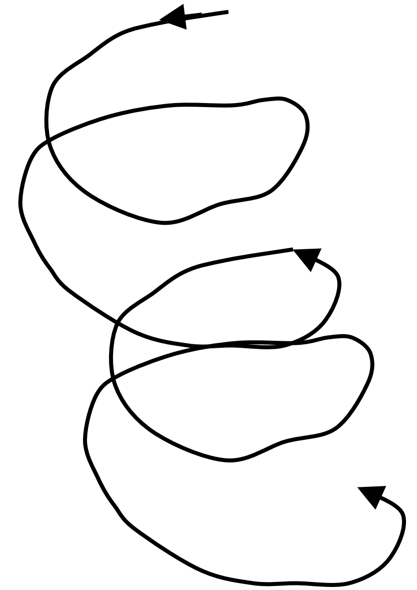


Some more examples of sweep representation.

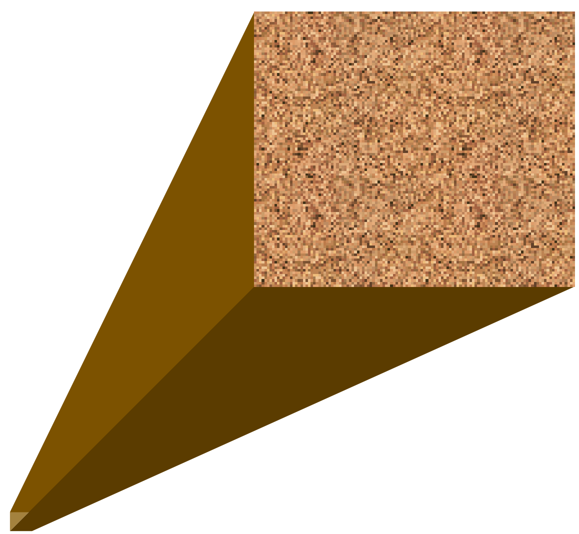








An arbitrary path for the sweep



B-reps (Boundary Representations)

Object is defined in terms of its surface boundaries, vertices, edges and faces. Curved surfaces are always approximated with polygons – piecewise linear/planar.

Very commonly used, in practice.

Use planar, polygonal boundaries, but may also use convex polygons or triangles.

Polyhedron **is a solid that is bounded by a set of polygons whose edges are each a member of an even number of polygons. Additional constraints will be discussed later on.**

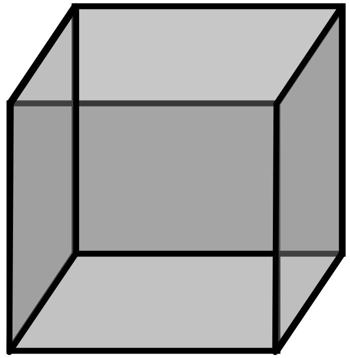
A simple polyhedron can always be deformed into a sphere. Polyhedron has no holes (not a torus).

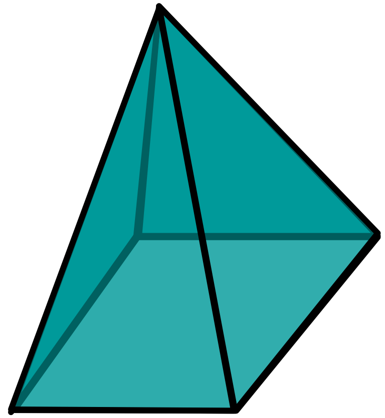
Euler's formula:

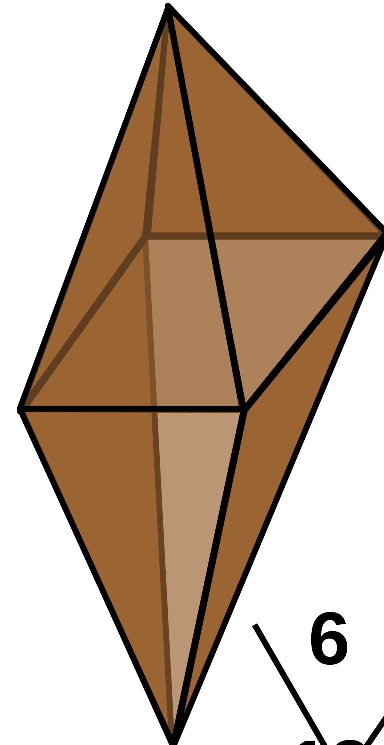
Let V be the number of vertices, E the number of edges, and F the number of faces of a simple polyhedron. Then

$$V - E + F = 2$$

Verify Euler's formula with these examples:





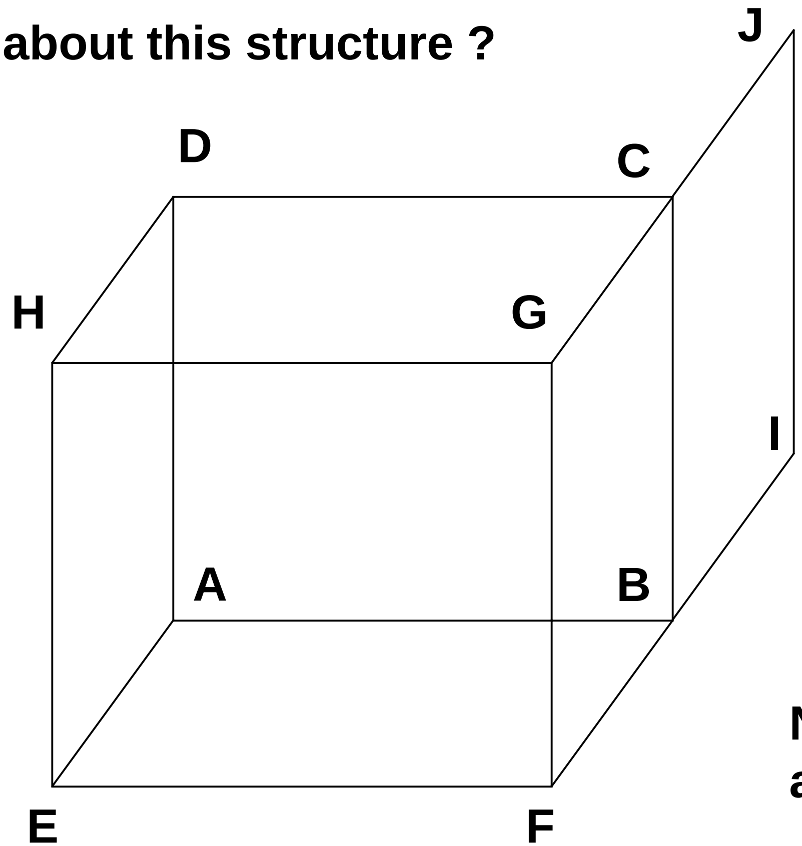


V =	8	5	~~6~~
E =	12	8	~~12~~
F =	6	5	~~9~~

Also applicable for curved edges and non-planar faces

	6
Actual:	12
	8

What about this structure ?



V = 10;

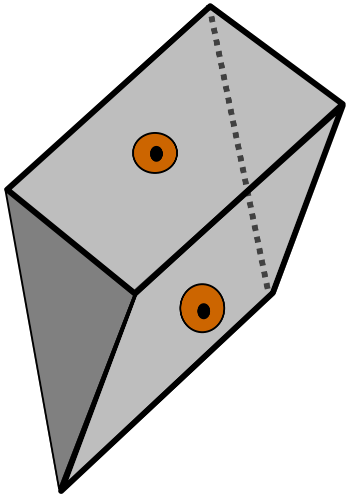
E = 15;

F = 7.

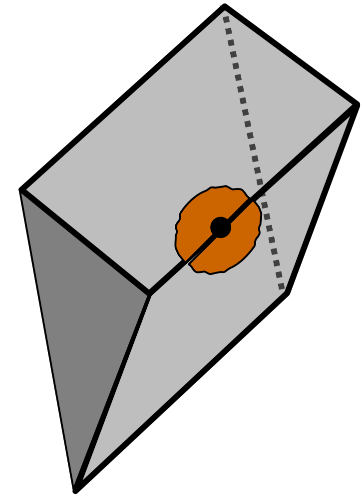
Formula still holds good,
but this is not a bound volume.

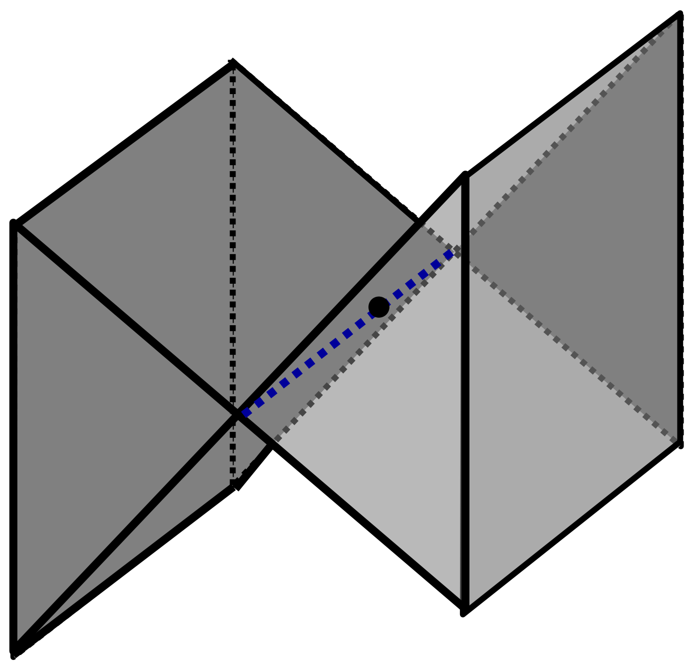
Need additional constraints to guarantee a solid object:

- Each edge must contain two vertices
- Must be shared exactly by two faces
- At least three edges must meet to form a vertex
- Faces must not interpenetrate



Solids with boundaries with 2-manifolds





Solids with boundaries but not 2-manifolds

Generalized Euler's formula for objects with 2-manifolds, and faces which may have holes:

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

where, H - No. of **HOLES** in the face;

G - No. of holes that pass through the object;

C - No. of separate **COMPONENTS** (or parts) of the object.

If C = 1, G is called as ***GENUS.*** If C > 1, G is the sum of the **GENERA** of its components.

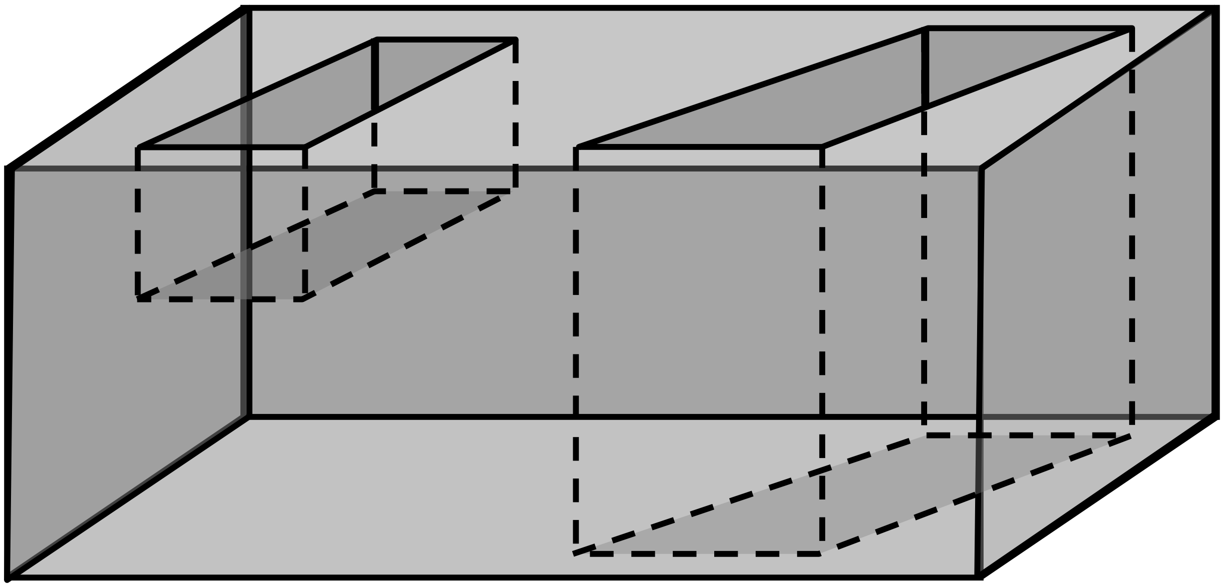
V = 24

E = 36

F = 15

H = 3

C = G = 1

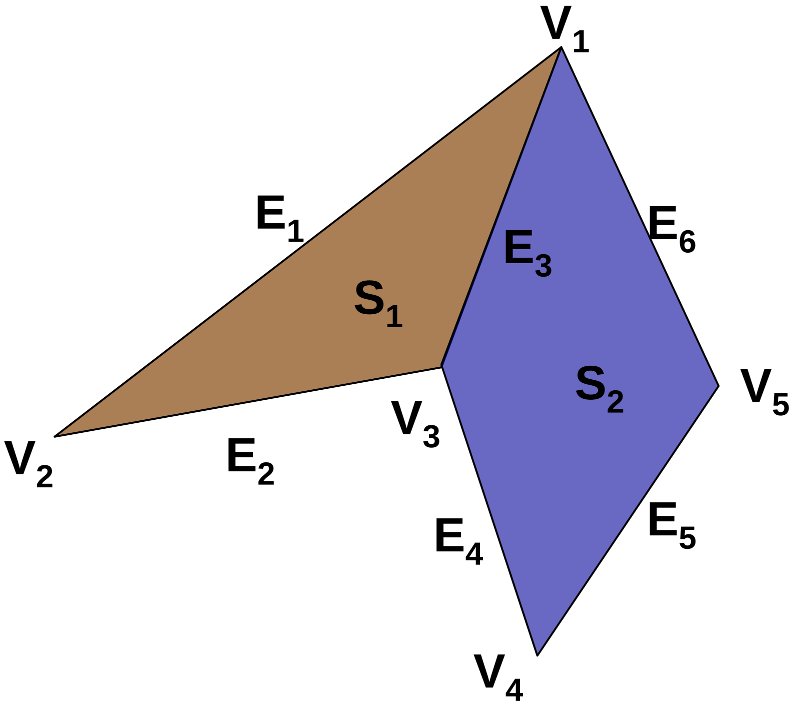


Vertex Table

V_1: X_1, Y_1, Z_1
V_2: X_2, Y_2, Z_2
V_3: X_3, Y_3, Z_3
V_4: X_4, Y_4, Z_4
V_5: X_5, Y_5, Z_5
V_6: X_6, Y_6, Z_6

Edge Table

E_1: V_1, V_2
E_2: V_2, V_3
E_3: V_3, V_1
E_4: V_3, V_4
E_5: V_4, V_5
E_6: V_5, V_1



Polygon Surface Table

S_1: E_1, E_2, E_3
S_2: E_3, E_4, E_5, E_6

Complete (expanded) Edge Table

E_1: V_1, V_2, S_1
E_2: V_2, V_3, S_1
E_3: V_3, V_1, S_1, S_2
E_4: V_3, V_4, S_2
E_5: V_4, V_5, S_2
E_6: V_5, V_1, S_2

Remember equations for plane normal, N ?

$$a=\sum_{i=1}^{n}(Y_i-Y_j)(Z_i+Z_j)$$

$$b=\sum_{i=1}^{n}(Z_i-Z_j)(X_i+X_j) \quad where \ j=i+1$$

$$c=\sum_{i=1}^{n}(X_i-X_j)(Y_i+Y_j)$$

Assume a triangle (or take any three vertices of a polygon):

$$If \ \ i=n, \quad j=1$$

$$c=(X_1-X_2)(Y_1+Y_2)+(X_2-X_3)(Y_2+Y_3)+(X_3-X_1)(Y_3+Y_1)$$

$$=X_1(Y_2-Y_3)+X_2(Y_3-Y_1)+X_3(Y_1-Y_2)$$

Similarly:

$$a=Y_1(Z_2-Z_3)+Y_2(Z_3-Z_2)+Y_3(Z_1-Z_2)$$

$$and$$

$$b=Z_1(X_2-X_3)+Z_2(X_3-X_2)+Z_3(X_1-X_2)$$

These expressions can be written in matrix form as:

$$a=\begin{vmatrix}1 & Y_1 & Z_1\\ 1 & Y_2 & Z_2\\ 1 & Y_3 & Z_3\end{vmatrix} \qquad b=\begin{vmatrix}X_1 & 1 & Z_1\\ X_2 & 1 & Z_2\\ X_3 & 1 & Z_3\end{vmatrix} \qquad a=\begin{vmatrix}X_1 & Y_1 & 1\\ X_2 & Y_2 & 1\\ X_3 & Y_3 & 1\end{vmatrix}$$

These are the solutions of the set of equations:

$$(A/D)X_k + (B/D)Y_k + (C/D)Z_k = -1; \;\; k = 1,2,3$$

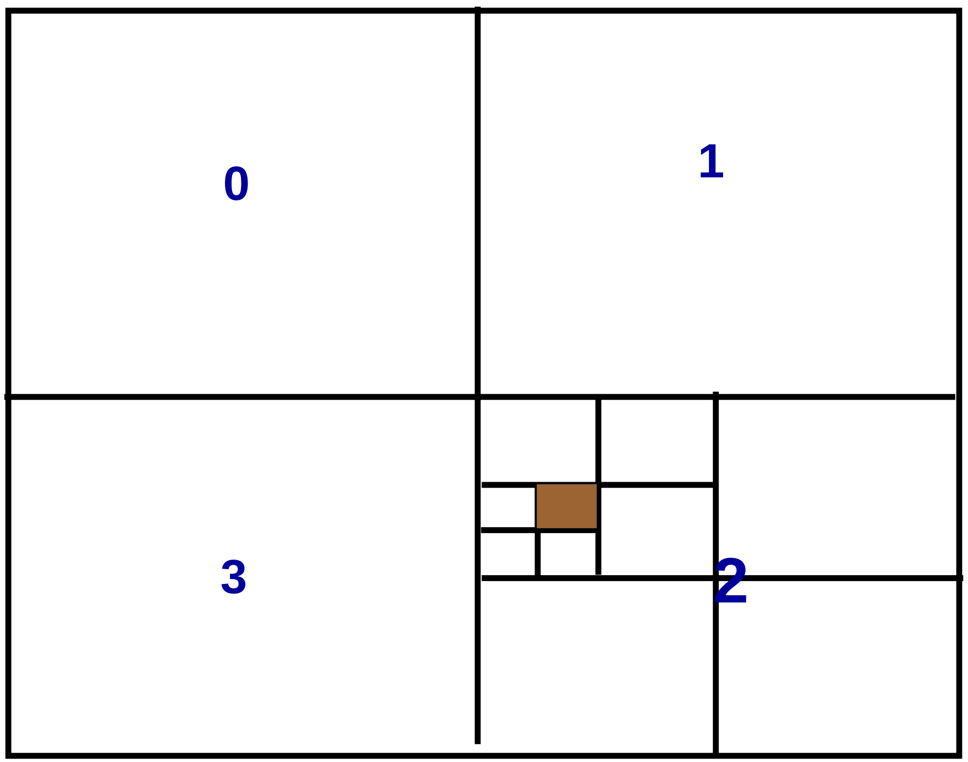
where,

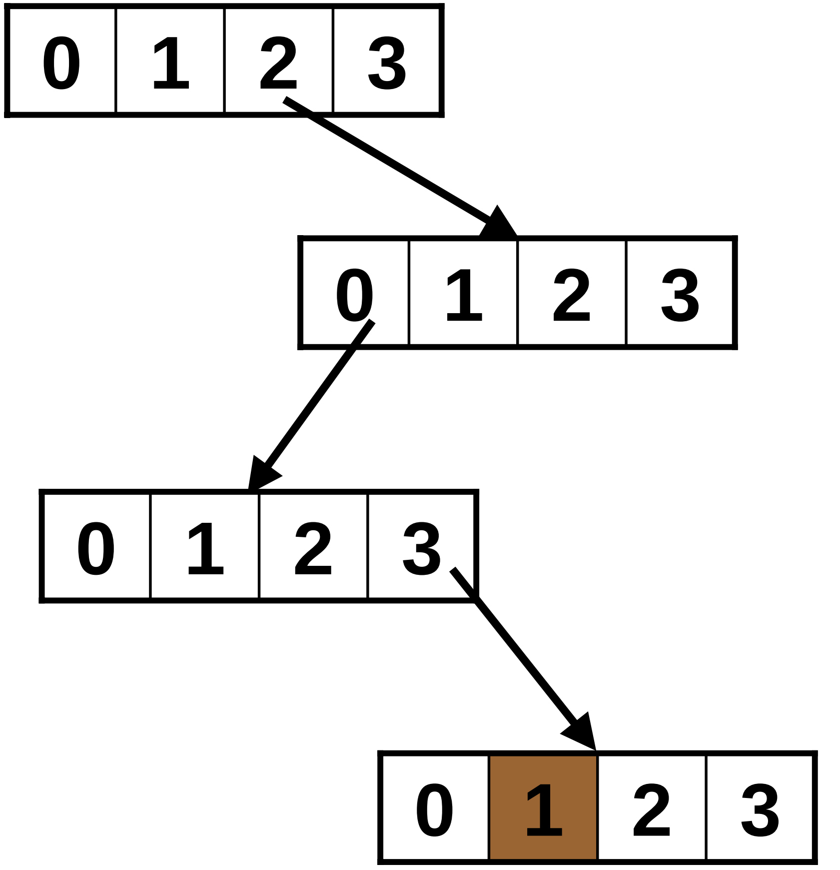
$$D = -X_1(Y_2Z_3 - Y_3Z_2) - X_2(Y_3Z_1 - Y_1Z_3) - X_3(Y_1Z_2 - Y_2Z_1)$$

Significance of D ? **D can also be obtained as, D = -N.P**

Octrees

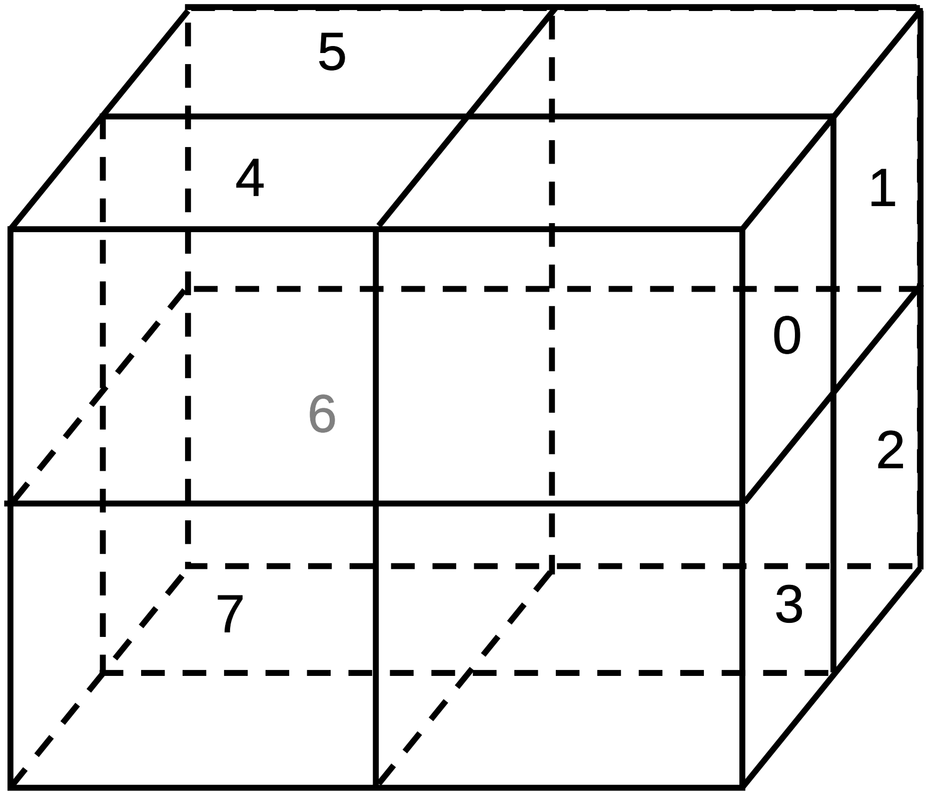
Let us visualize 2D **Quadtree** representation of an image or view.





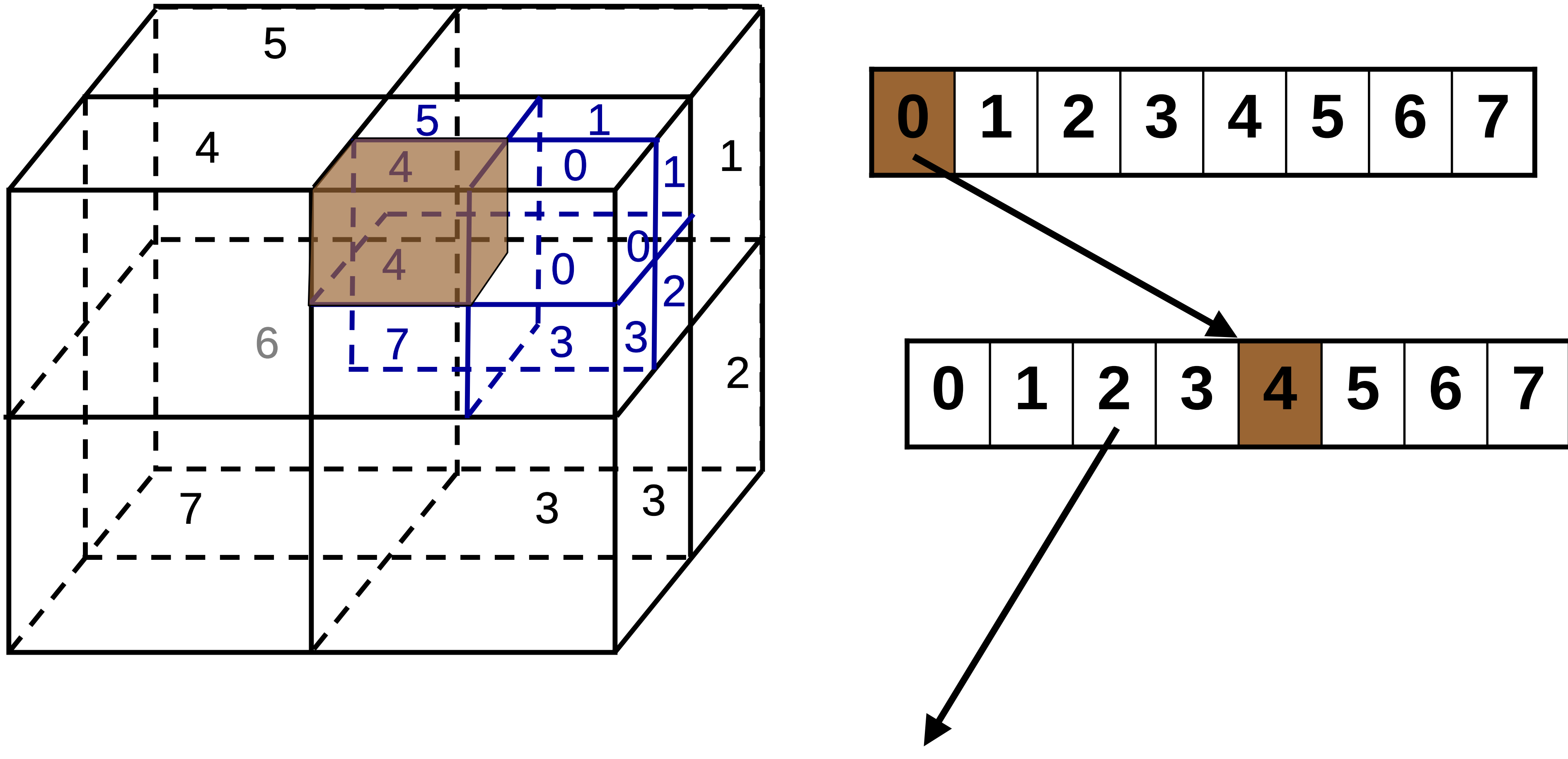
For an image with n*n pixels, the maximum number of levels is log(n).

Octrees

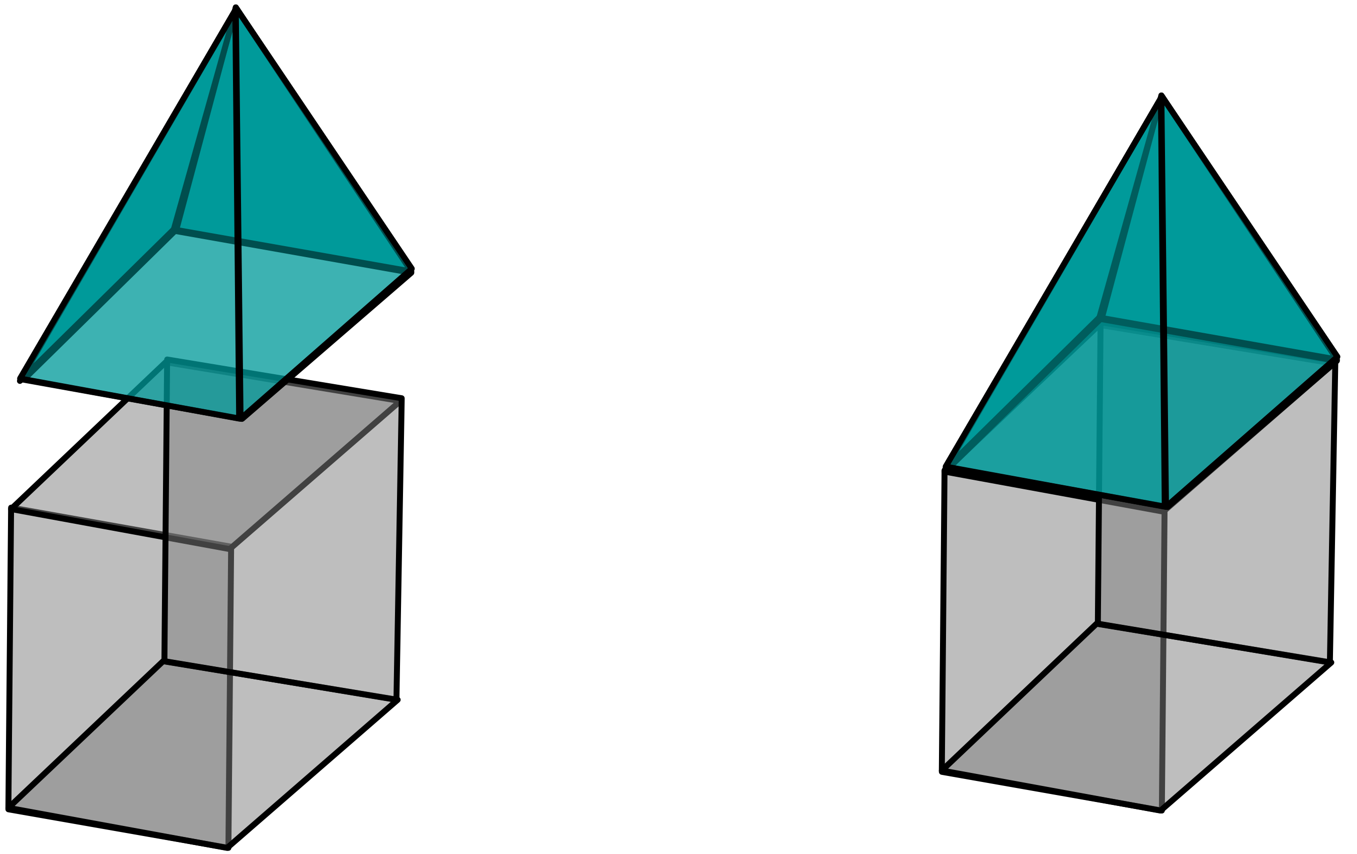


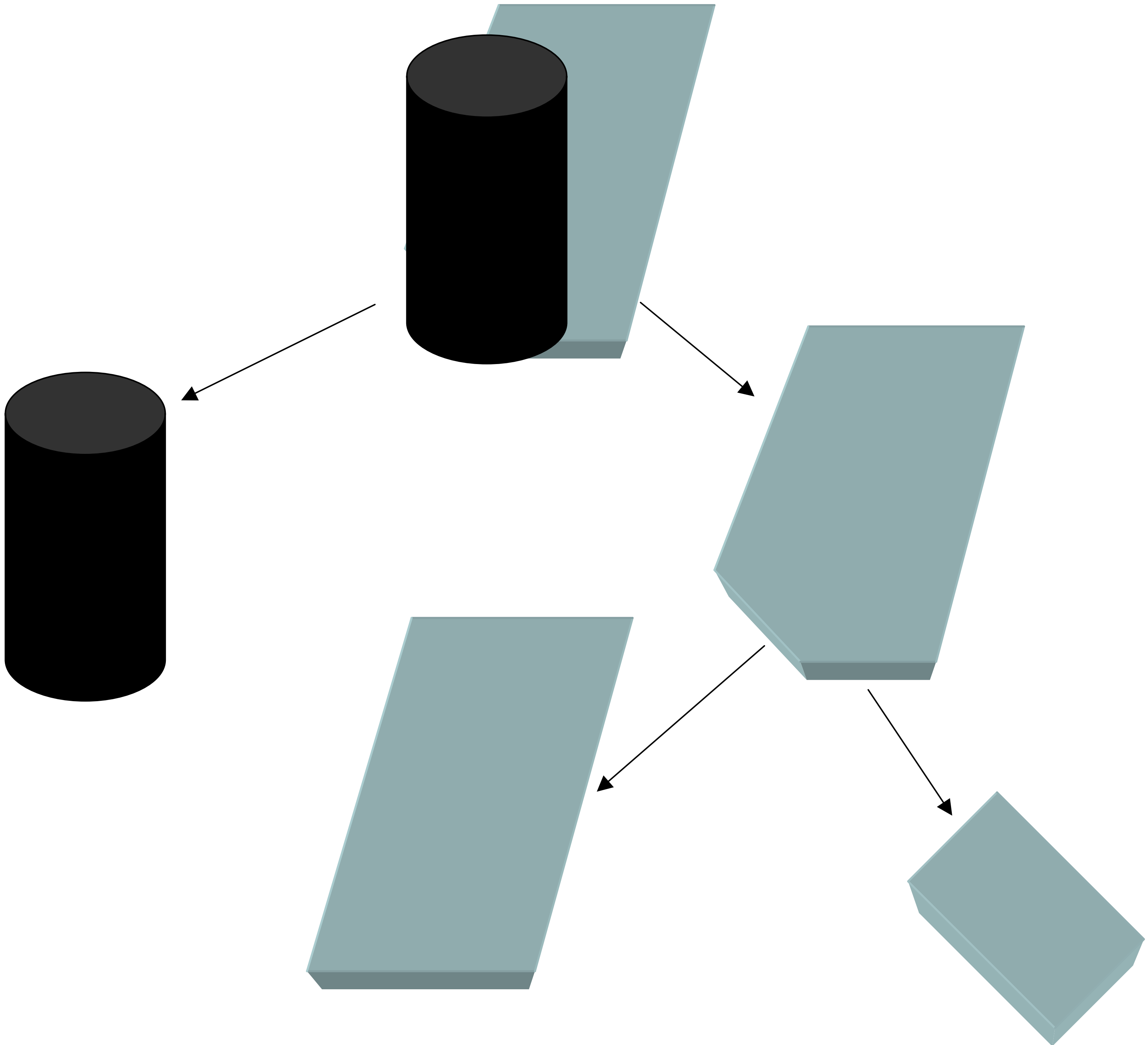
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**

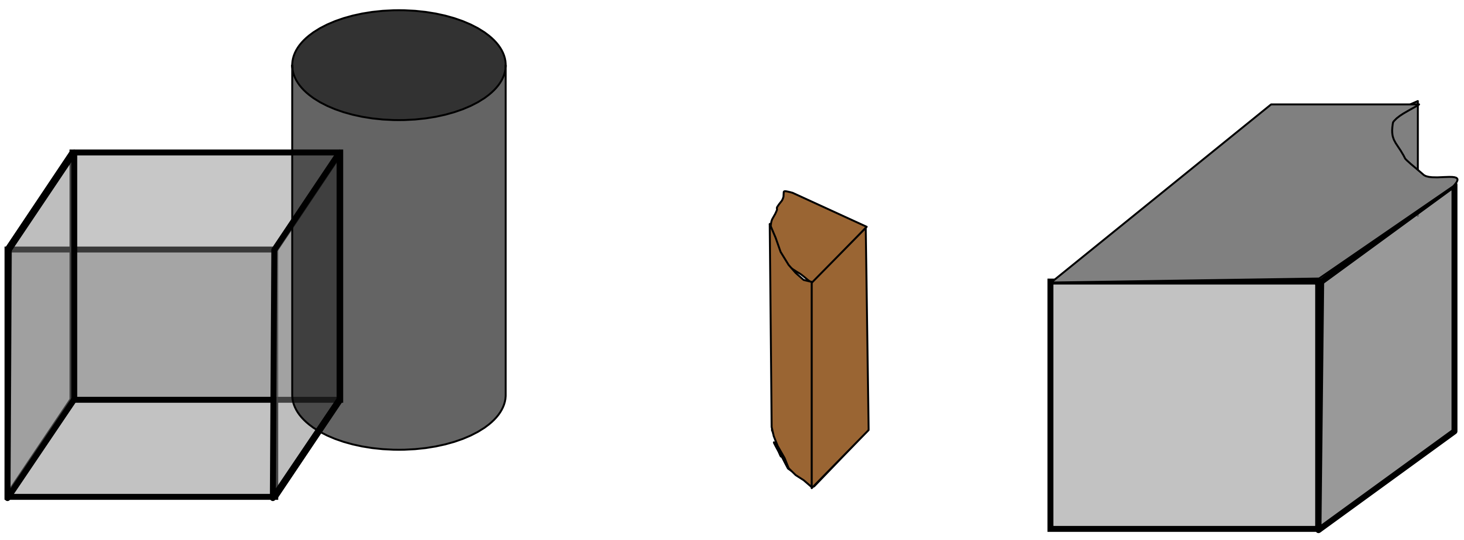
Individual elements are called Volume elements or <u>VOXELS</u>

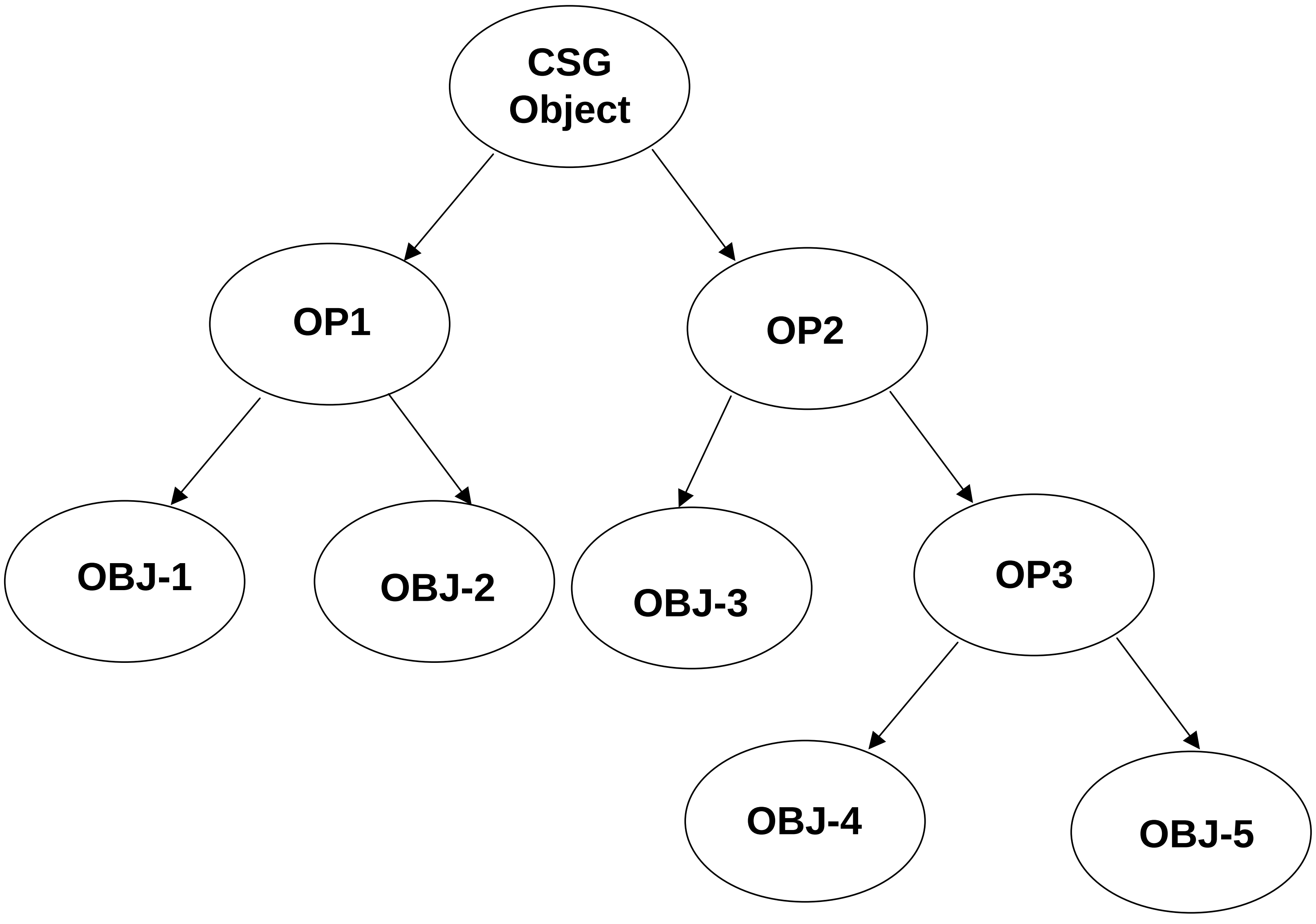

0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7

CSG – Constructive Solid Geometry








CSG
Object
OP1
OP2
OBJ-1
OBJ-2
OBJ-3
OP3
OBJ-4
OBJ-5

Criteria for comparing different solid modeling techniques:

- **Accuracy – Octrees and polygonal B-reps produce an approximation of objects. CSG with non-polyhedral primitives and B-reps that allow curved surfaces are good for high-quality graphics.**

- **Domain – Sweeps have limited domain. Octrees and B-reps provide a wide class of objects.**

- **Uniqueness – Octrees guarantee the uniqueness of a representation.**

- **Validity – B-reps are the most difficult to validate. One must find a way to ensure that vertex, edge and face data structures are valid (faces and edges may intersect). BSP trees and CSG are better.**

- **Closure (after Boolean operations) – Primitive instancing and sweeps are worse. B-reps can be used, but additional checking is necessary.**

- **Efficiency – Octress are better for hardware based solid modeling systems for a faster response with coarse pictures (results). Most algorithms are based on CSG and B-reps which are widely used for generating complex pictures.**