



C O M P A S

```
    if not callable(callback):
        raise Exception('Callback is not callable.')
    if not mesh:
        raise Exception('mesh is not defined')
    if not key in mesh:
        raise Exception('key is not defined')
    if not key in mesh.vertices():
        if any(mesh.vertices() == key):
            continue
        else:
            raise Exception('key is not a vertex')
    p = key_xyz[key]
    nbrs = mesh.vertex_neighbours(key, ordered=True)
    c = center_of_mass_polygon([key_xyz[nbr] for nbr in nbrs])
    attr = mesh.vertex[key]
    attr['x'] += d * (c[0] - p[0])
    attr['y'] += d * (c[1] - p[1])
    attr['z'] += d * (c[2] - p[2])
    if callback:
        callback(mesh, k, callback_args)
def smooth_mesh_length(mesh, lmin, lmax, fixed=None, kmax=100):
    if not mesh:
        raise Exception('mesh is not defined')
    if not lmin < lmax:
        raise Exception('lmin must be less than lmax')
    if not fixed:
        fixed = []
    if not callable(callback):
        raise Exception('Callback is not callable.')
    if not fixed or [fixed]:
        fixed = set(fixed)
    for k in range(kmax):
```

Module 3

Volumetric Modelling

dear all

for today's COMPAS course, please try to install the necessary package and library as follows:

1.in the terminal (mac) or anaconda prompt (pc), activate the environment

`conda activate ita19`

2.install compas_vol directly from github:

`pip install git+https://github.com/dbt-ethz/compas_vol`

3.if you don't have git installed, first do so by running

`conda install git`

4.you can also clone the repository with your client of choice (sourcetree, github desktop, ...), then cd into the folder and run `pip install -e .`

5.you will also need ipyvolume for the mesh display inside a notebook.

install it by running

`conda install -c conda-forge ipyvolume`

let me know if you encounter any problems, i'm happy to help

Jipa, A., Bernhard, M., Ruffray, N., Wangler, T., Flatt, R., & Dillenburger, B. (2017). *skelETHon Formwork*. In SIGraDi 2017, XXI Congreso de la Sociedad Ibero-americana de Gráfica Digital. Concepción, Chile.

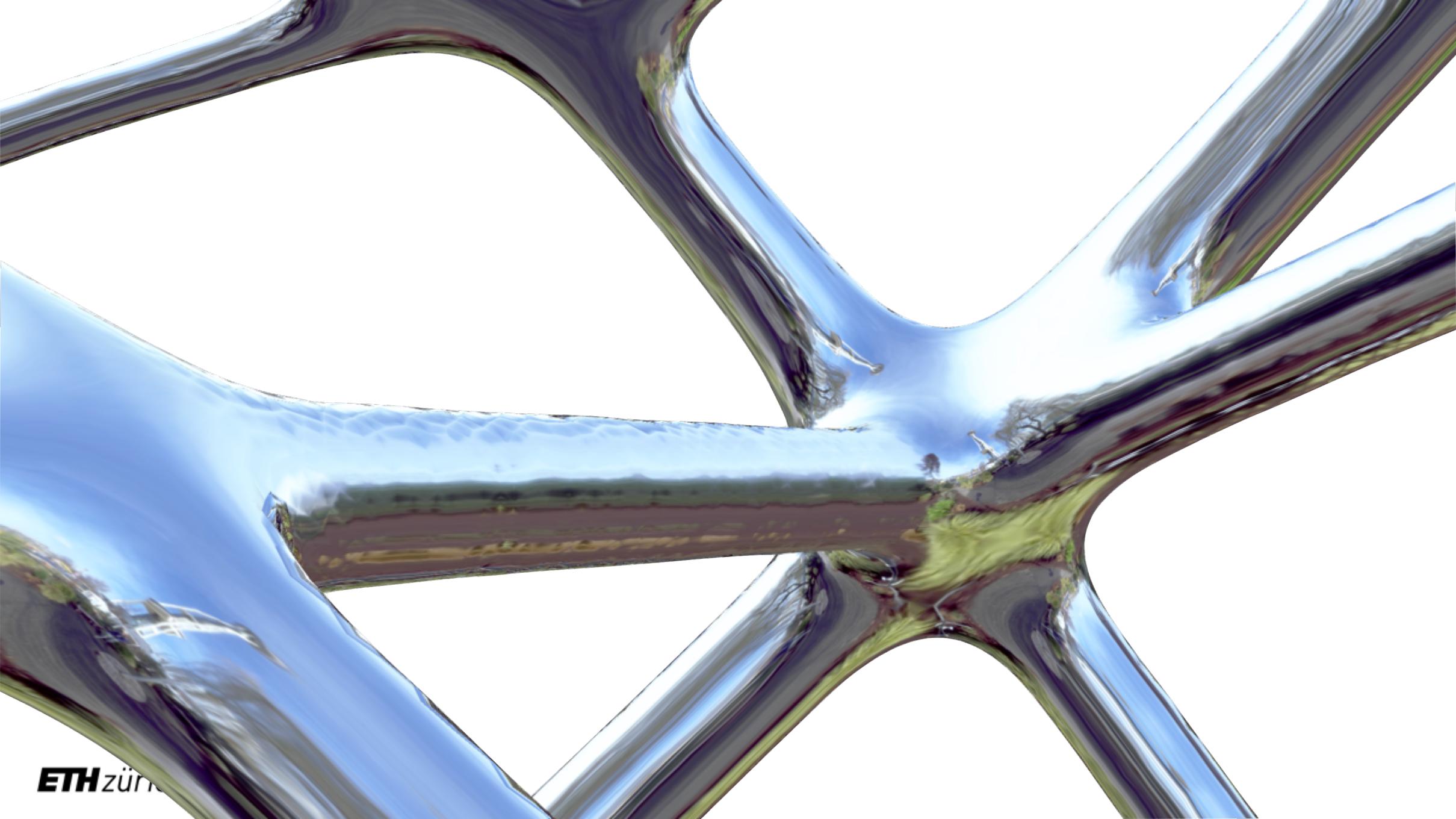
Jipa, A., Bernhard, M., Ruffray, N., Wangler, T., Flatt, R., & Dillenburger, B. (2017). *skelETHon Formwork 3D Printed Plastic Formwork for Load-Bearing Concrete Structures*. In Blucher Design Proceedings (pp. 345–352). São Paulo

Jipa, A., Bernhard, M., & Dillenburger, B. (2017). *Submillimeter Formwork*. 2017 TxA Emerging Design + Technology. Austin, Texas, USA.

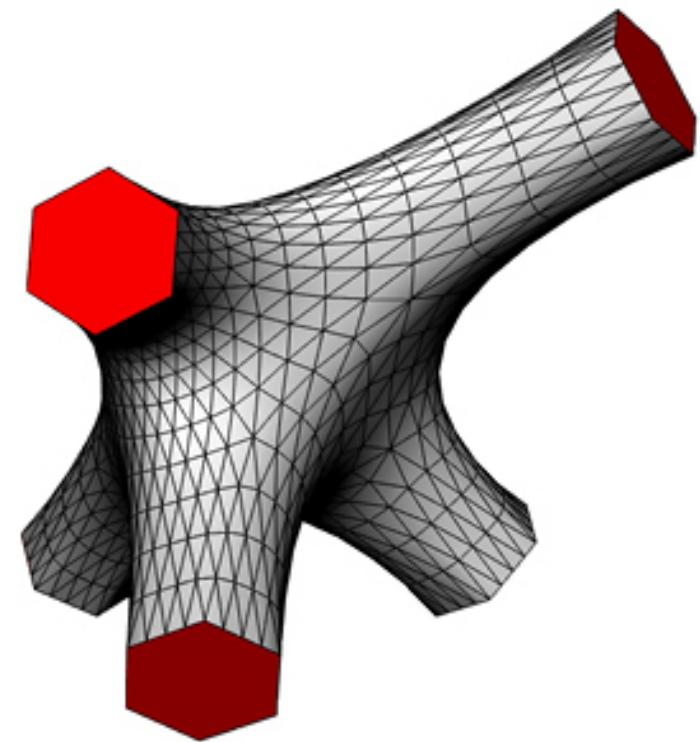
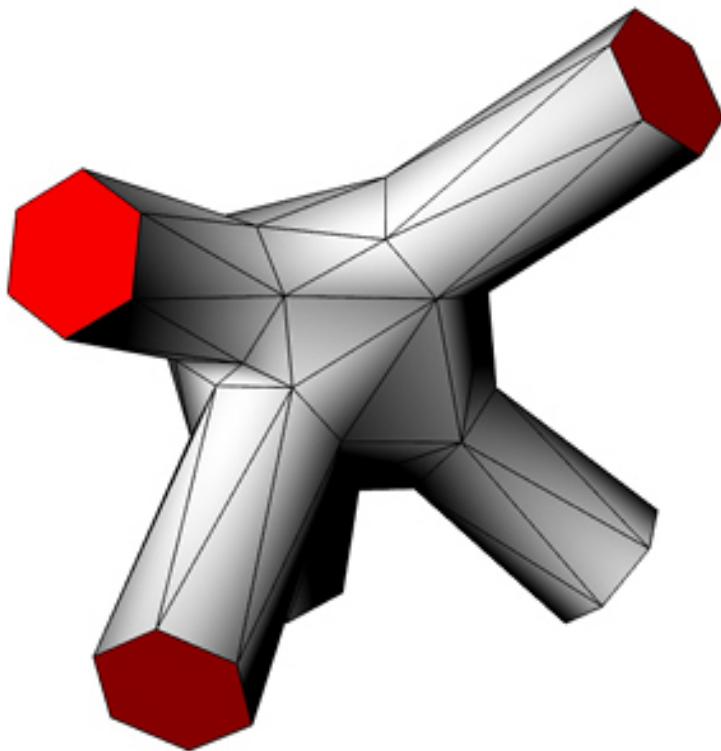
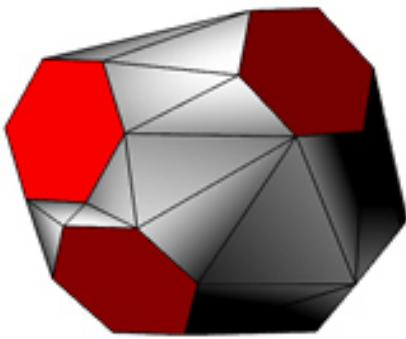
Concrete Canoe

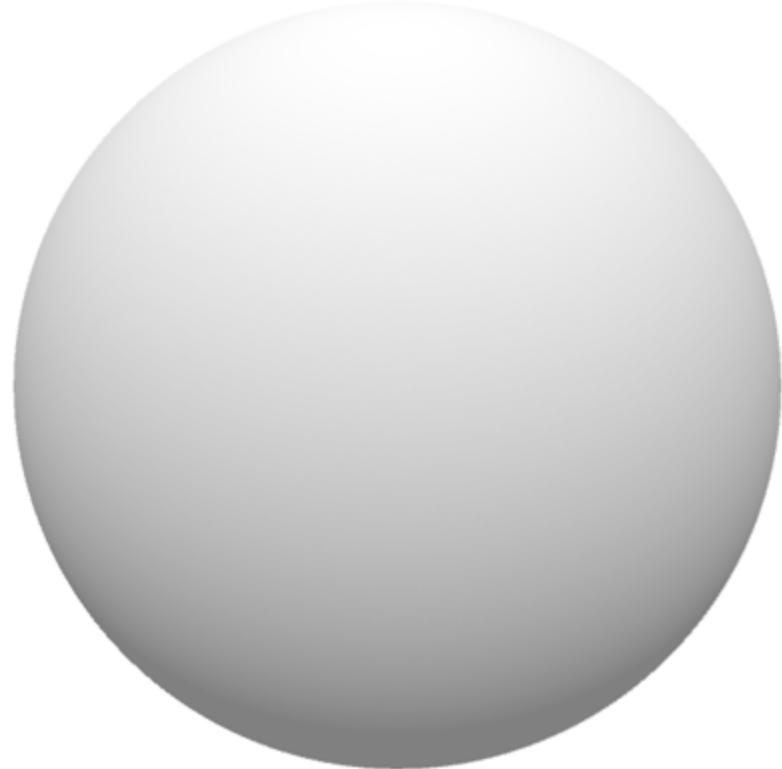
Photo: Andrei Jipa

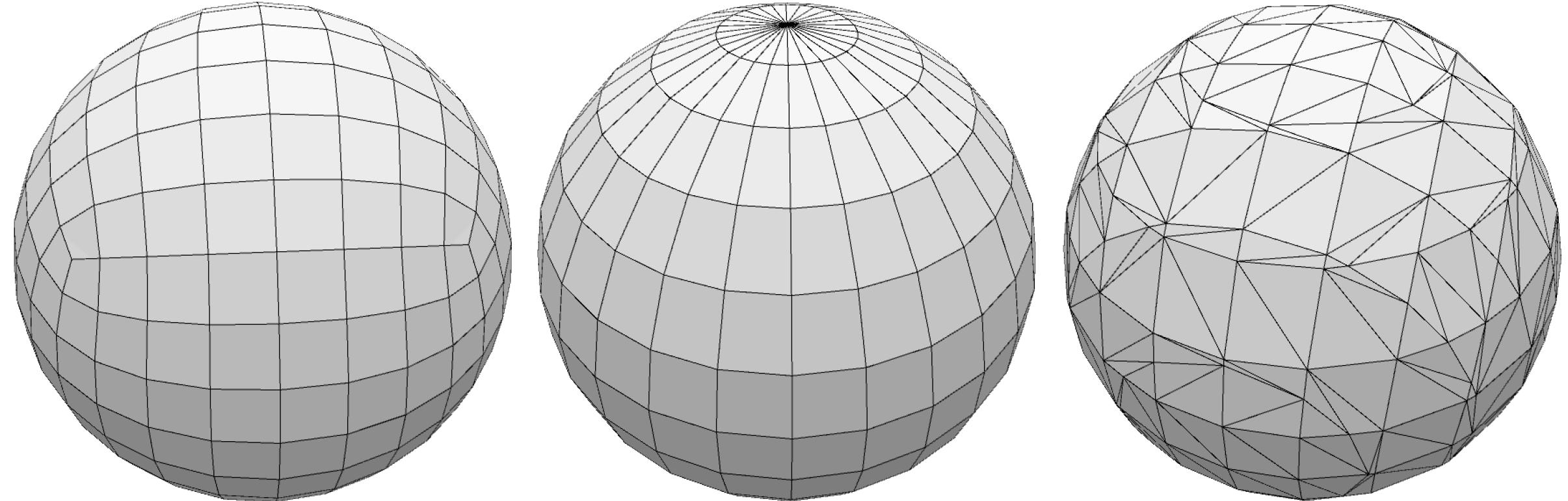




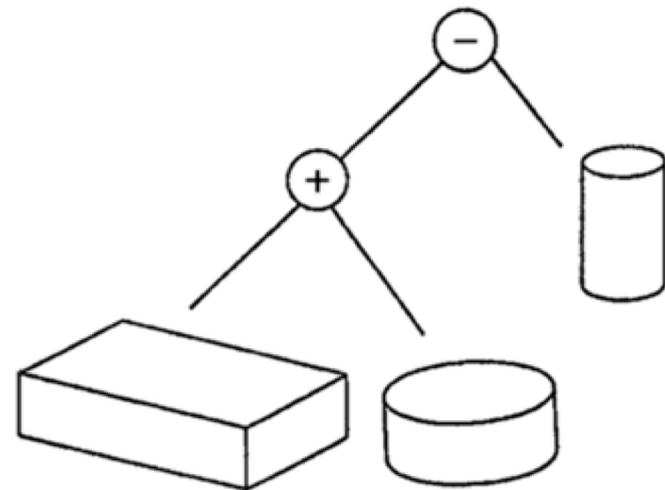
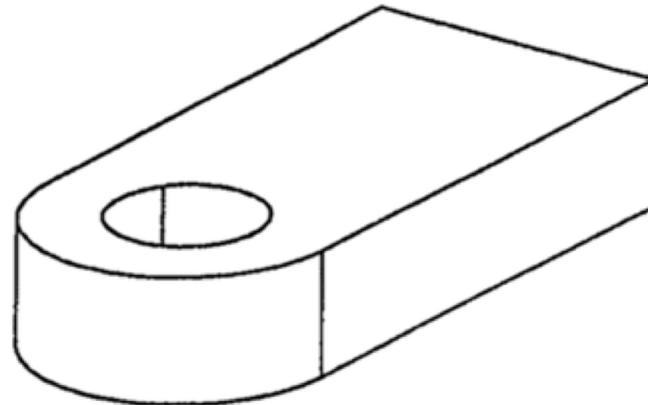
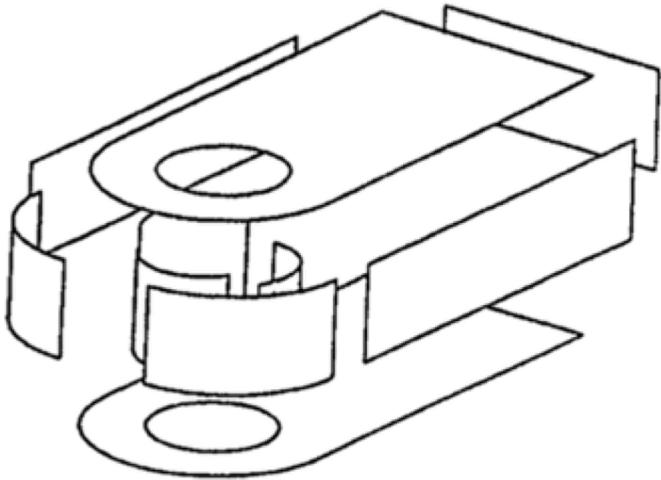
Exoskeleton



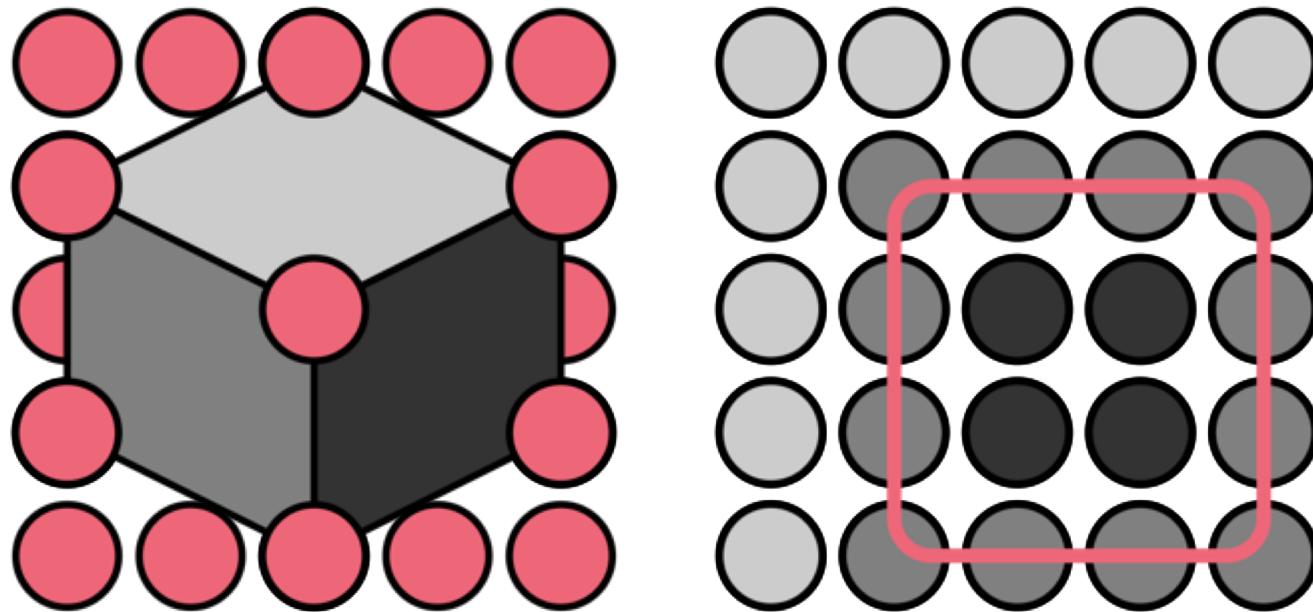




Surface versus Solid Modelling



On Full and
Empty Space



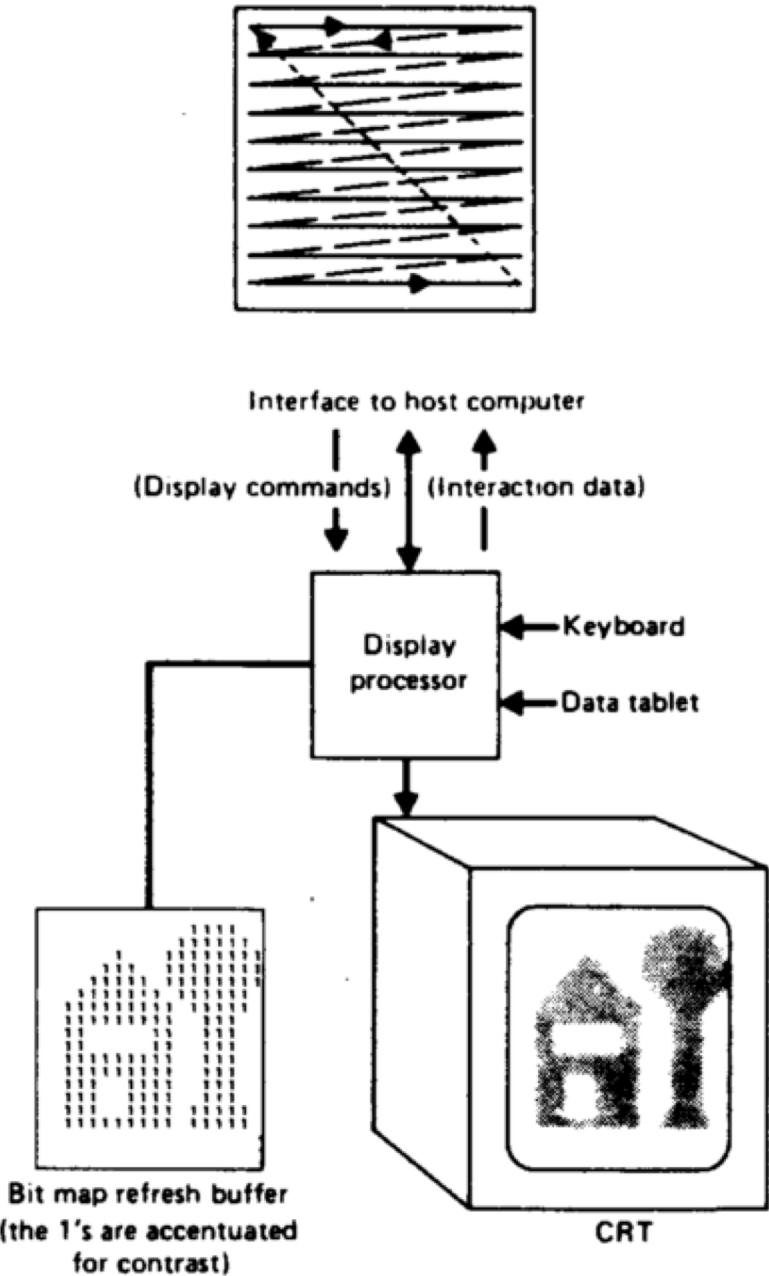
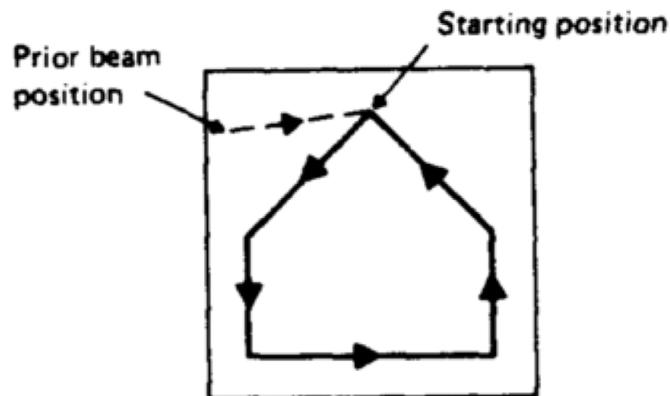
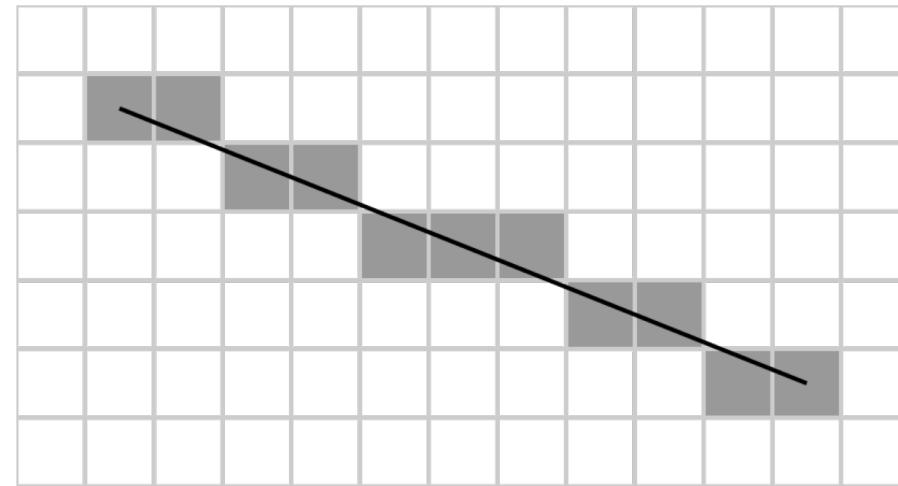
Voxels?

MagicaVoxel



Display Technology

Andries van Dam, 1983
An Overview of Interactive Computer Graphics



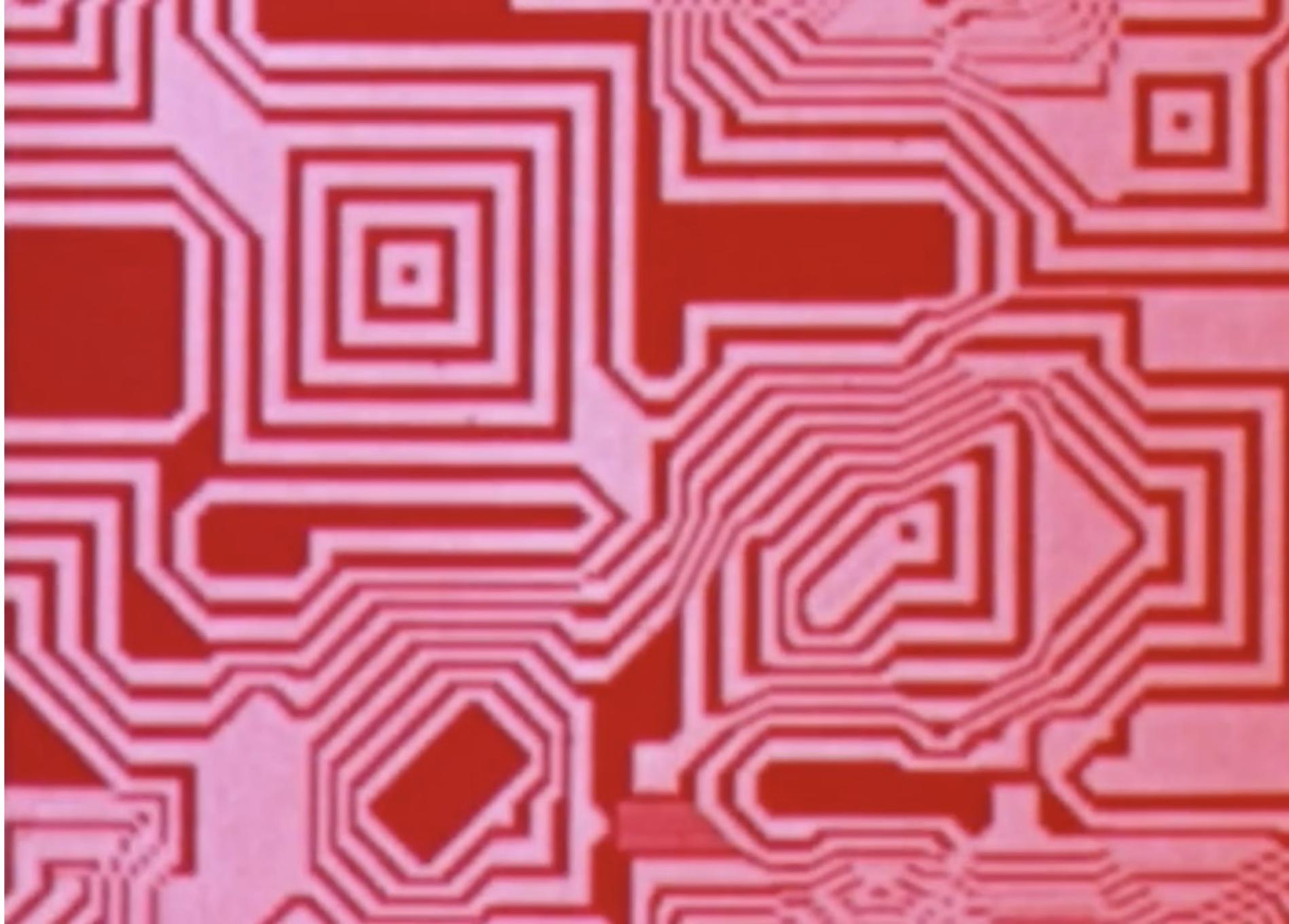
Cathode Ray Tubes

*Ivan Sutherland, 1963
Sketchpad*



Pixillation

Pixillation
Lillian F. Schwartz, 1970

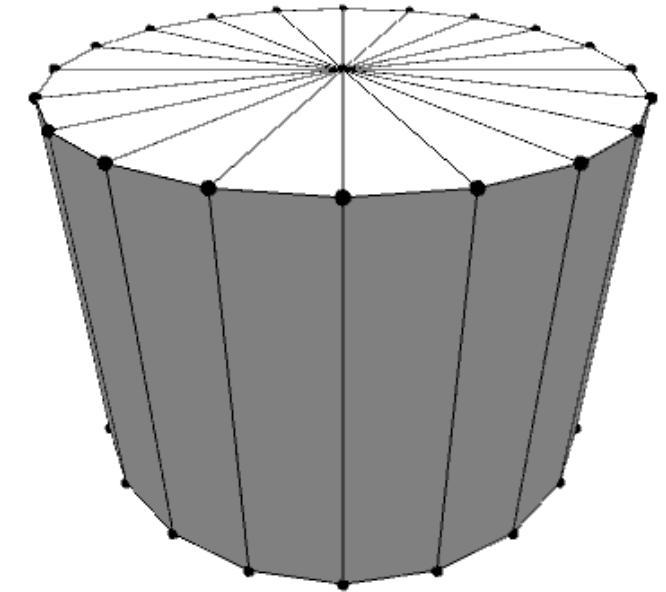
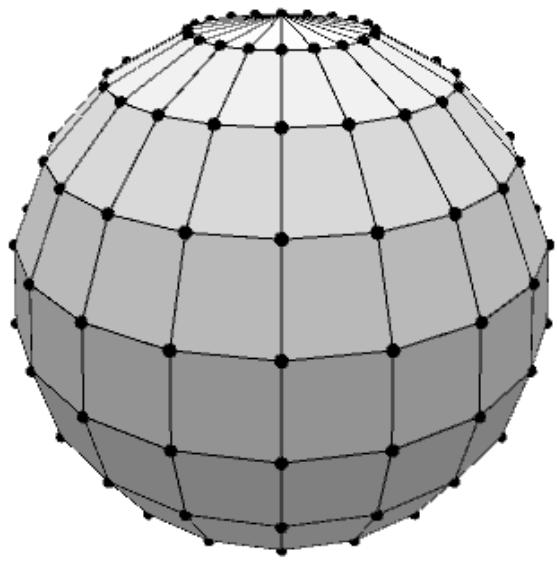
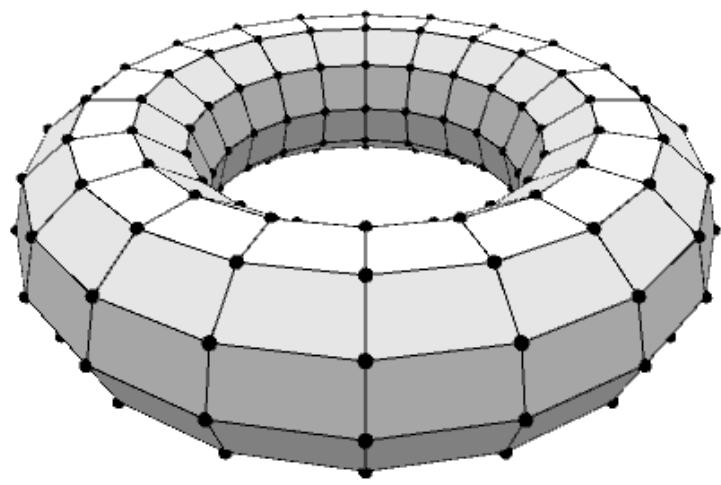


卷之三

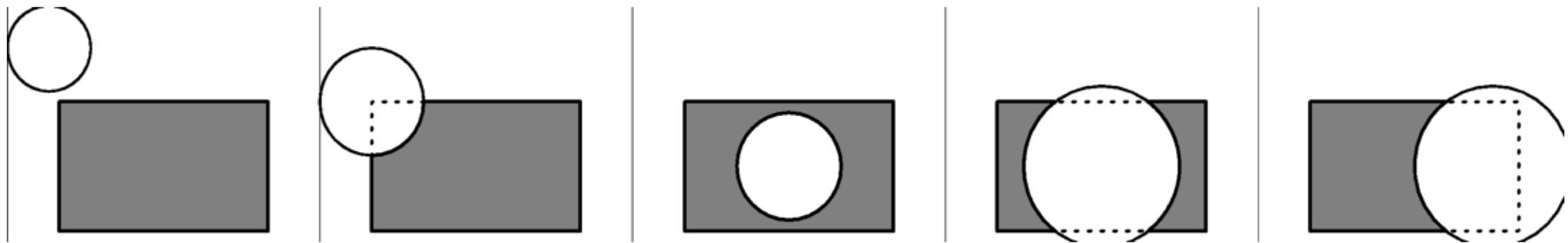
卷之三

卷之三

卷之三

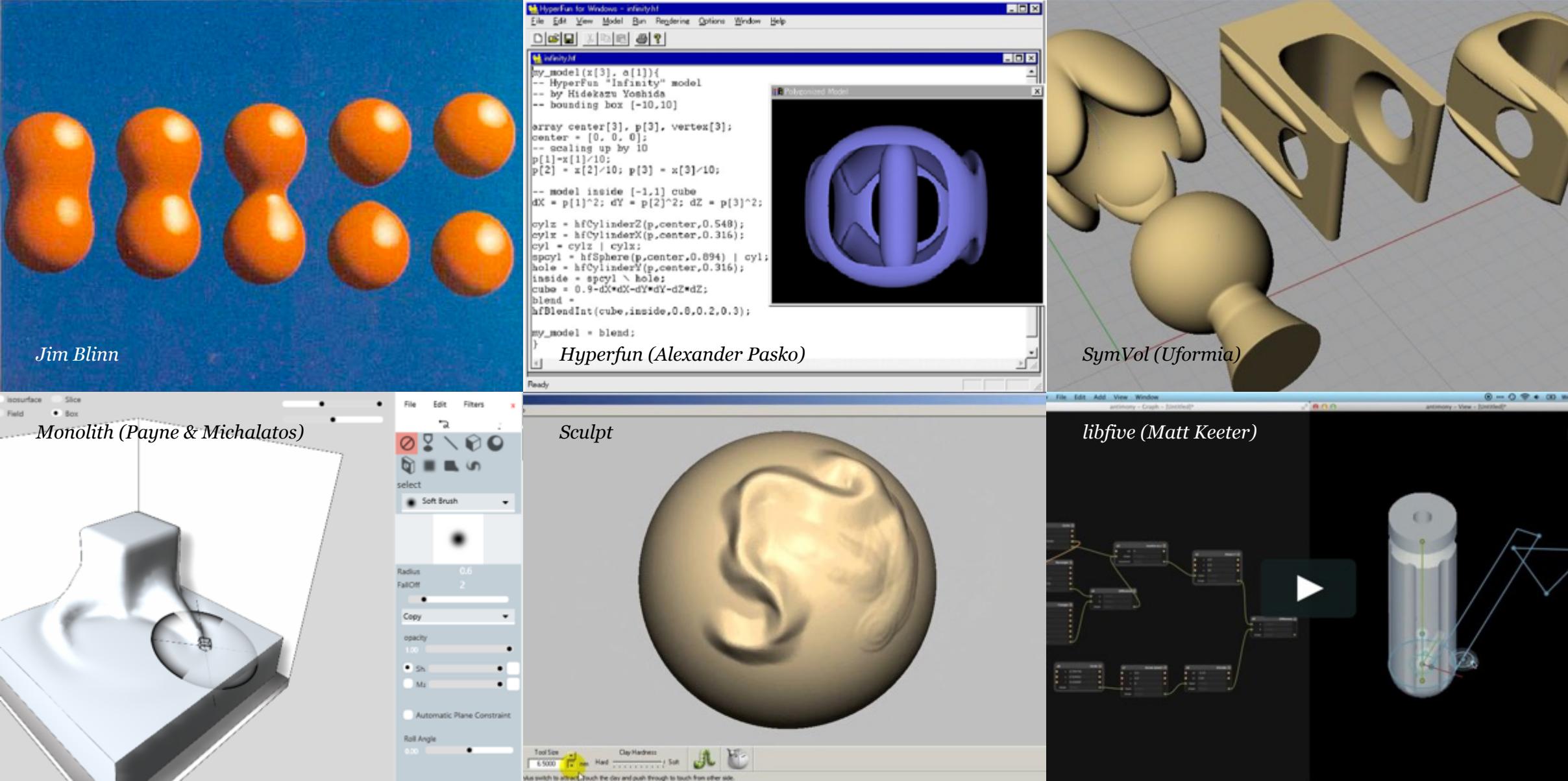


Boolean Operations



<https://wiki.mcneel.com/rhino/booleanfaq>

<https://wiki.mcneel.com/rhino/jbbooleanfix>



Signed Distance Functions (SDF)

$$f(x,y) = \sqrt{(cx-x)^2 + (cy-y)^2} - r$$

3.1	2.4	1.8	1.4	1.1	1.0	1.1	1.4	1.8	2.4	3.1	3.8	4.6	5.4	6.3
2.4	1.7	1.0	0.5	0.1	0.0	0.1	0.5	1.0	1.7	2.4	3.2	4.0	4.9	5.8
1.8	1.0	0.2	-0.4	-0.8	-1.0	-0.8	-0.4	0.2	1.0	1.8	2.7	3.6	4.5	5.5
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.3	3.3	4.2	5.2
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.1	2.1	3.1	4.1	5.1
1.0	0.0	-1.0	-2.0	-3.0	-4.0	-3.0	-2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.1	2.1	3.1	4.1	5.1
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.3	3.3	4.2	5.2
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.1	2.1	3.1	4.1	5.1
1.0	0.0	-1.0	-2.0	-3.0	-4.0	-3.0	-2.0	1.0	0.0	1.0	2.0	3.0	4.0	5.0
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.1	2.1	3.1	4.1	5.1
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.3	3.3	4.2	5.2
1.8	1.0	0.2	-0.4	-0.8	-1.0	-0.8	-0.4	0.2	1.0	1.8	2.7	3.6	4.5	5.5
2.4	1.7	1.0	0.5	0.1	0.0	0.1	0.5	1.0	1.7	2.4	3.2	4.0	4.9	5.8
3.1	2.4	1.8	1.4	1.1	1.0	1.1	1.4	1.8	2.4	3.1	3.8	4.6	5.4	6.3
3.8	3.2	2.7	2.3	2.1	2.0	2.1	2.3	2.7	3.2	3.8	4.5	5.2	6.0	6.8
4.6	4.1	3.6	3.3	3.1	3.0	3.1	3.3	3.6	4.1	4.6	5.2	5.9	6.6	7.4
5.4	4.9	4.5	4.2	4.1	4.0	4.1	4.2	4.5	4.9	5.4	6.0	6.6	7.3	8.0
6.3	5.8	5.5	5.2	5.1	5.0	5.1	5.2	5.5	5.8	6.3	6.8	7.4	8.0	8.7

$$u(x,y) = \min(f(x,y), g(x,y))$$

3.1	2.4	1.8	1.4	1.1	1.0	1.1	1.4	1.8	2.4	3.1	3.8	4.6	5.4	6.3
2.4	1.7	1.0	0.5	0.1	0.0	0.1	0.5	1.0	1.7	2.4	3.2	4.0	4.0	4.0
1.8	1.0	0.2	-0.4	-0.8	-1.0	-0.8	-0.4	0.2	1.0	1.8	2.7	3.0	3.0	3.0
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.0	2.0	2.0	2.0
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.0	1.0	1.0	1.0	2.0
1.0	0.0	-1.0	-2.0	-3.0	-4.0	-3.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.0	1.0	1.0	1.0	2.0
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.3	3.3	4.2	5.2
1.8	1.0	0.2	-0.4	-0.8	-1.0	-0.8	-0.4	0.2	1.0	1.8	2.7	3.6	4.5	5.5
2.4	1.7	1.0	0.5	0.1	0.0	0.1	0.5	1.0	1.7	2.4	3.2	4.1	4.9	5.8
3.1	2.4	1.8	1.4	1.1	1.0	1.1	1.4	1.8	2.4	3.1	3.8	4.6	5.4	6.3
3.8	3.2	2.7	2.3	2.1	2.0	2.1	2.3	2.7	3.2	3.8	4.5	5.2	6.0	6.8
4.6	4.1	3.6	3.3	3.1	3.0	3.1	3.3	3.6	4.1	4.6	5.2	5.9	6.6	7.4
5.4	4.9	4.5	4.2	4.1	4.0	4.1	4.2	4.5	4.9	5.4	6.0	6.6	7.3	8.0
6.3	5.8	5.5	5.2	5.1	5.0	5.1	5.2	5.5	5.8	6.3	6.8	7.4	8.0	8.7

$$s1(x,y) = \max(f(x,y), -g(x,y))$$

3.1	2.4	1.8	1.4	1.1	1.0	1.1	1.4	1.8	2.4	3.1	3.8	4.6	5.4	6.3
2.4	1.7	1.0	0.5	0.1	0.0	0.1	0.5	1.0	1.7	2.4	3.2	4.1	4.9	5.8
1.8	1.0	0.2	-0.4	-0.8	-1.0	-0.8	-0.4	0.2	1.0	1.8	2.7	3.6	4.5	5.5
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.3	3.3	4.2	5.2
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.1	2.1	3.1	4.1	5.1
1.0	0.0	-1.0	-2.0	-3.0	-4.0	-3.0	-2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1.1	0.1	-0.8	-1.8	-2.6	-3.0	-2.6	-1.8	0.8	0.1	1.0	1.0	1.0	1.0	2.0
1.4	0.5	-0.4	-1.2	-1.8	-2.0	-1.8	-1.2	0.4	0.5	1.4	2.3	3.3	4.2	5.2
1.8	1.0	0.2	-0.4	-0.8	-1.0	-0.8	-0.4	0.2	1.0	1.8	2.7	3.6	4.5	5.5
2.4	1.7	1.0	0.5	0.1	0.0	0.1	0.5	1.0	1.7	2.4	3.2	4.1	4.9	5.8
3.1	2.4	1.8	1.4	1.1	1.0	1.1	1.4	1.8	2.4	3.1	3.8	4.6	5.4	6.3
3.8	3.2	2.7	2.3	2.1	2.0	2.1	2.3	2.7	3.2	3.8	4.5	5.2	6.0	6.8
4.6	4.1	3.6	3.3	3.1	3.0	3.1	3.3	3.6	4.1	4.6	5.2	5.9	6.6	7.4
5.4	4.9	4.5	4.2	4.1	4.0	4.1	4.2	4.5	4.9	5.4	6.0	6.6	7.3	8.0
6.3	5.8	5.5	5.2	5.1	5.0	5.1	5.2	5.5	5.8	6.3	6.8	7.4	8.0	8.7

$$s(x,y) = \text{abs}(u(x,y)) - d/2$$

2.1	1.4	0.8	0.4	0.1	0.0	0.1	0.4	0.8	1.4	2.1	2.8	3.6	4.0	4.0
1.4	0.7	0.0	-0.5	-0.9	-1.0	-0.9	-0.5	0.0	0.7	1.4	2.2	3.0	3.0	3.0
0.8	0.0	-0.8	-0.6	-0.2	0.0	-0.2	-0.6	-0.8	0.0	1.7	2.0	2.0	2.0	2.0
0.4	-0.5	-0.6	0.2	0.8	1.0	0.8	0.2	-0.6	-0.5	0.4	1.0	1.0	1.0	1.0
0.1	-0.9	-0.2	0.8	1.6	2.0	1.6	0.8	-0.2	-0.9	0.0	0.0	0.0	0.0	1.0
0.0	-1.0	0.0	1.0	2.0	3.0	2.0	1.0	0.0	-1.0	-1.0	-1.0	-1.0	0.0	1.0
0.1	-0.9	-0.2	0.8	1.6	2.0	1.6	0.8	0.0	0.0	0.0	0.0	0.0	-1.0	0.0
0.4	-0.5	-0.6	0.2	0.8	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
0.8	0.0	-0.8	-0.6	-0.2	0.0	1.0	2.0	2.0	1.0	0.0	-1.0	0.0	0.0	0.0
1.4	0.7	0.0	-0.5	-1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2.1	1.4	0.8	0.0	-1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2.8	2.0	1.0	0.0	-1.0	-1.0	-1.0	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
3.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

$$g(x,y) = \max(\text{abs}(cx-x)-a/2, \text{abs}(cy-y)-b/2)$$

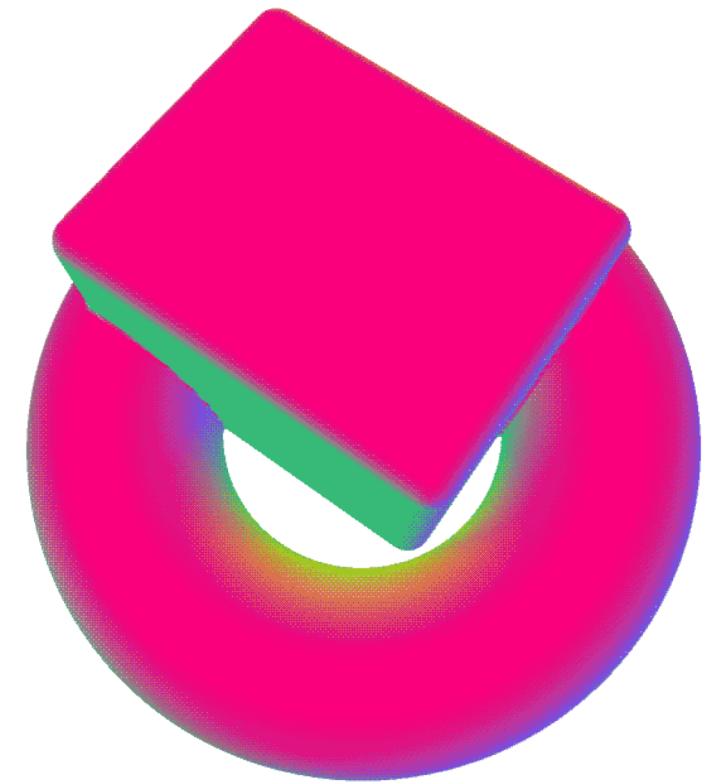
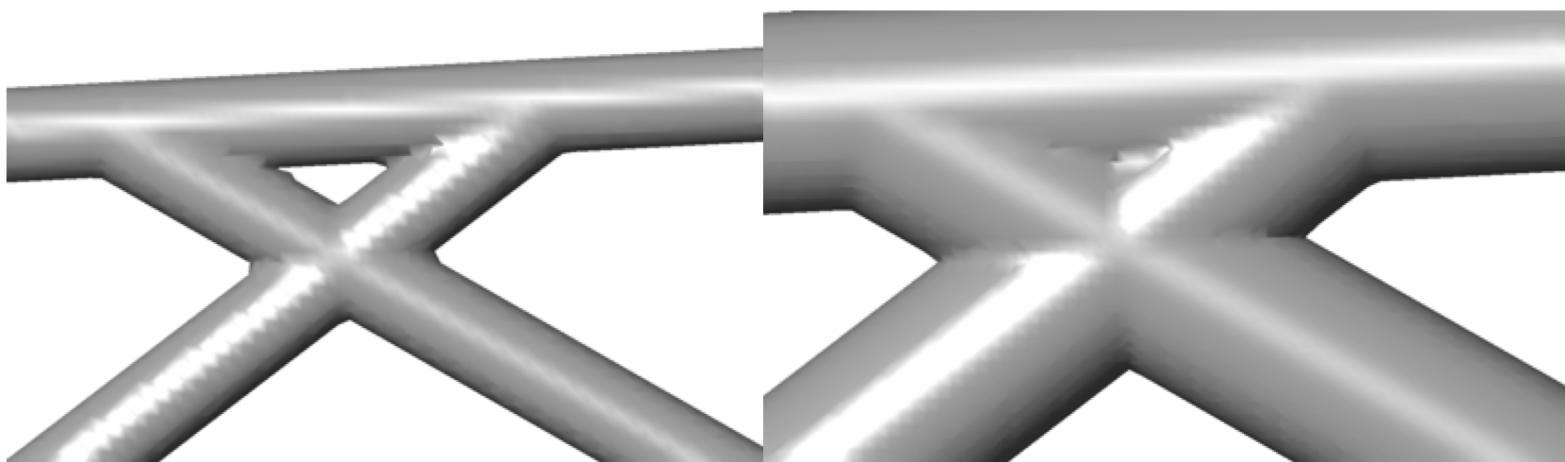
$$i(x,y) = \max(f(x,y), g(x,y))$$

$$s2(x,y) = \max(-f(x,y), g(x,y))$$

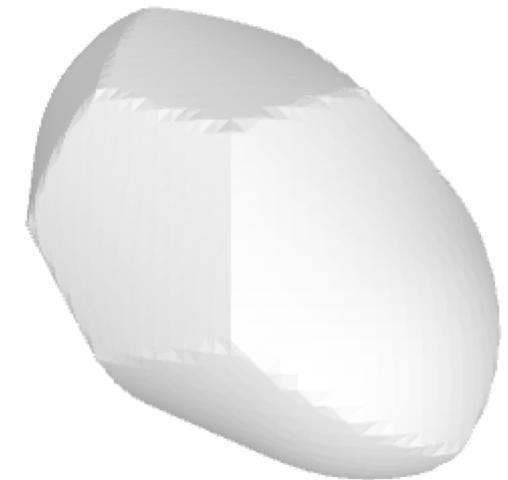
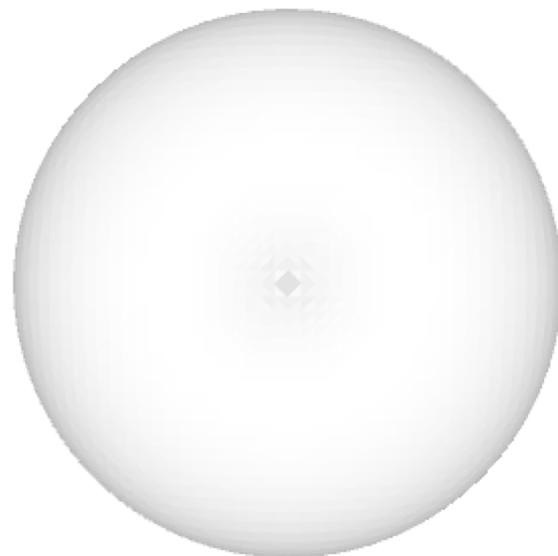
$$b(x,y) = u(x,y) - e^{2*0.25/r}$$

$$e = \max(r - \text{abs}(f(x,y) - g(x,y)), o)$$

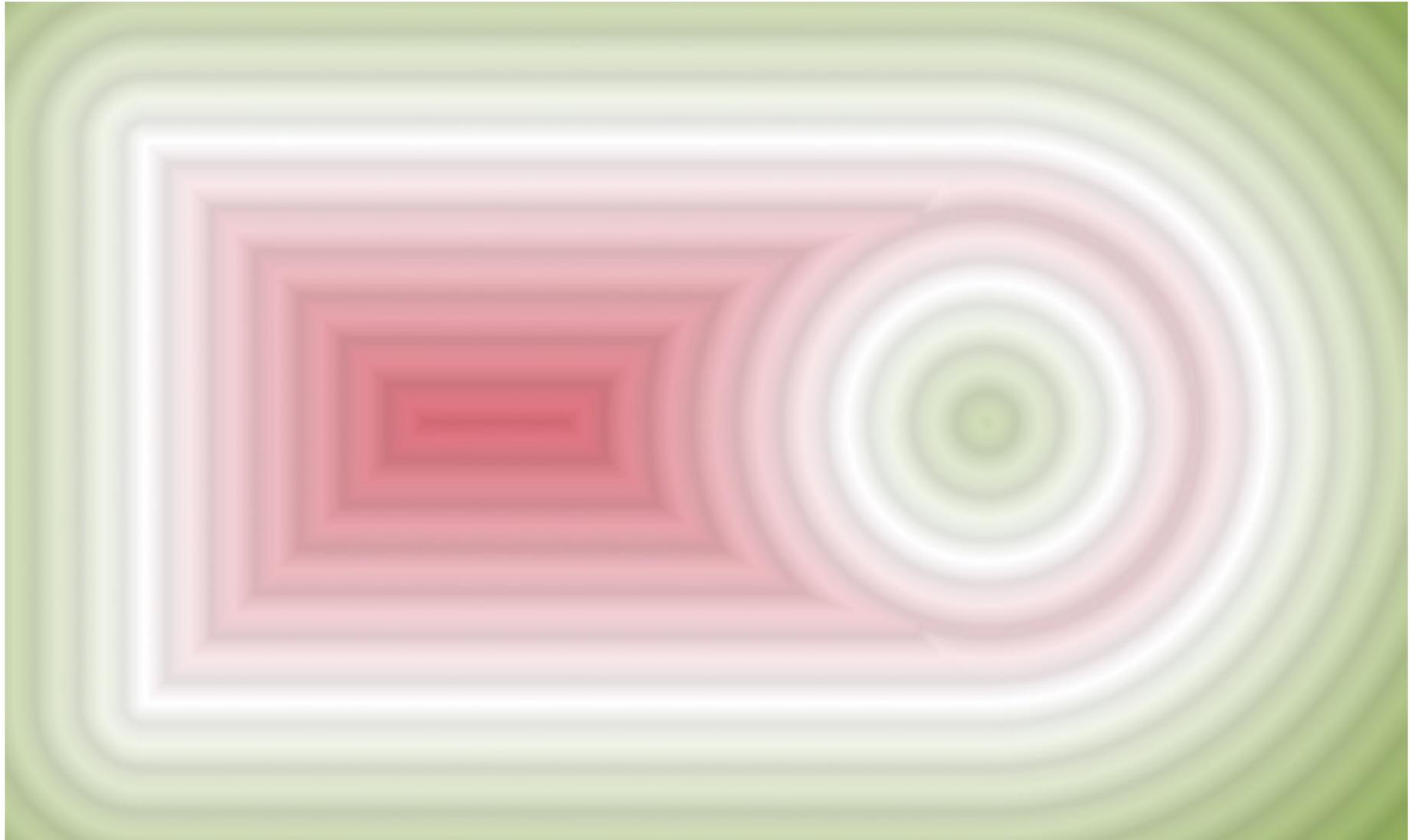
Topology



Continuous Blends

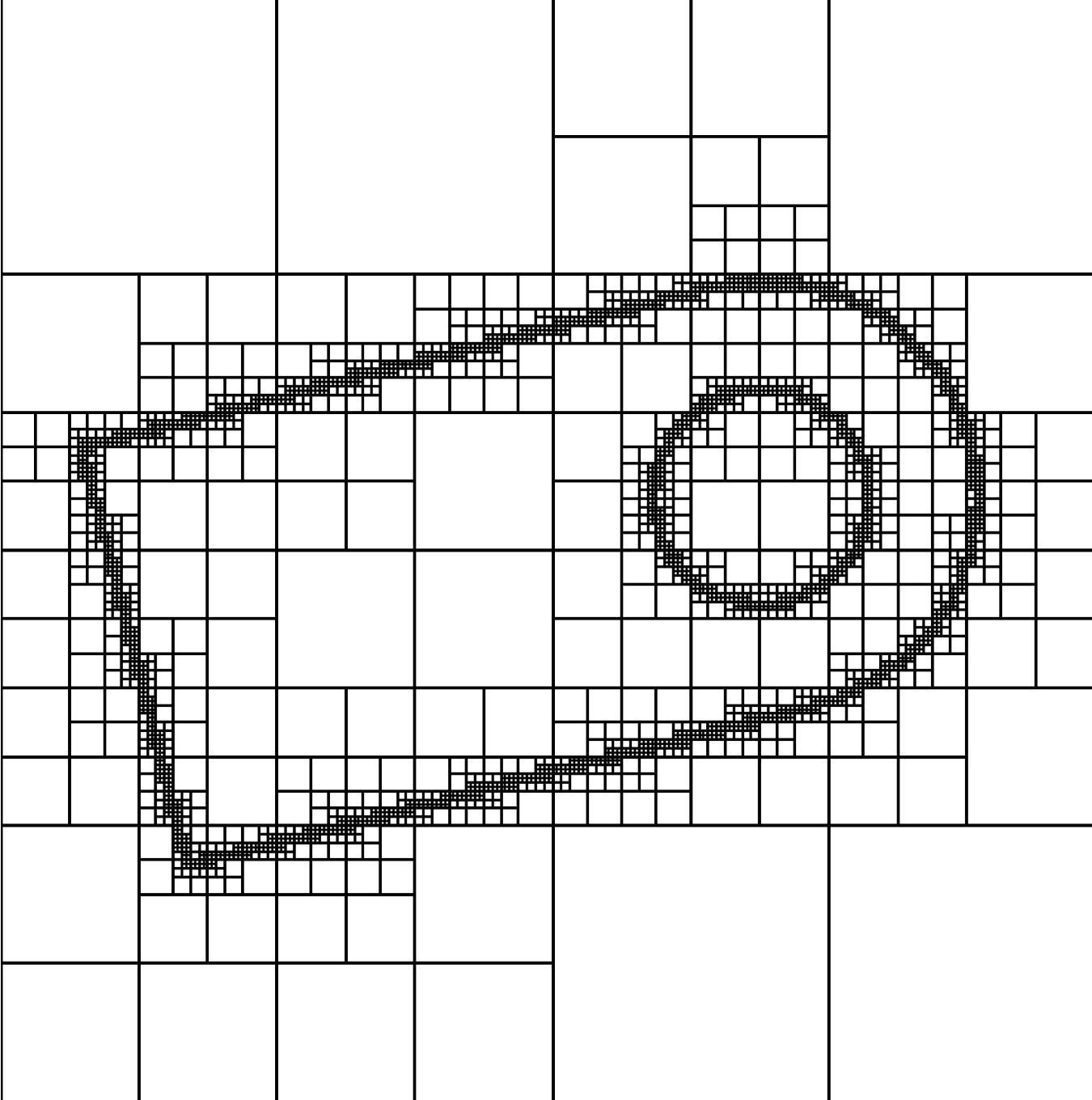


Signed Distance Function

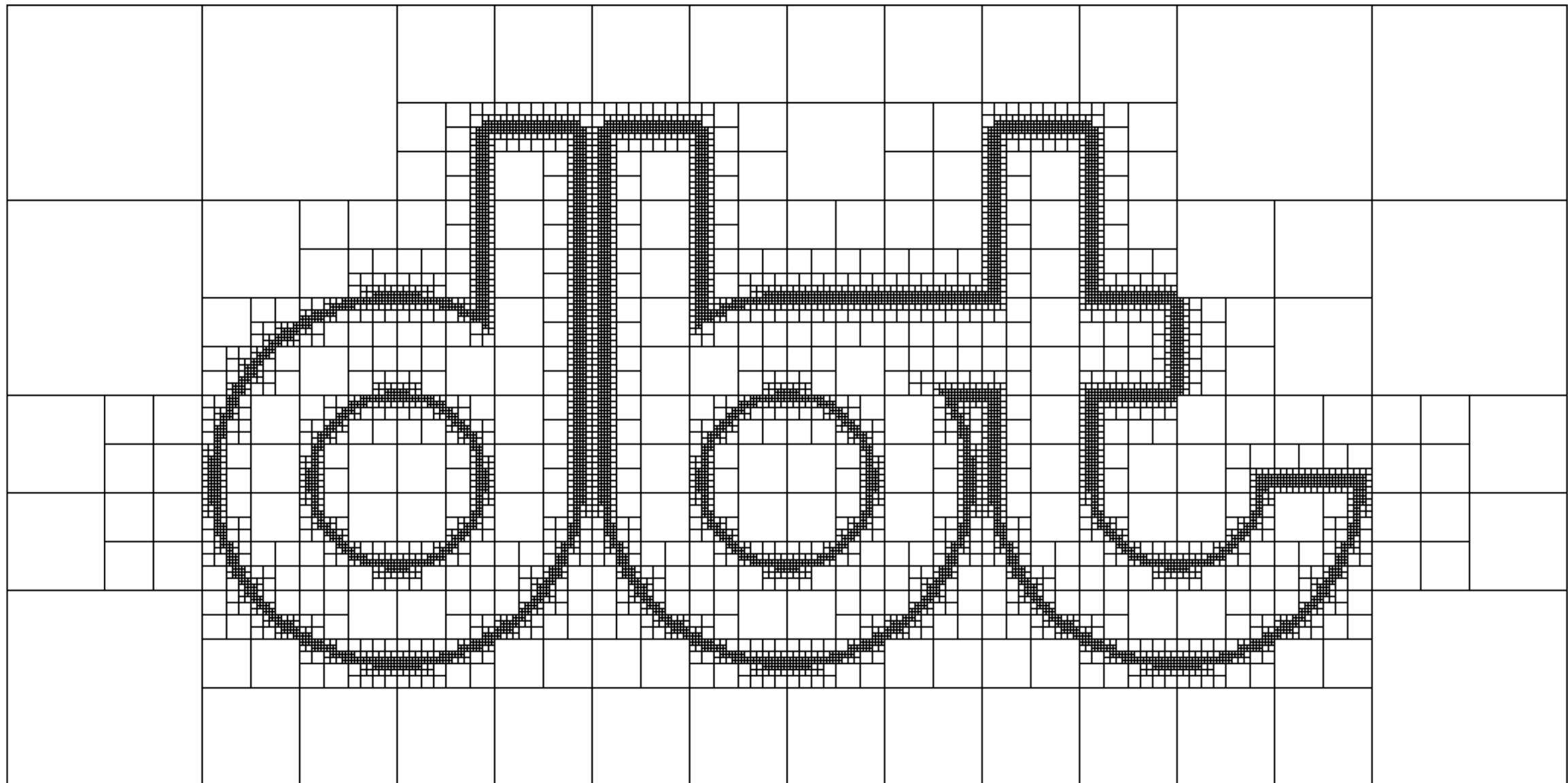


dotG

Quadtree



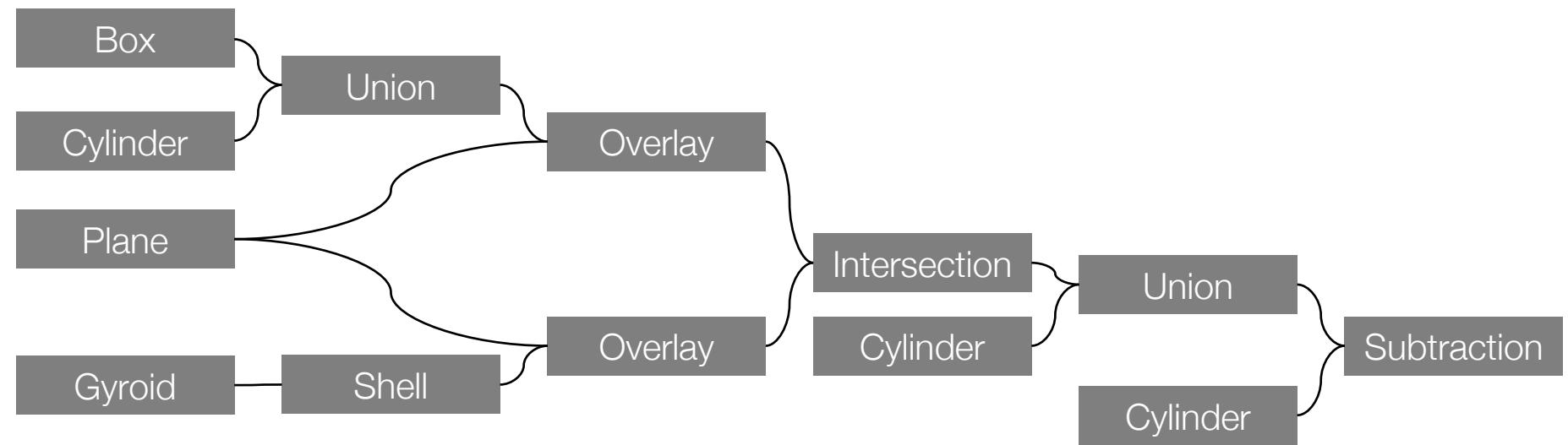




Wrench
by Frustum



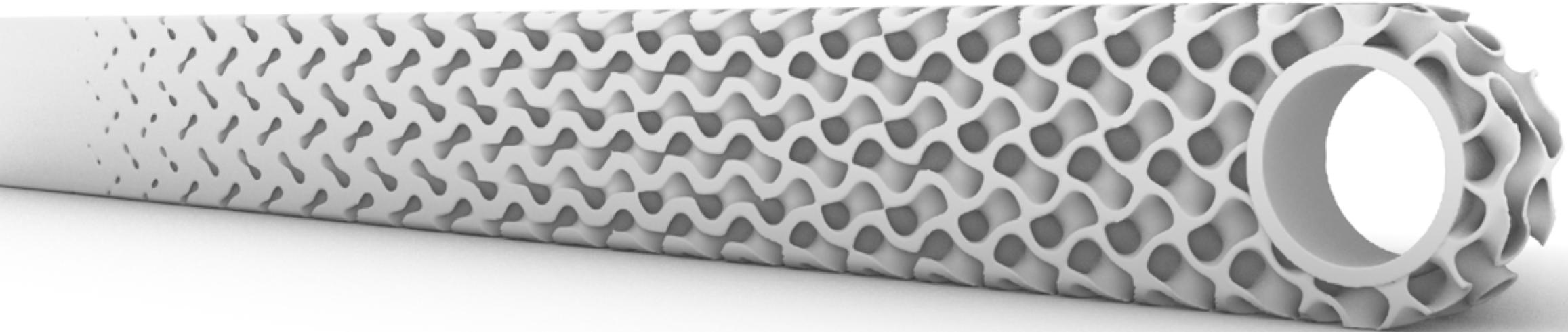
Constructive Solid Geometry Tree (CSG)



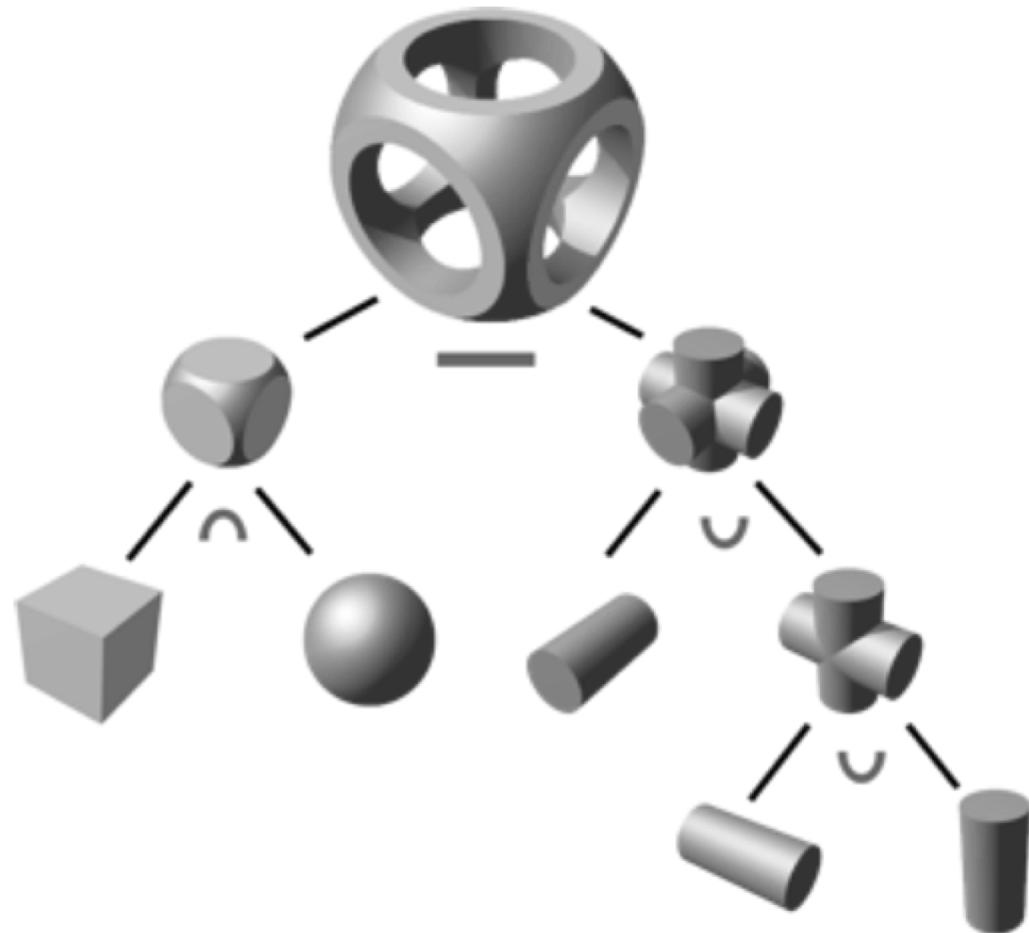
Code

```
shaft = VolBox(Box(Frame.worldXY(), 250, 30, 10), 1.5)
cyl_plane = Plane((125,0,0),(0,0,1))
roundcap = VolCylinder(Cylinder(Circle(cyl_plane, 15), 10))
handle = Union(shaft, roundcap)
gyroid = TPMS(tpmstype='Gyroid', wavelength=5.0)
shell = Shell(gyroid, 2.0, 0.5)
ol_plane = VolPlane(Plane((0,0,0), (1,0,0)))
thicken_tpms = Overlay(shell, ol_plane, 0.005)
taper = Overlay(handle, ol_plane, -0.01)
porous_handle = Intersection(thicken_tpms, taper)
solid_outer = VolCylinder(Cylinder(Circle(cyl_plane, 12), 13))
void_inner = VolCylinder(Cylinder(Circle(cyl_plane, 10), 20))
hole_reinforce = Union(porous_handle, solid_outer)
wrench = Subtraction(hole_reinforce, void_inner)
```

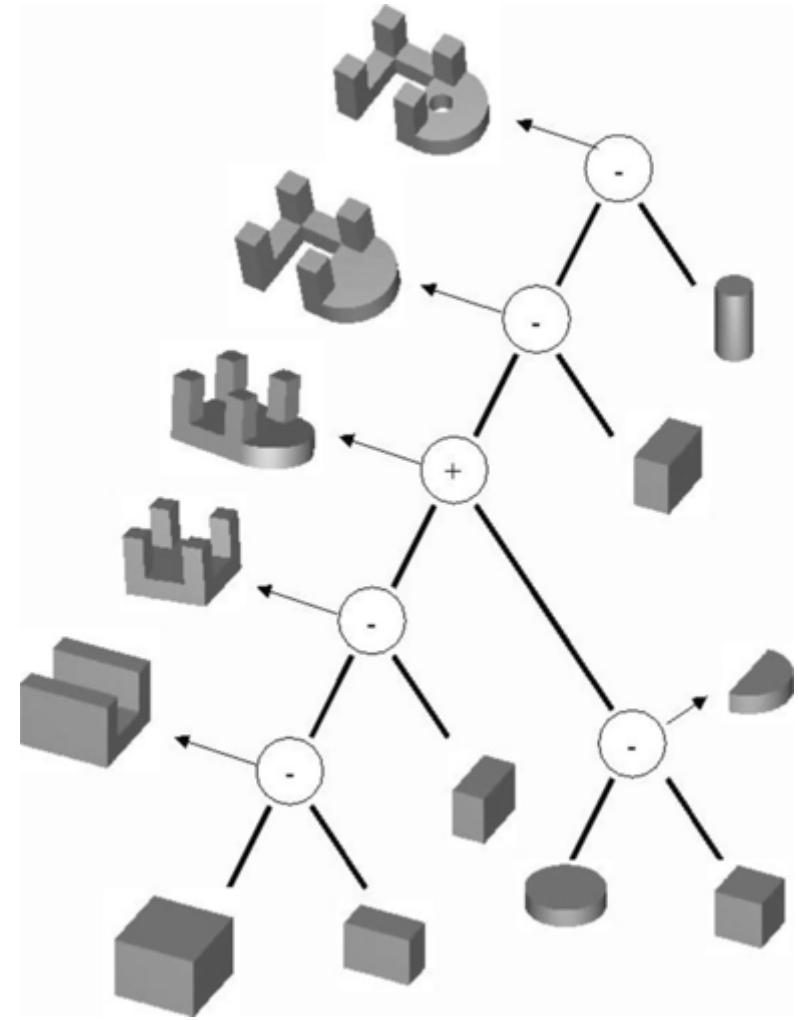
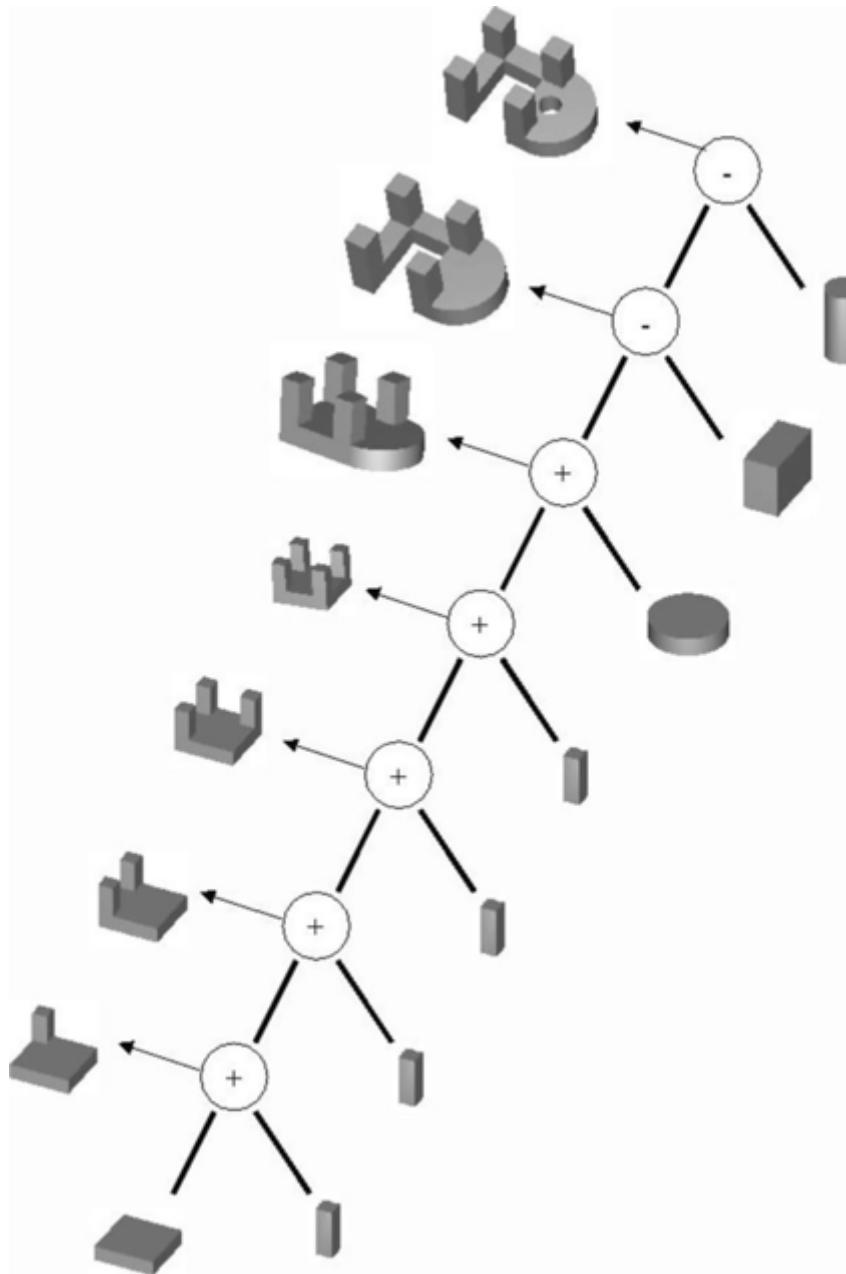
Wrench in
compas_vol

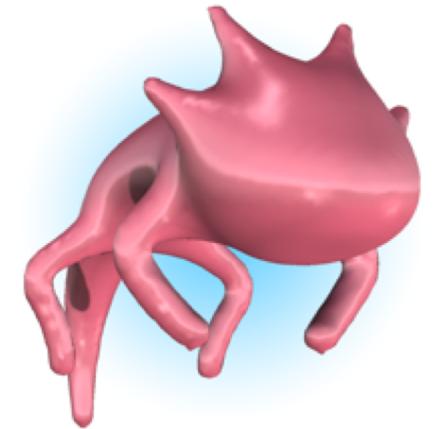


Constructive Solid Geometry (CSG)



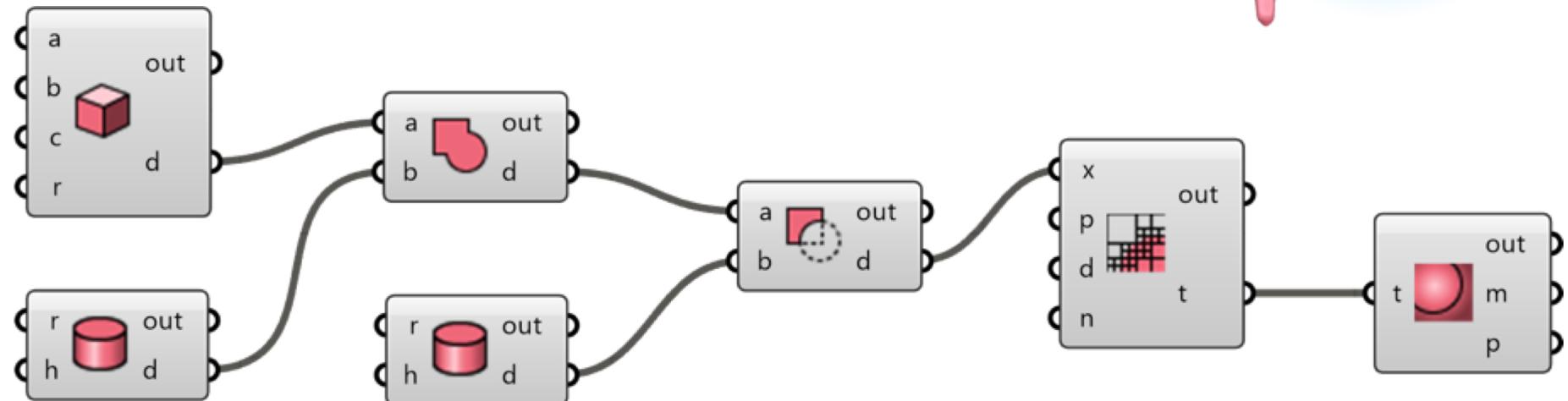
Tree Balancing





Axolotl

food4rhino.com/app/axolotl

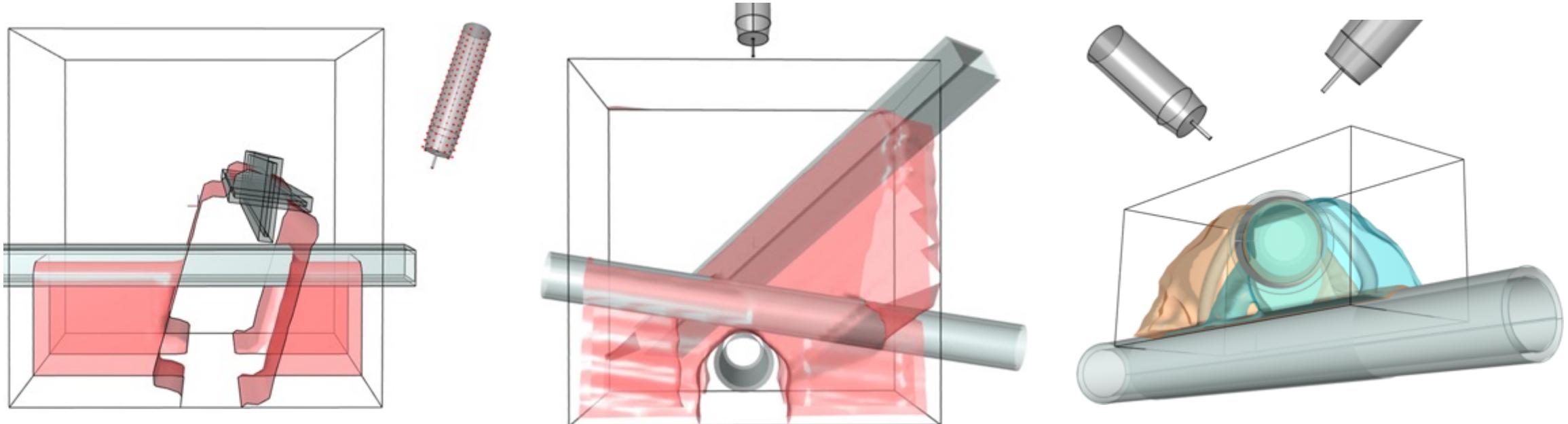


CSG tree

Octree

Isosurface

Numerical Sculpting



Mitropoulou, Ioanna, Inés Ariza, Mathias Bernhard, Benjamin Dillenburger, Fabio Gramazio, and Matthias Kohler. 2019. "Numerical Sculpting - Volumetric Modelling Tools for in Place Spatial Additive Manufacturing." In *DMSB - Impact: Design with All Senses*. Springer.