이화여자대학교
EWHA WOMANS UNIVERSITY


Sergei Berstein
(1880-1968)


Paul de Casteljau
(1930-1968)


Pierre Bezier
(1910-1999)

EWHA,
THE FUTURE
WE CREATE

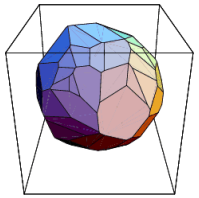
3D Object Representation

Young J. Kim

Polyhedron

Use a set of polygons to represent the surface of a polyhedron

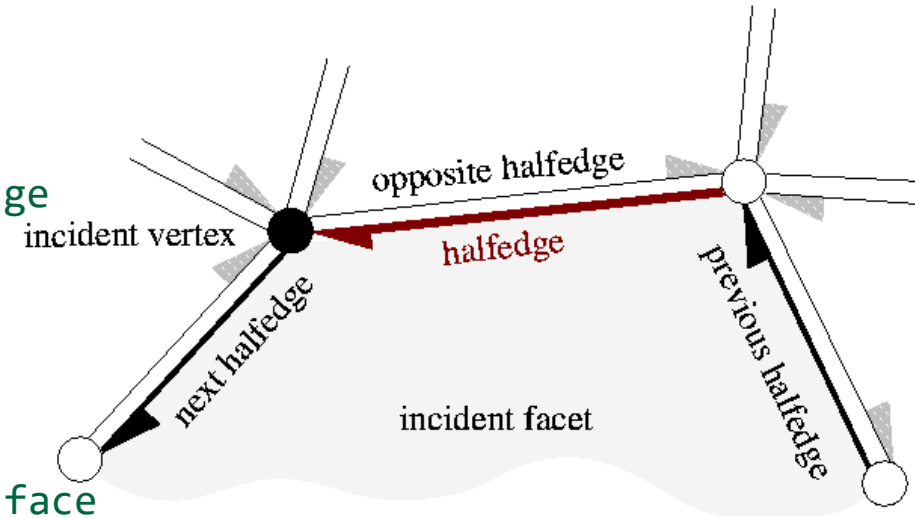
Consisting of Vertex, Edge, Face (facet)



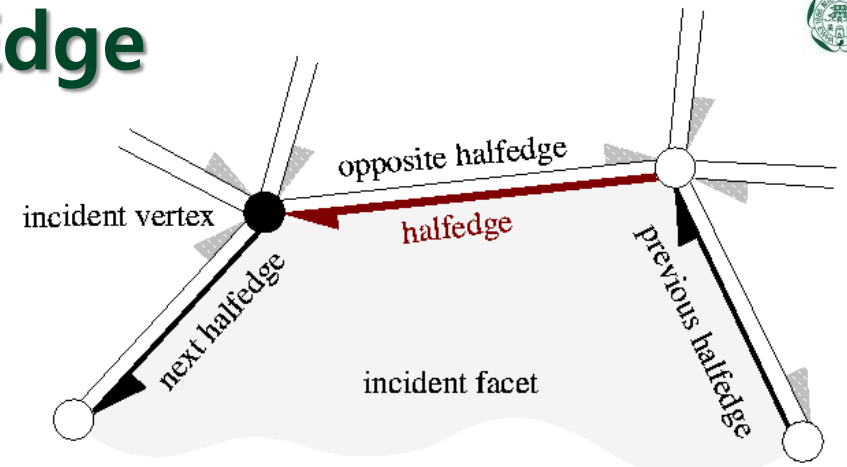
Data structure

- Half edge or doubly-connected edge list (DCEL)

```
struct HE_vert {
        float x; float y; float z;
        HE_edge* edge; // one of the half-edges emanating from the vertex
};

struct HE_edge {
        HE_vert* vert; // vertex at the end of the half-edge
        HE_edge* pair; // oppositely oriented adjacent half-edge
        HE_face* face; // face the half-edge borders
        HE_edge* next; // next half-edge around the face
};

struct HE_face {
        HE_edge* edge; // one of the half-edges bordering the face
};
```



Adjacency Queries using Half Edge



1. Finding vertices and face adjacent to an edge

```
HE_vert* vert_e = edge->vert;

HE_vert* vert_b = edge->pair->vert;
HE_face* face1 = edge->face;

HE_face* face2 = edge->pair->face;
```

2. Find all edges adjacent to a face

```
HE_edge* edge = face->edge;
do {
      edge = edge->next;

} while (edge != face->edge);
```

3. Find all edges adjacent to a vertex

```
HE_edge* edge = vert->edge;
do {
      edge = edge->pair->next;
} while (edge != vert->edge);
```

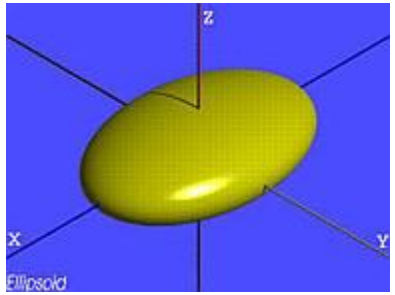
```
struct HE_vert {
 float x; float y; float z;
 HE_edge* edge; // one of the half-edges emanating from the vertex
};

struct HE_edge {
 HE_vert* vert; // vertex at the end of the half-edge
 HE_edge* pair; // oppositely oriented adjacent half-edge
 HE_face* face; // face the half-edge borders
 HE_edge* next; // next half-edge around the face
};

struct HE_face {
 HE_edge* edge; // one of the half-edges bordering the face
};
```

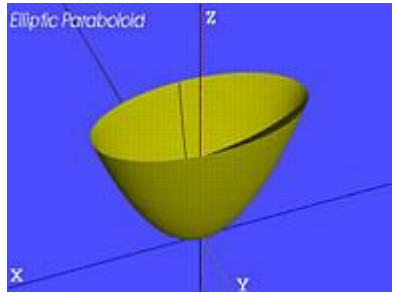
Quadric Surfaces

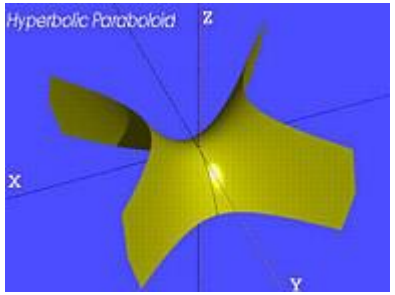
General form

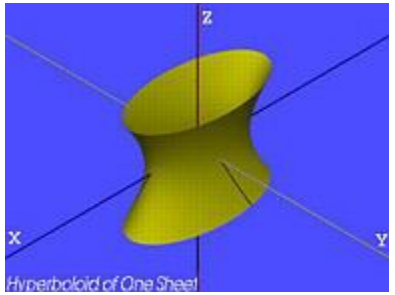
$$\sum_{i,j=1}^{D} Q_{i,j} x_i x_j + \sum_{i=1}^{D} P_i x_i + R = 0$$

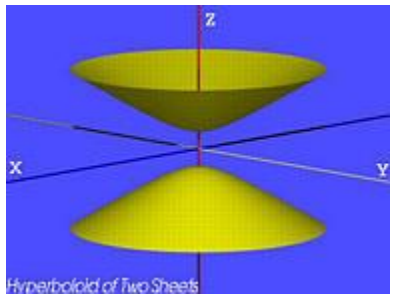
$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0.$$

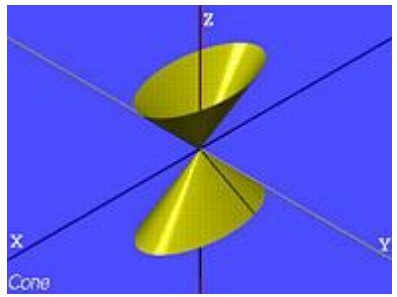


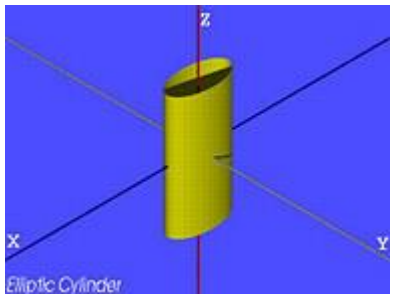


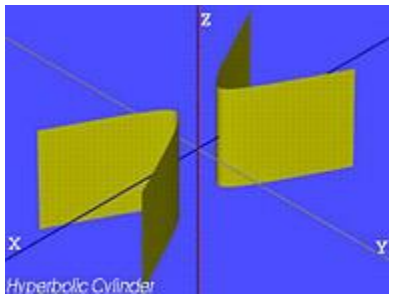










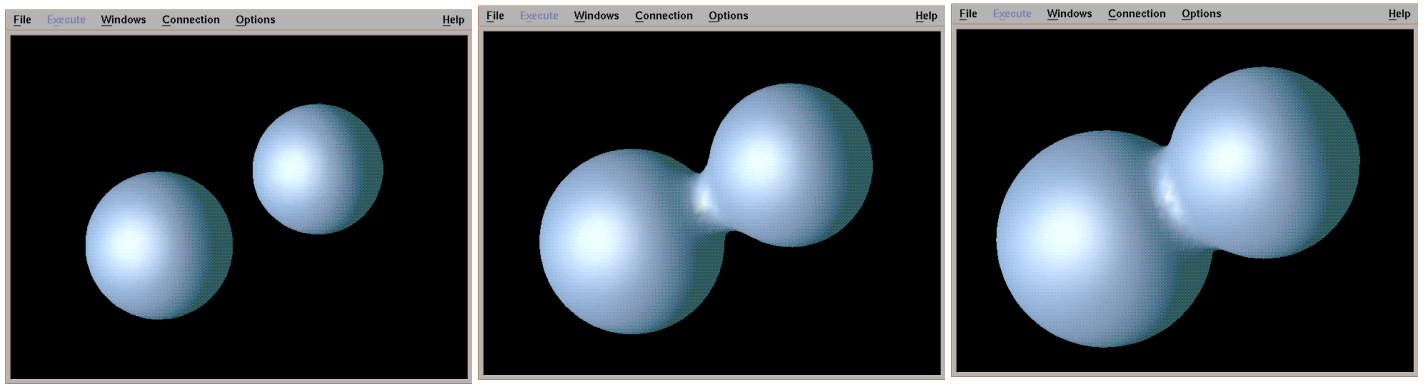


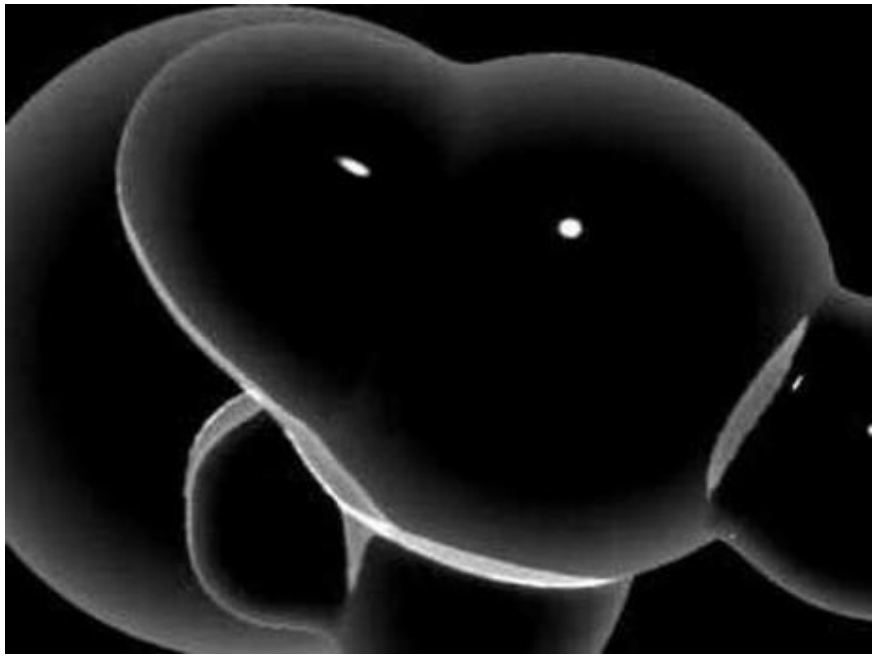
Blobby Object

Used to model non-rigid objects:

- Cloth, rubber, molecules, liquids, droplets

$$f(x,y,z) = \sum_k b_k e^{-a_k r_k^2} - T = 0 \qquad r_k^2 = x_k^2 + y_k^2 + z_k^2$$





1D Illustration 2D Illustration

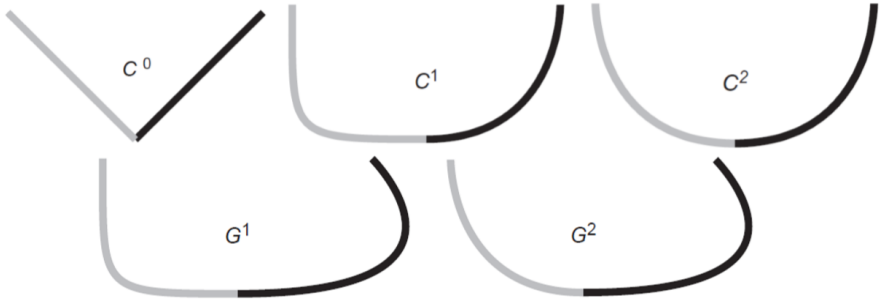
Splines

Controls points

Approximation vs. interpolation

Piecewise construction

- Continuity (C^n vs G^n)



Bezier, b-spline, NURBS

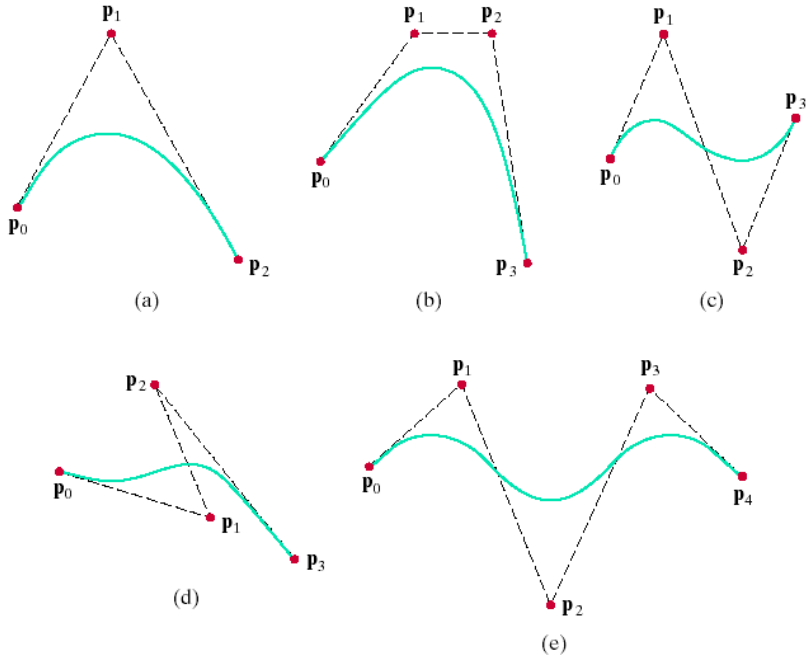
Bezier Curves and Surfaces

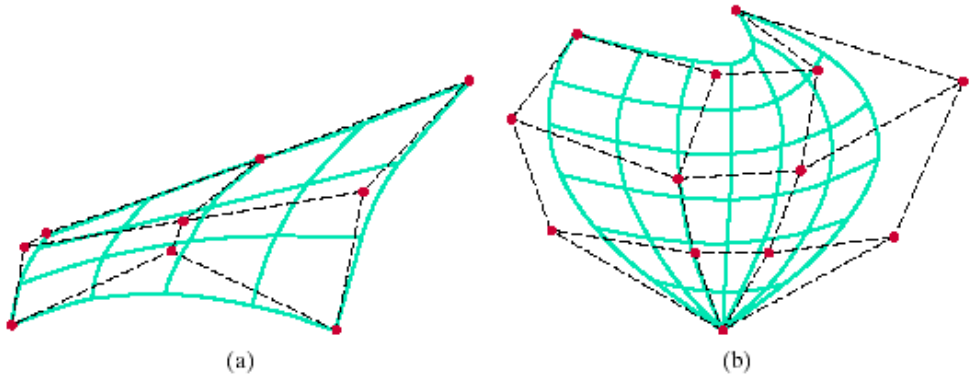
$$\{b_{0,0}(u) = 1\}$$

$$\left\{\begin{matrix} b_{0,1}(u) = 1-u \\ b_{1,1}(u) = u \end{matrix}\right\}$$

$$\left\{\begin{matrix} b_{0,2}(u) = (1-u)^2 \\ b_{1,2}(u) = 2(1-u)u \\ b_{2,2}(u) = u^2 \end{matrix}\right\}$$

$$\left\{\begin{matrix} b_{0,3}(u) = (1-u)^3 \\ b_{1,3}(u) = 3(1-u)^2u \\ b_{2,3}(u) = 3(1-u)u^2 \\ b_{3,3}(u) = u^3 \end{matrix}\right\}$$





$$P(u) = \sum_{k=0}^{n} p_k b_{k,n}(u), \qquad 0 \le u \le 1$$

$$b_{k,n}(u) = {}_nC_k u^k (1-u)^{n-k}, \qquad {}_nC_k = \frac{n!}{k!\,(n-k)!}$$

Bernstein Polynomials

Since $\sum_{k=0}^{n} b_{k,n}(u) = 1$, Bezier curves/surfaces have the convex hull property.

de Castlejau's Algorithm

$$P(u) = \sum_{k=0}^{n} p_k \, {}_nC_k u^k (1-u)^{n-k}, \; 0 \le u \le 1$$

이화여자대학교 EWHA WOMANS UNIVERSITY

Insert Web Page

This app allows you to insert secure web pages starting with https:// into the slide deck. Non-secure web pages are not supported for security reasons.

Please enter the URL below.

https:// webglfundamentals.org/webgl/lessons/

Preview

Web Viewer Terms | Privacy & Cookies

Insert Web Page

This app allows you to insert secure web pages starting with https:// into the slide deck. Non-secure web pages are not supported for security reasons.

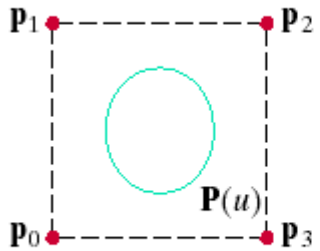
Please enter the URL below.

https:// webglfundamentals.org/webgl/webgl-3d-lathe-step-01.htm

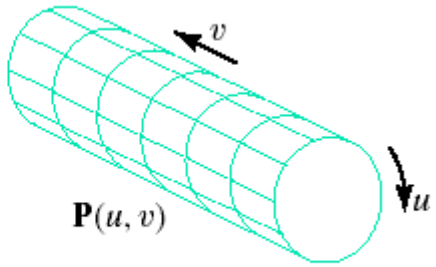
Preview

Web Viewer Terms | Privacy & Cookies

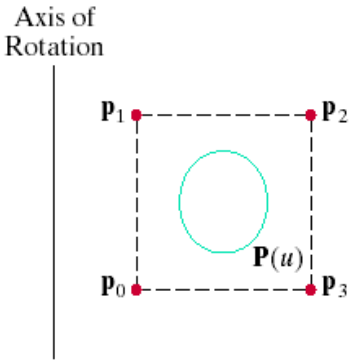
Sweep Representation



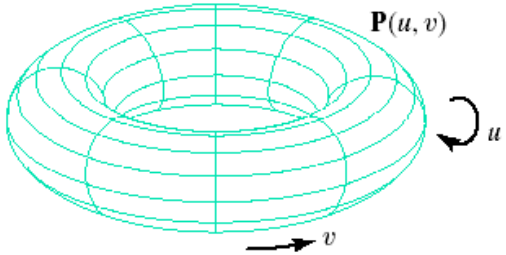
(a)



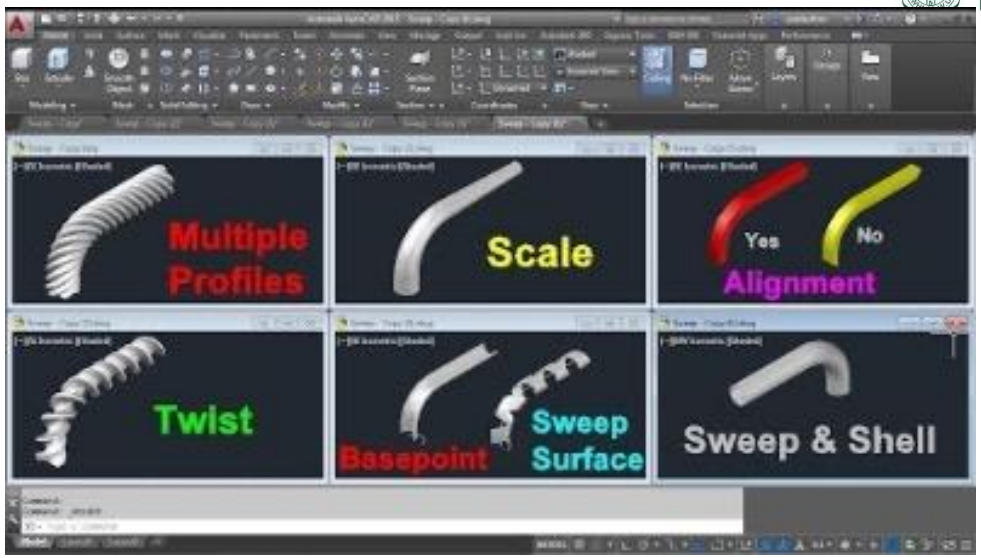
(b)

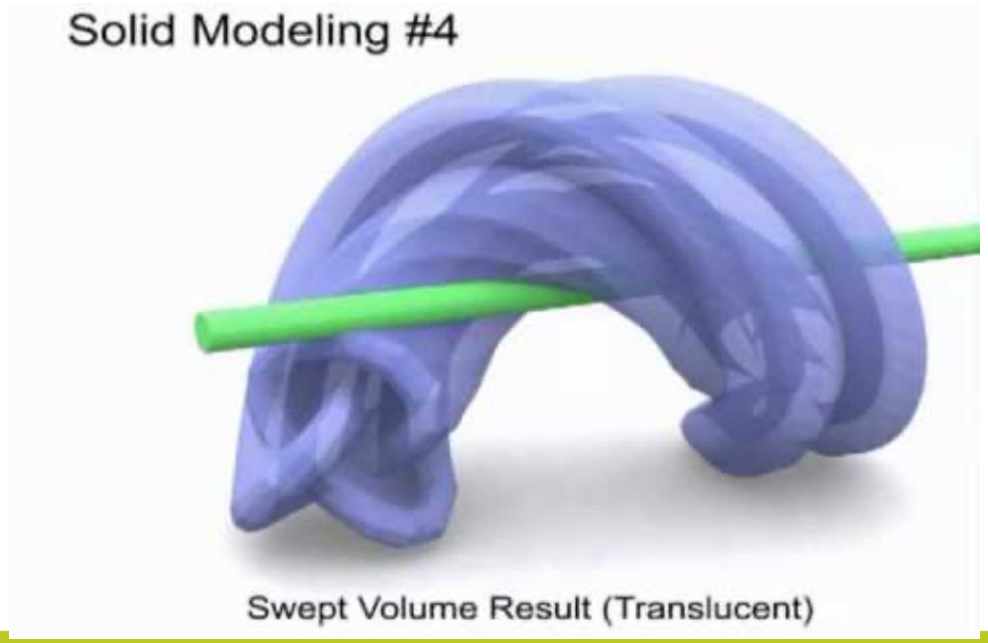


(a)



(b)



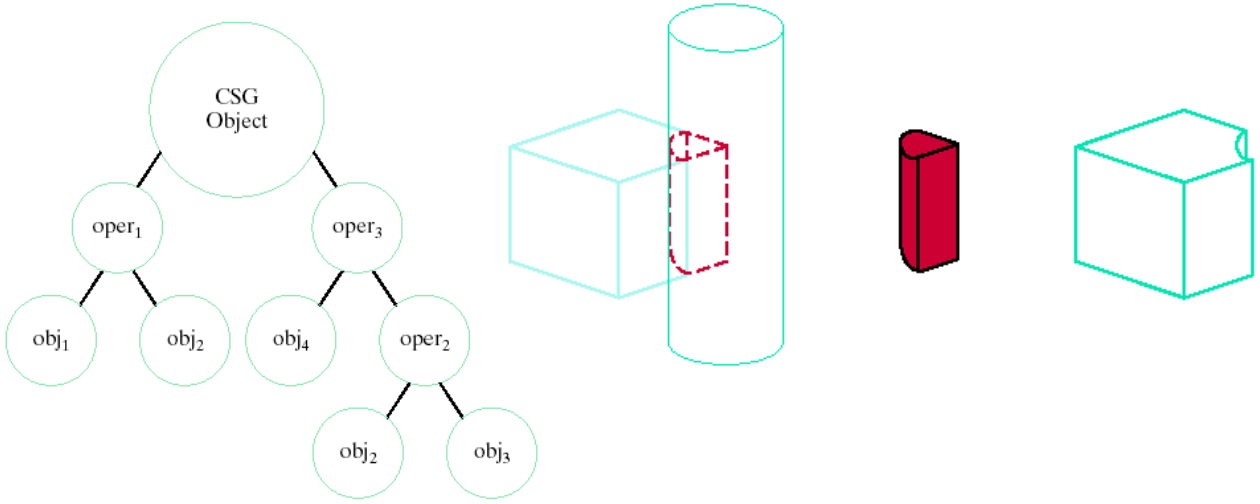


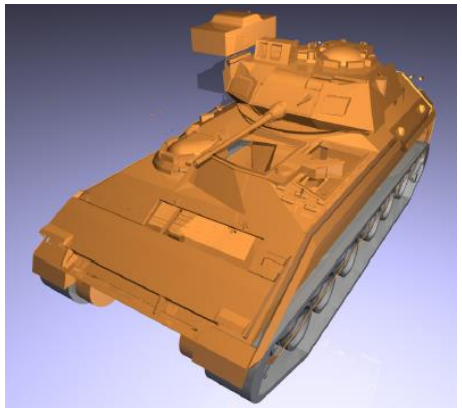
Constructive Solid Geometry

CSG

- Create a new object by performing a series of Boolean operations (union, intersection, difference)

CSG tree





8456 CSG operations

Zbrush