Computational Design + Fabrication: 3D Design

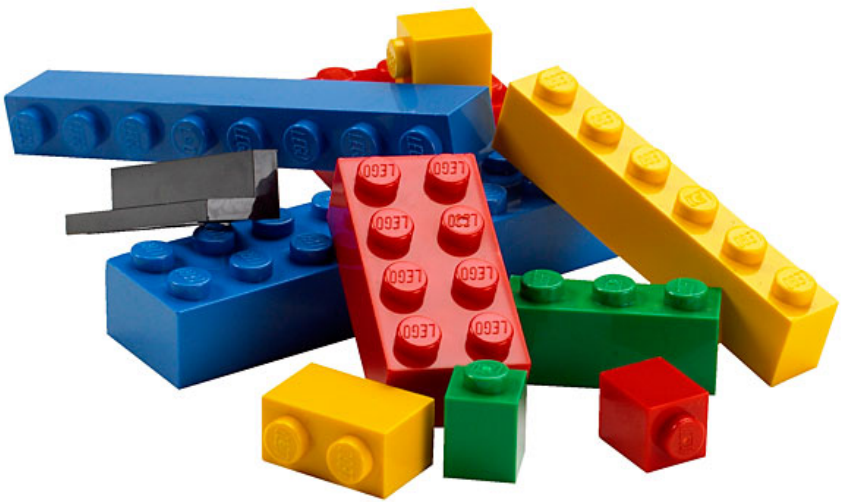
Jonathan Bachrach

EECS UC Berkeley

September 24, 2015

Today

- News
- 3D Design
- 3D Geometry
- Paper Review



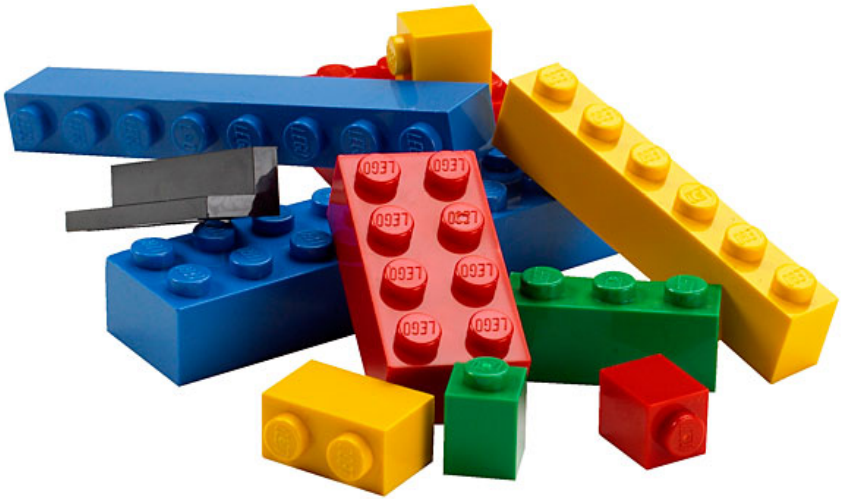
News

- lab 2 due thursday
- section tomorrow 2-3p in soda 373 ???
- jacobs 3d printer training
  `https://bcourses.berkeley.edu/courses/1353091`

Shop Policy Violations

- safety glasses are to be worn all the time when using diwire machine
- students are responsible for supplying their own safety glasses
- students must put away all tools when done

- bricks and legos
- 3d printing



- complexity is free
- less waste
- no assembly required
- less skill

3D Design Today

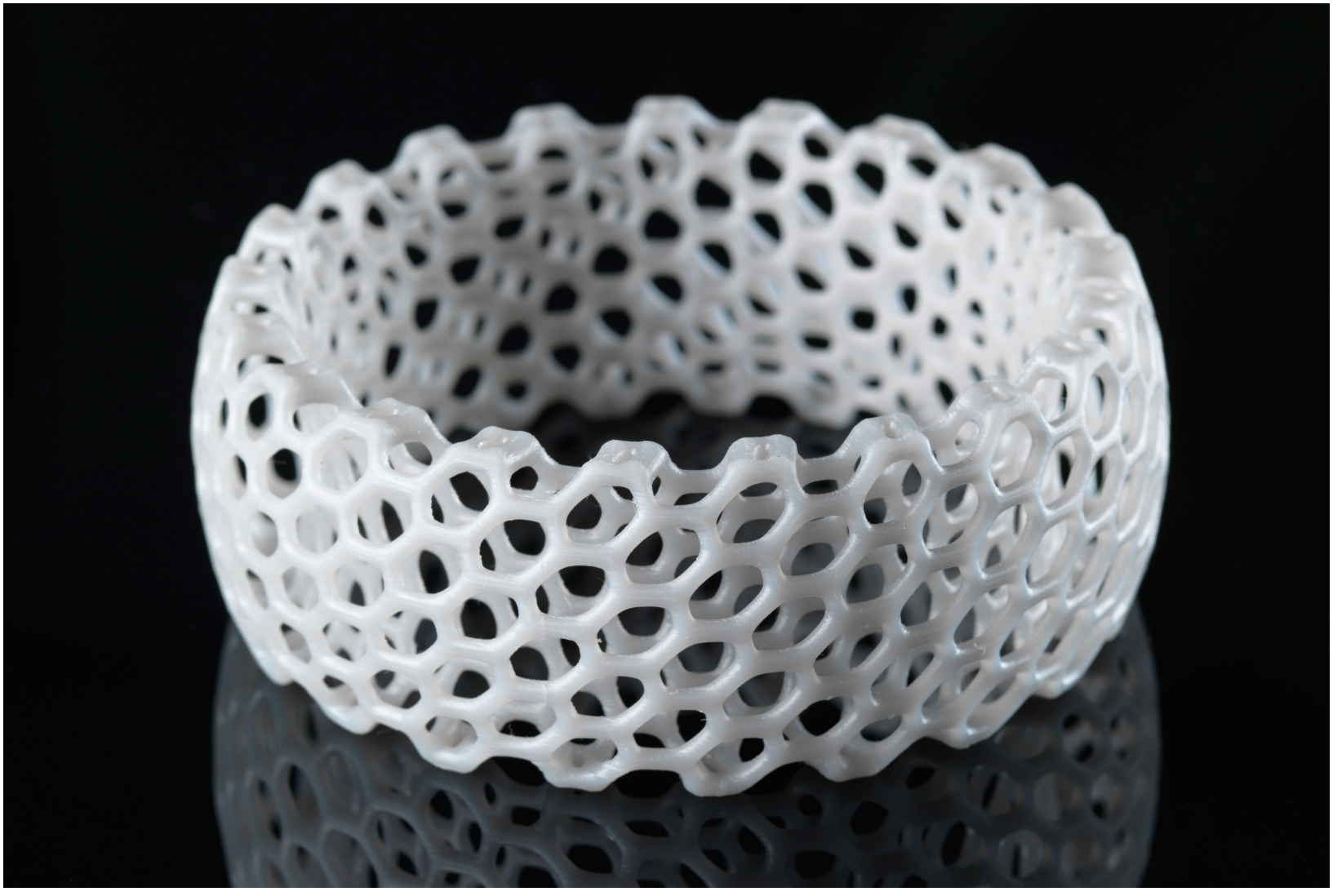
- examples
- solid modeling
- digifab support

3D Printed Lamp

by 74fdc.wordpress

Atom 3D Printed Guitar

by cubify

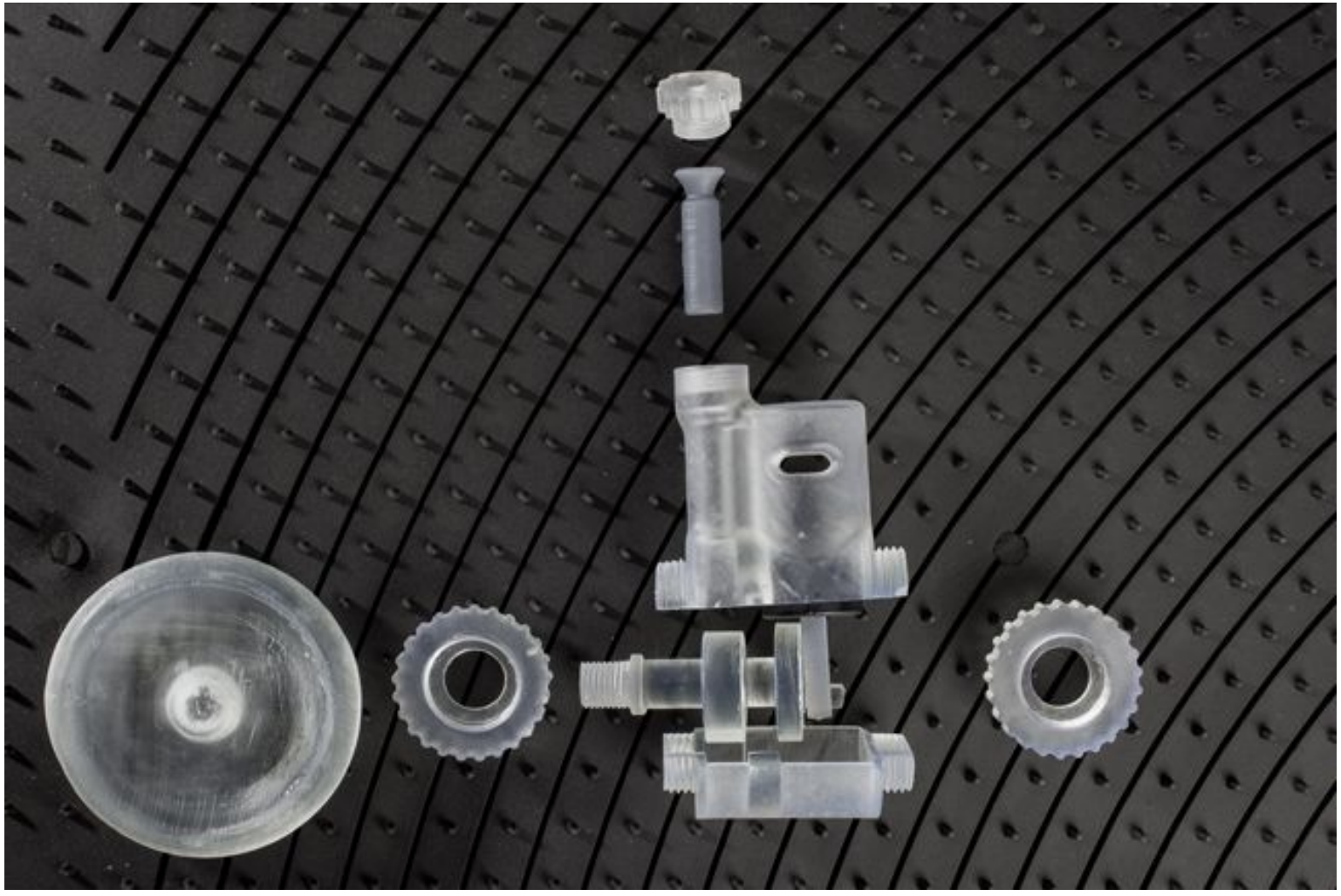


by Nervous Systems

Insect Model

by Klaus Leitl

Intake Manifold



by Kevin Gautier for Formula SAE

Orthodontic Retainer

by Michael Lyons et al via FormLabs

Printed Fabric

by Designer Jiri Evenhuis, in collaboration with Janne Kyttanen of Freedom of Creation

Sketch Figurine

by Crayon Creations

Hand-made Camera Lens

by yukiSUZUKI

by Chinese Anonymous

by Anonymous

by Foodini

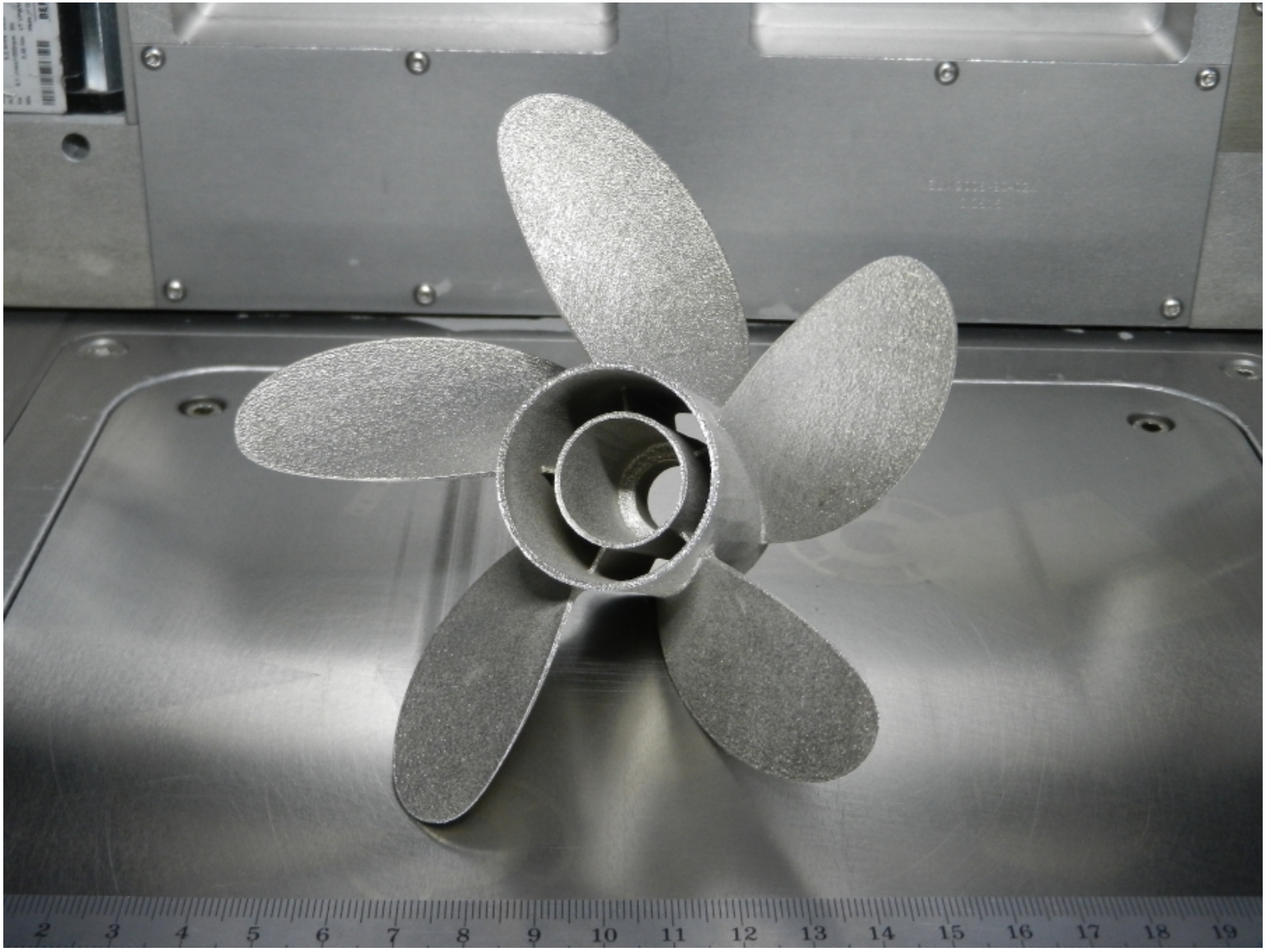
Tortoise Prosthesis

by Roger Henry

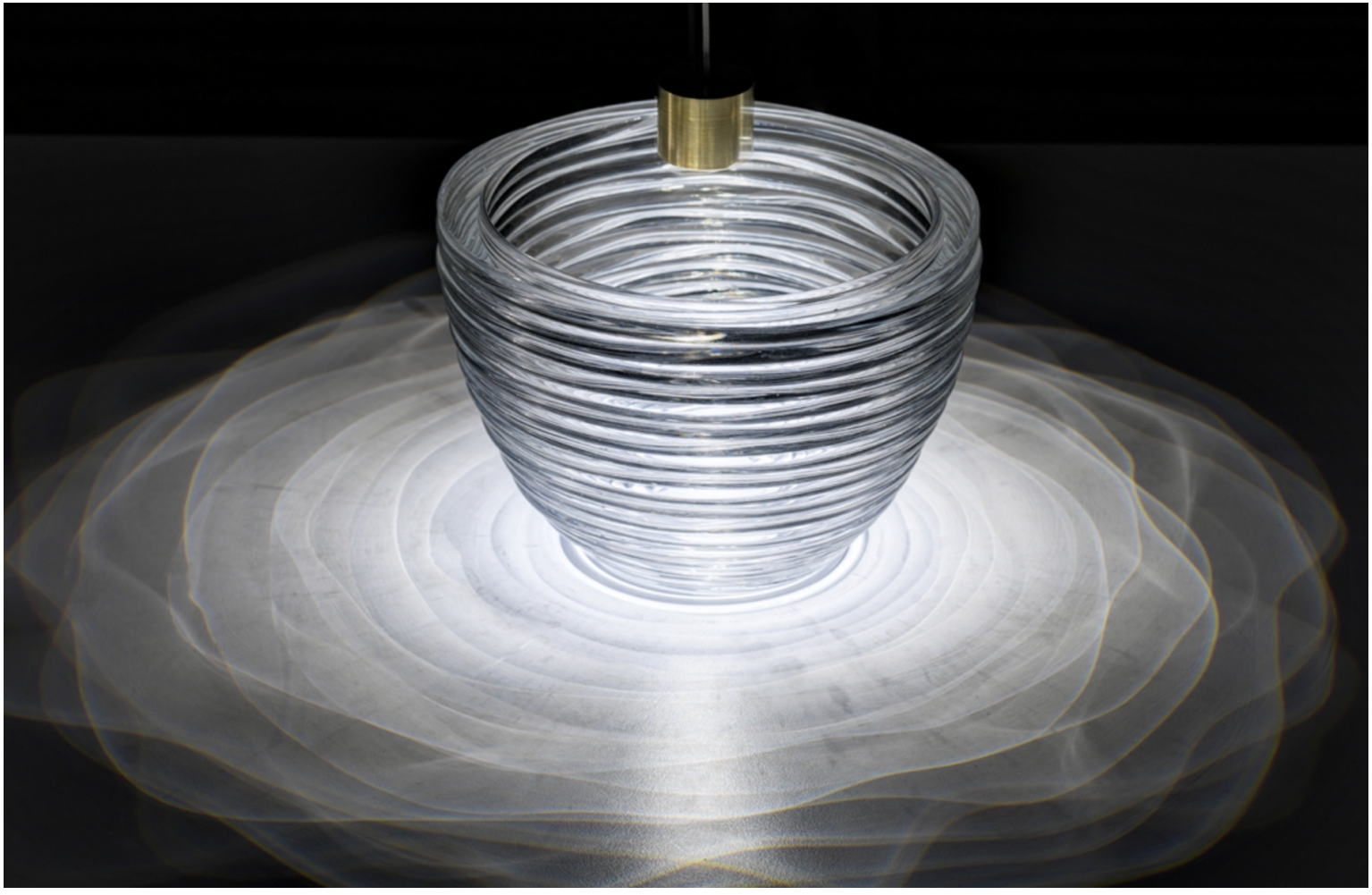
by TBWA/Hakuhodo

Lamp Phone Connector

by Anonymous



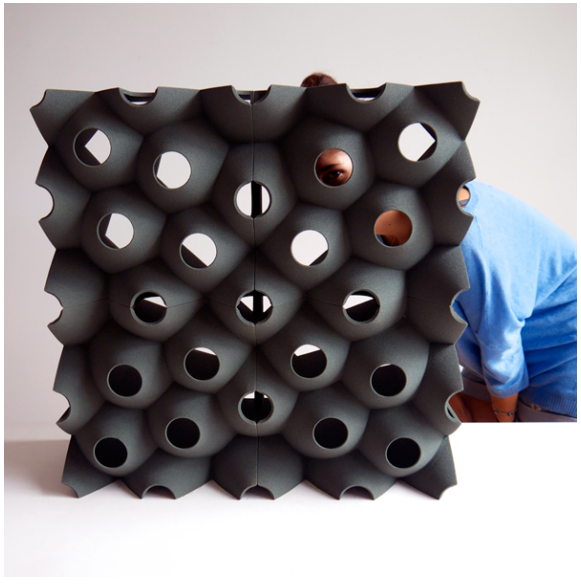
by protoshape



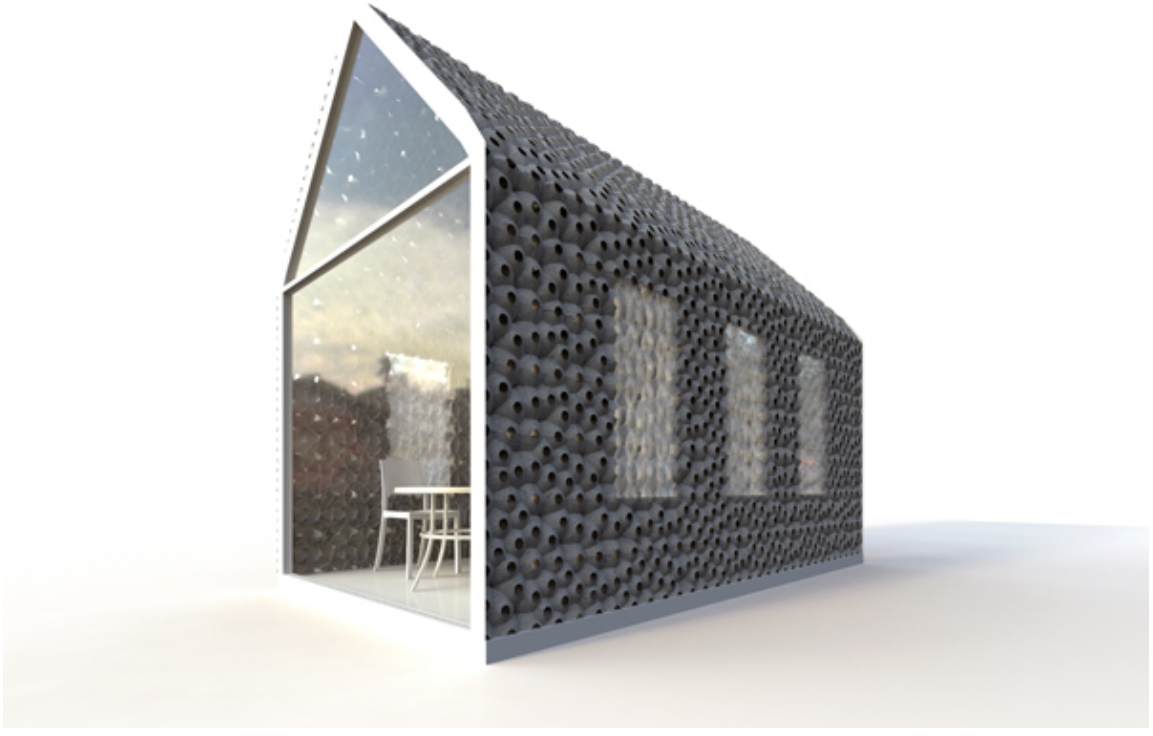
by mediated matter group at mit – photo by andy ryan

by Gramazio and Kohler + ETH Zurich

Tiled 3DP Bricks



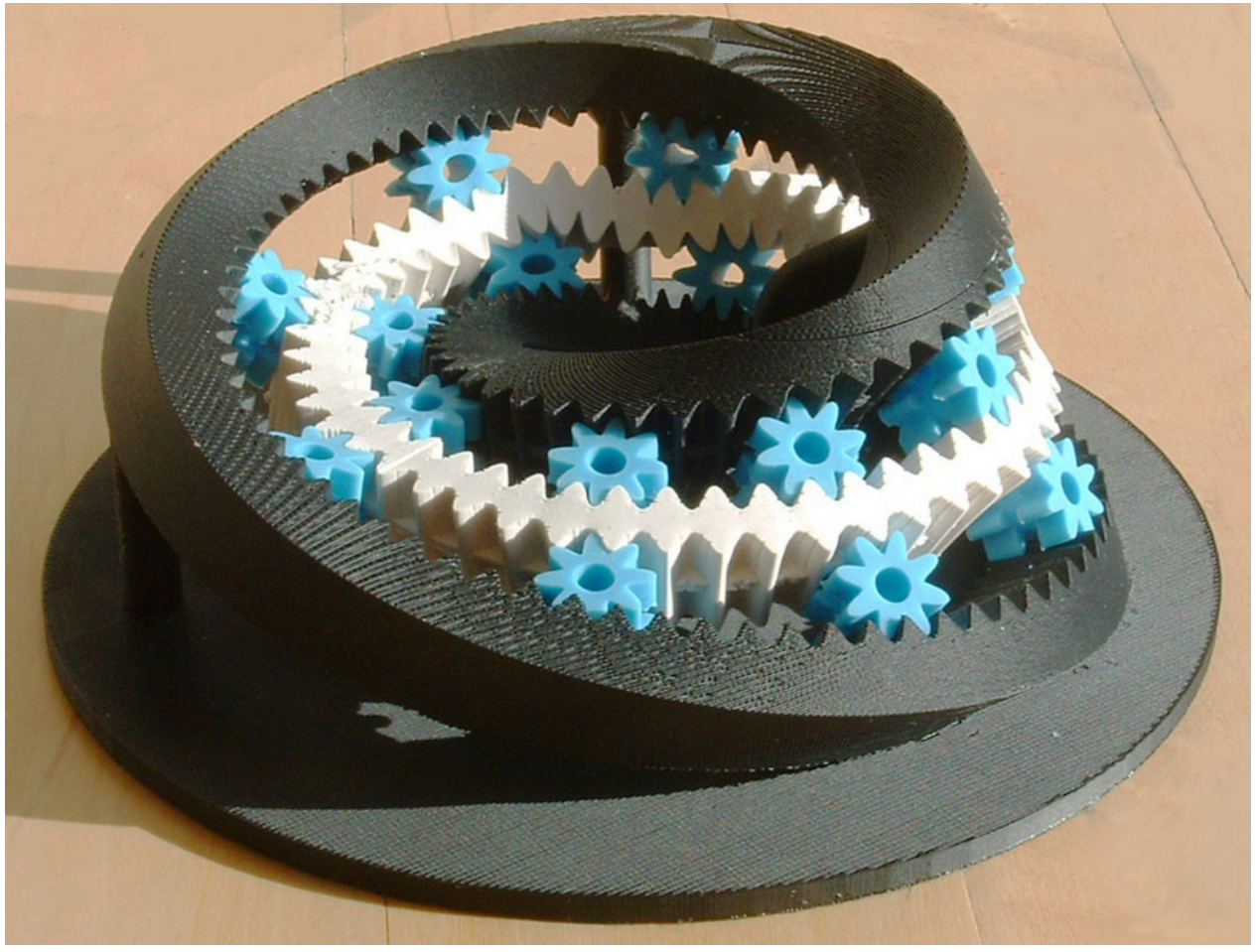
picoroco block in sand



shed out of blocks

by Emerging Objects

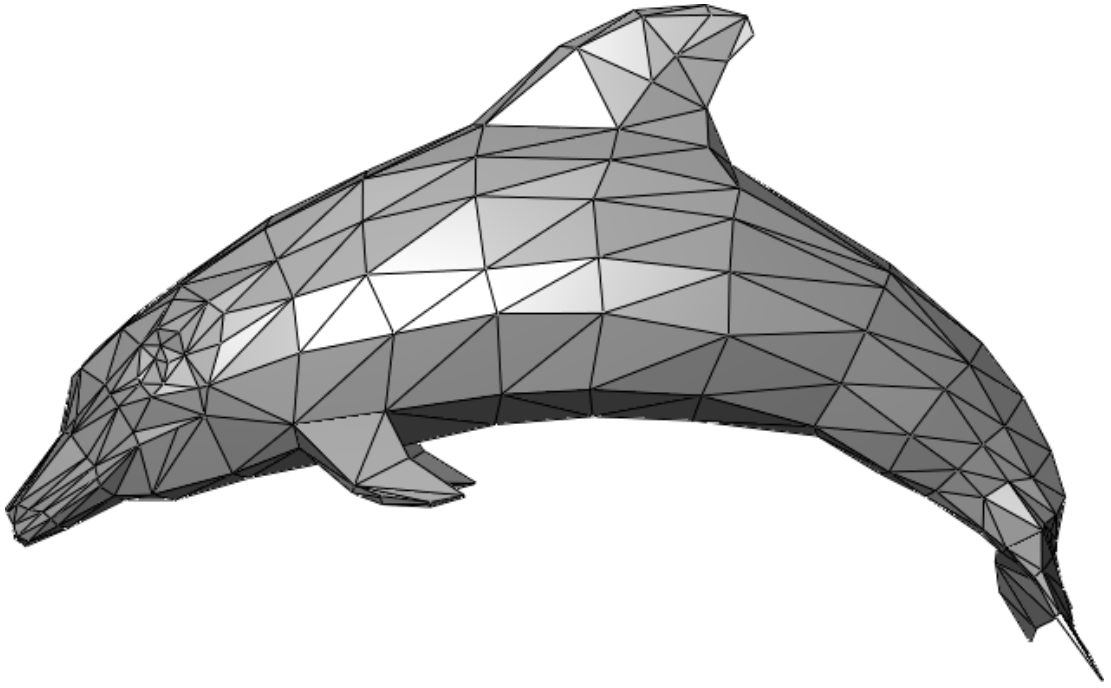
- digital models of physical objects
- represent interiors of objects



Aaron Hoover

Boundary Representations (B-Reps)

- represents shapes using their limits
- using connected surface elements
- boundary between solid and non-solid



Pros

- flexible
- wide spread

Cons

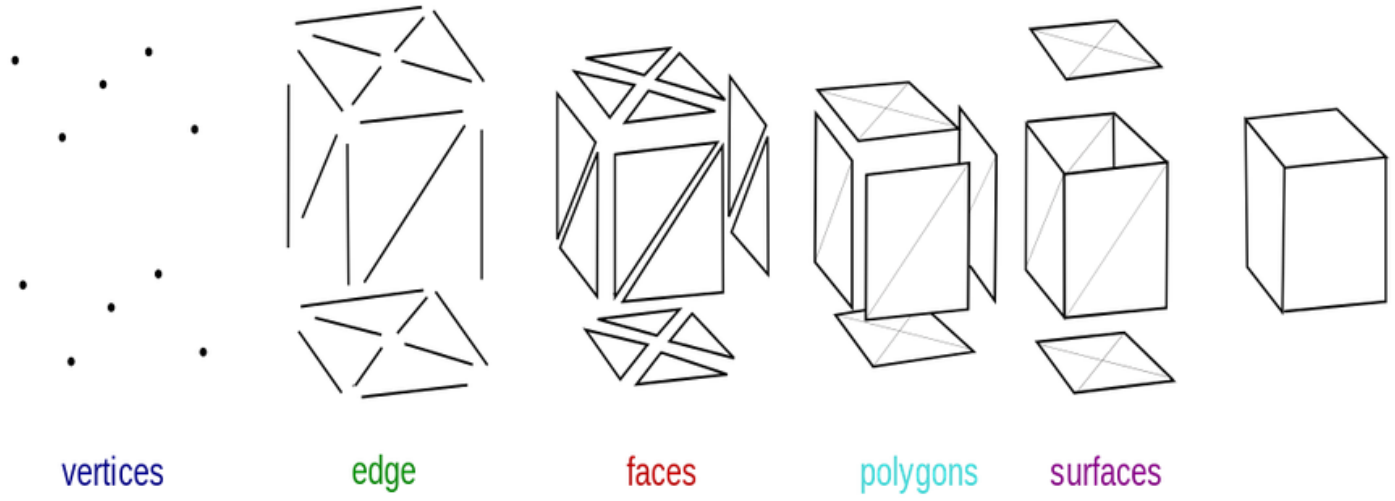
- ill-defined solids (not closed under ops)

main topological components are:

- face – bounded portion of a surface
- edge – bounded piece of a curve
- vertex – lies at a point

other elements are:

- shell – connected set of faces
- loop – circuit of edges bounding a face



Face-Vertex Representation

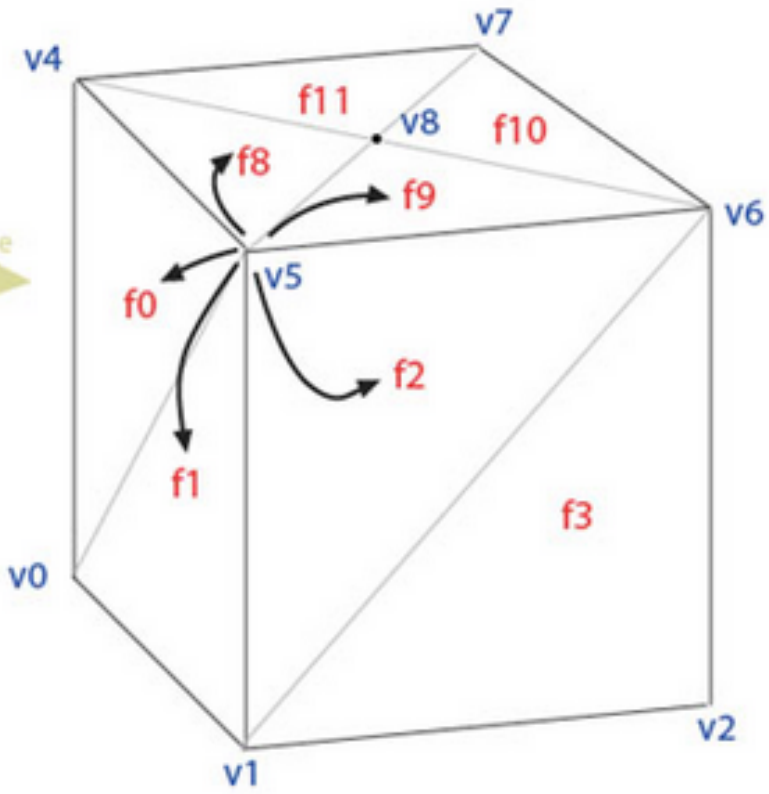
Face List

f0	v0 v4 v5
f1	v0 v5 v1
f2	v1 v5 v6
f3	v1 v6 v2
f4	v2 v6 v7
f5	v2 v7 v3
f6	v3 v7 v4
f7	v3 v4 v0
f8	v8 v5 v4
f9	v8 v6 v5
f10	v8 v7 v6
f11	v8 v4 v7
f12	v9 v5 v4
f13	v9 v6 v5
f14	v9 v7 v6
f15	v9 v4 v7

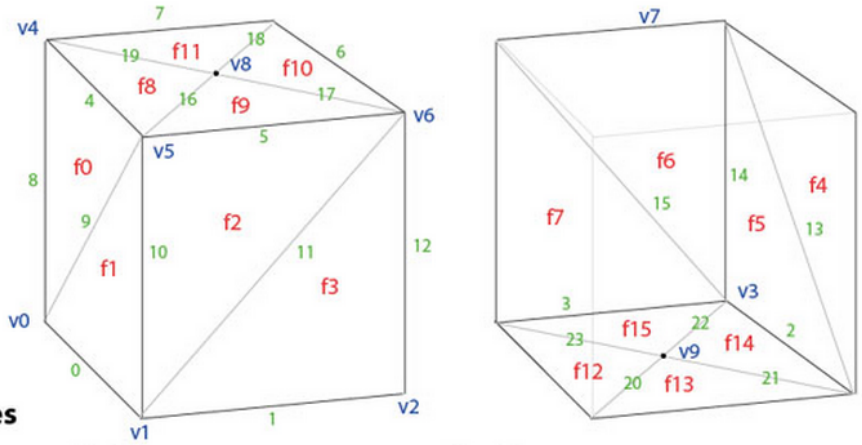
Vertex List

v0	0,0,0	f0 f1 f12 f15 f7
v1	1,0,0	f2 f3 f13 f12 f1
v2	1,1,0	f4 f5 f14 f13 f3
v3	0,1,0	f6 f7 f15 f14 f5
v4	0,0,1	f6 f7 f0 f8 f11
v5	1,0,1	f0 f1 f2 f9 f8
v6	1,1,1	f2 f3 f4 f10 f9
v7	0,1,1	f4 f5 f6 f11 f10
v8	.5,.5,0	f8 f9 f10 f11
v9	.5,.5,1	f12 13 14 15

example



Winged-Edge Representation



Winged-Edge Meshes

Face List

f0	4 8 9
f1	0 10 9
f2	5 10 11
f3	1 12 11
f4	6 12 13
f5	2 14 13
f6	7 14 15
f7	3 8 15
f8	4 16 19
f9	5 17 16
f10	6 18 17
f11	7 19 18
f12	0 23 20
f13	1 20 21
f14	2 21 22
f15	3 22 23

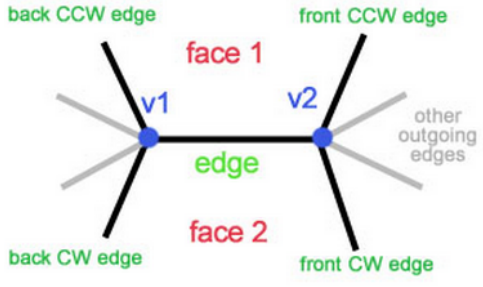
Edge List

e0	v0 v1	f1 f12	9 23 10 20
e1	v1 v2	f3 f13	11 20 12 21
e2	v2 v3	f5 f14	13 21 14 22
e3	v3 v0	f7 f15	15 22 8 23
e4	v4 v5	f0 f8	19 8 16 9
e5	v5 v6	f2 f9	16 10 17 11
e6	v6 v7	f4 f10	17 12 18 13
e7	v7 v4	f6 f11	18 14 19 15
e8	v0 v4	f7 f0	3 9 7 4
e9	v0 v5	f0 f1	8 0 4 10
e10	v1 v5	f1 f2	0 11 9 5
e11	v1 v6	f2 f3	10 1 5 12
e12	v2 v6	f3 f4	1 13 11 6
e13	v2 v7	f4 f5	12 2 6 14
e14	v3 v7	f5 f6	2 15 13 7
e15	v3 v4	f6 f7	14 3 7 15
e16	v5 v8	f8 f9	4 5 19 17
e17	v6 v8	f9 f10	5 6 16 18
e18	v7 v8	f10 f11	6 7 17 19
e19	v4 v8	f11 f8	7 4 18 16
e20	v1 v9	f12 f13	0 1 23 21
e21	v2 v9	f13 f14	1 2 20 22
e22	v3 v9	f14 f15	2 3 21 23
e23	v0 v9	f15 f12	3 0 22 20

Vertex List

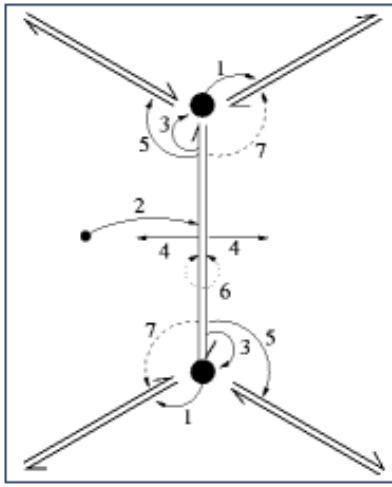
v0	0,0,0	8 9 0 23 3
v1	1,0,0	10 11 1 20 0
v2	1,1,0	12 13 2 21 1
v3	0,1,0	14 15 3 22 2
v4	0,0,1	8 15 7 19 4
v5	1,0,1	10 9 4 16 5
v6	1,1,1	12 11 5 17 6
v7	0,1,1	14 13 6 18 7
v8	.5,.5,0	16 17 18 19
v9	.5,.5,1	20 21 22 23

Figure 4. Winged-edge meshes



Winged Edge Structure

Half-Edge Representation



- Each **vertex** references one outgoing halfedge, i.e. a halfedge that starts at this vertex (1).
- Each **face** references one of the halfedges bounding it (2).
- Each **halfedge** provides a handle to
  - the vertex it points to (3),
  - the face it belongs to (4)
  - the next halfedge inside the face (ordered counter-clockwise) (5),
  - the opposite halfedge (6),
  - (optionally: the previous halfedge in the face (7)).

www.openmesh.org

- represent interior

dr. stefan wirth

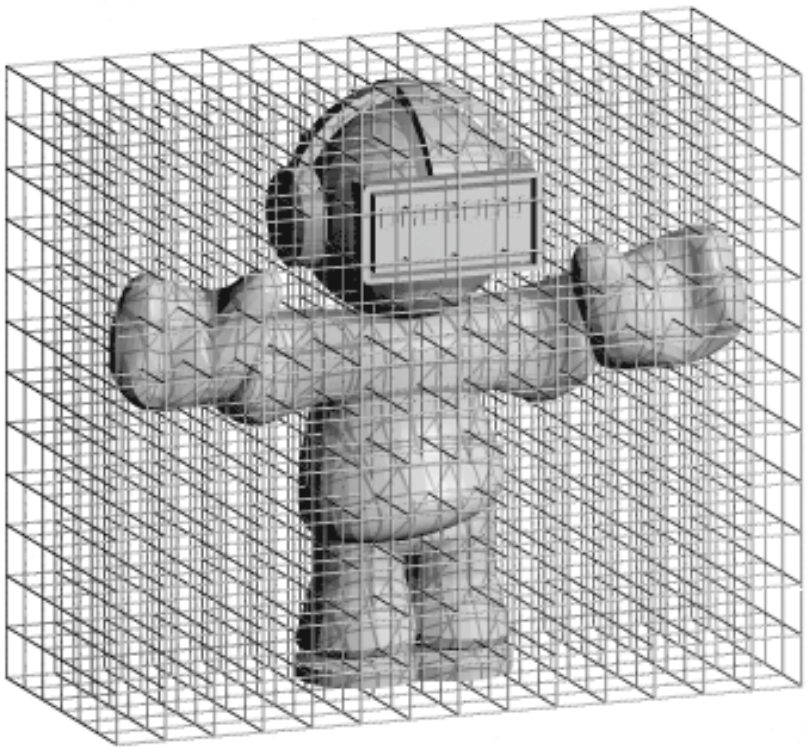
Why Volumetric Representations?

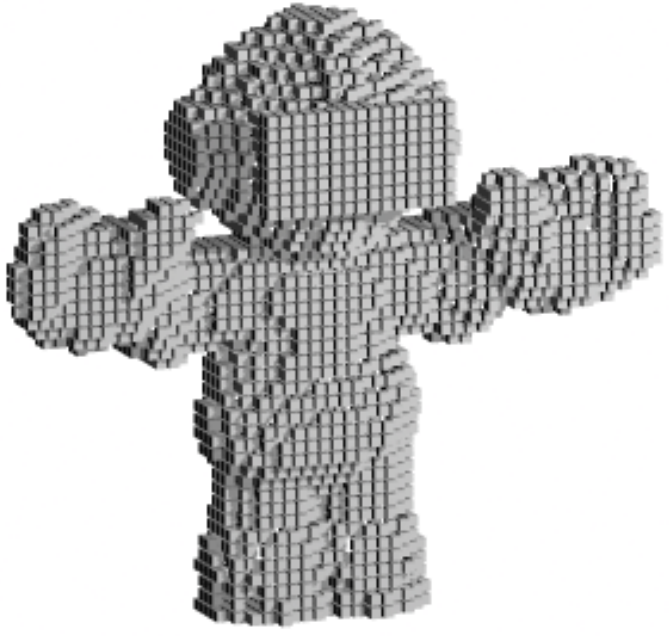
- scanning produces solids
- some apps require solids
- algorithms require solids
- some operations easier with solids

Some Volumetric Representations

- discrete volume representations
- implicit representations

- overlay grid on solid
- represent grid cells called voxels

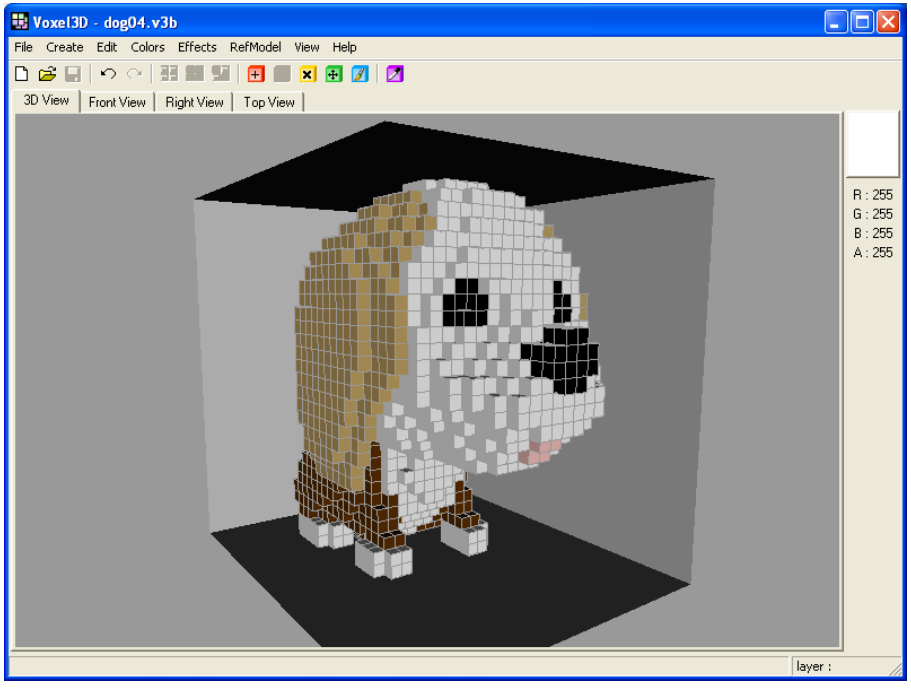




Voxel Properties

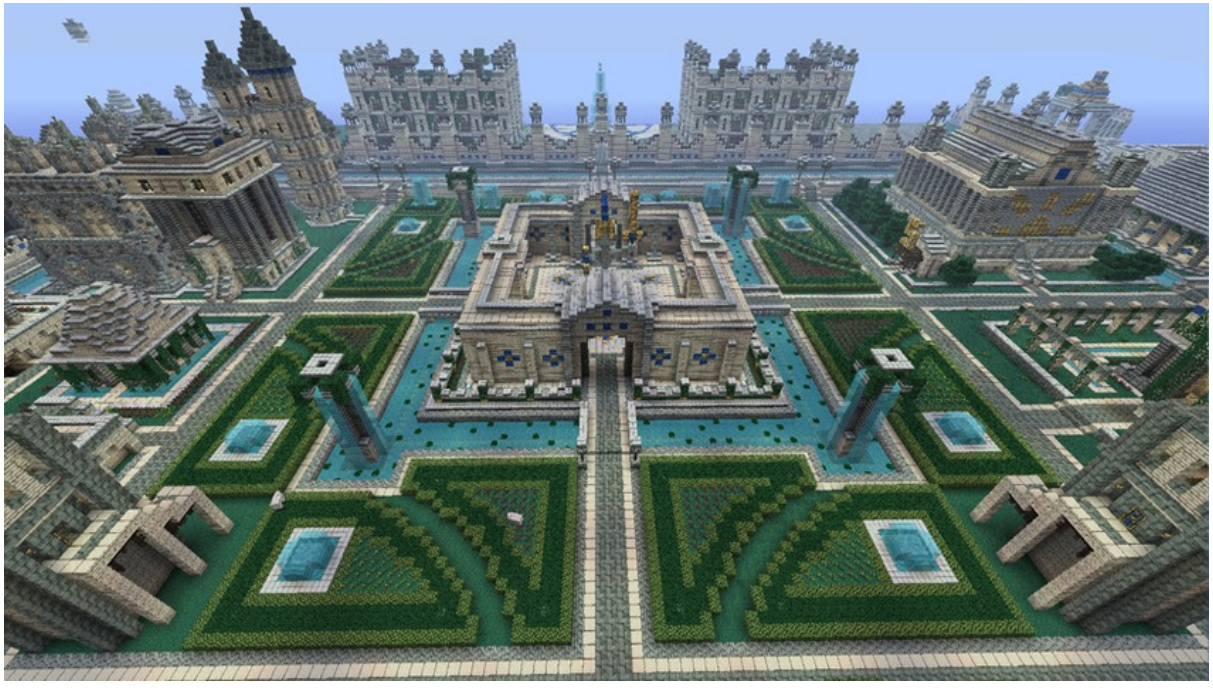
can store solid properties in voxels such as

- occupancy
- color
- density



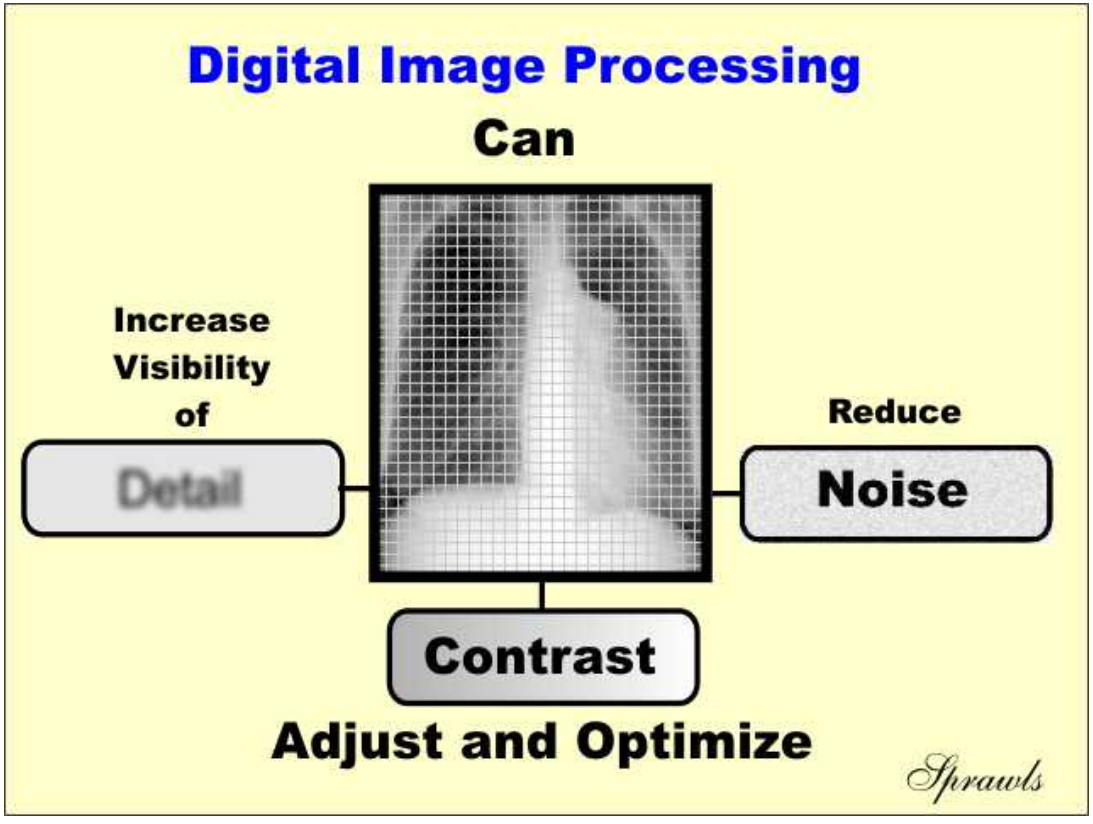
voxel3d

- $O(n^3)$ voxels
- for example 1 billion voxels in 1000^3



Sanakan Soryu

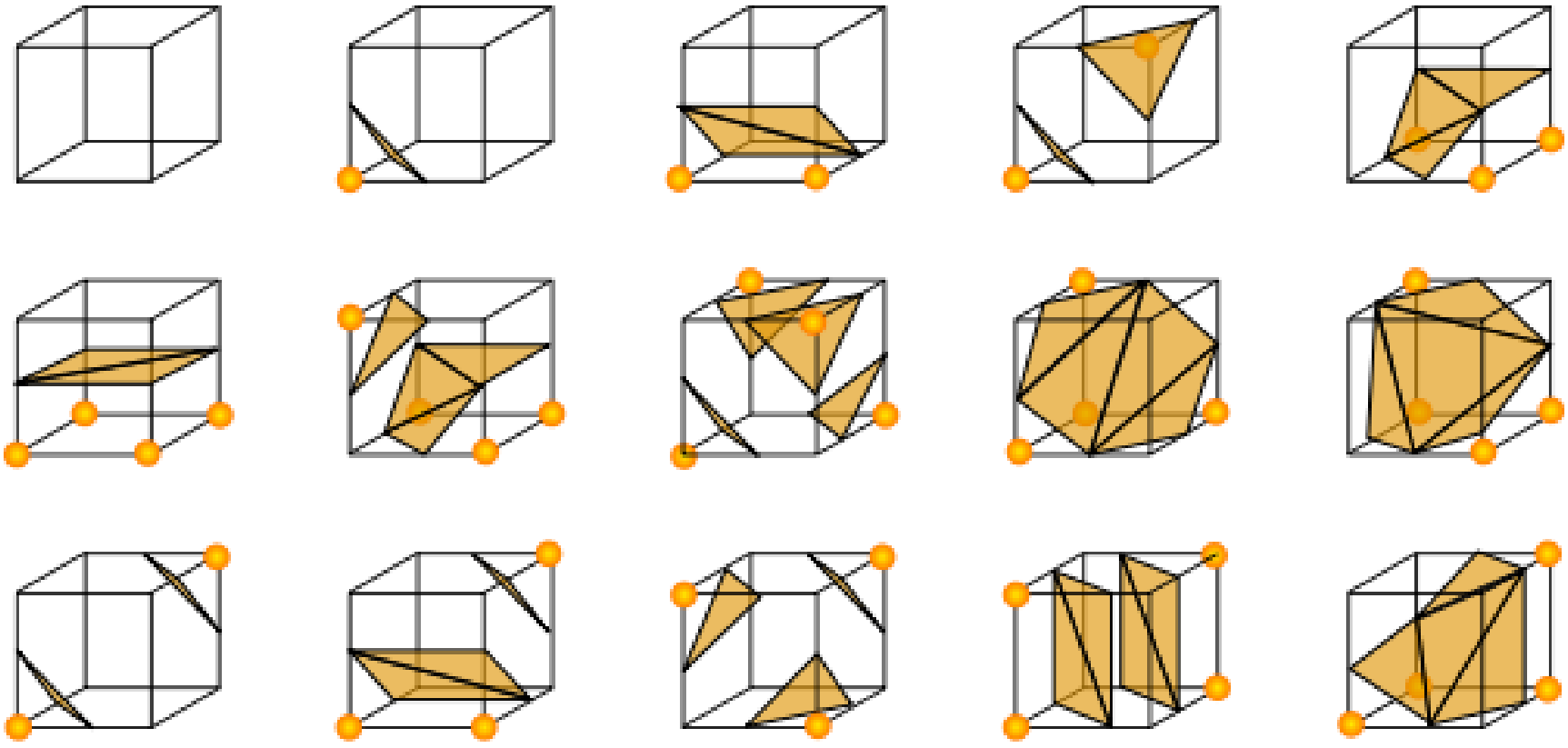
- like image processing
- resampling / resizing
- examples blur, sharpen, edge detection, ...



- ray casting
- slicing
- isosurfaces

Conversion from Voxels to Surfaces

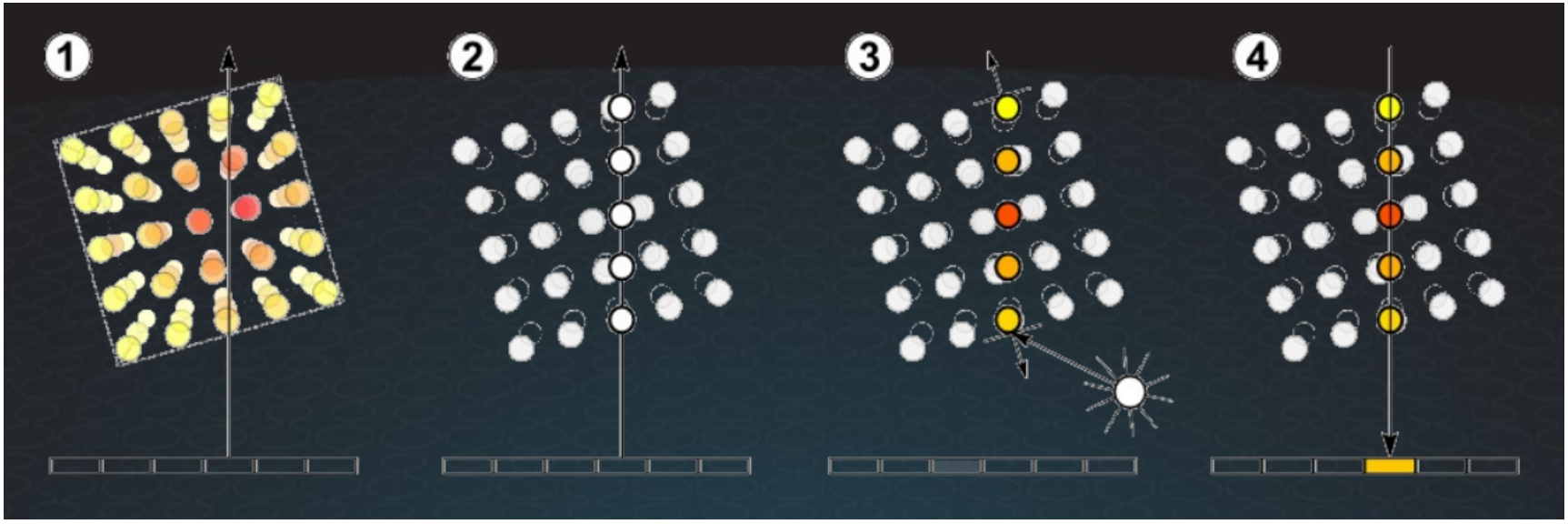
- marching cubes



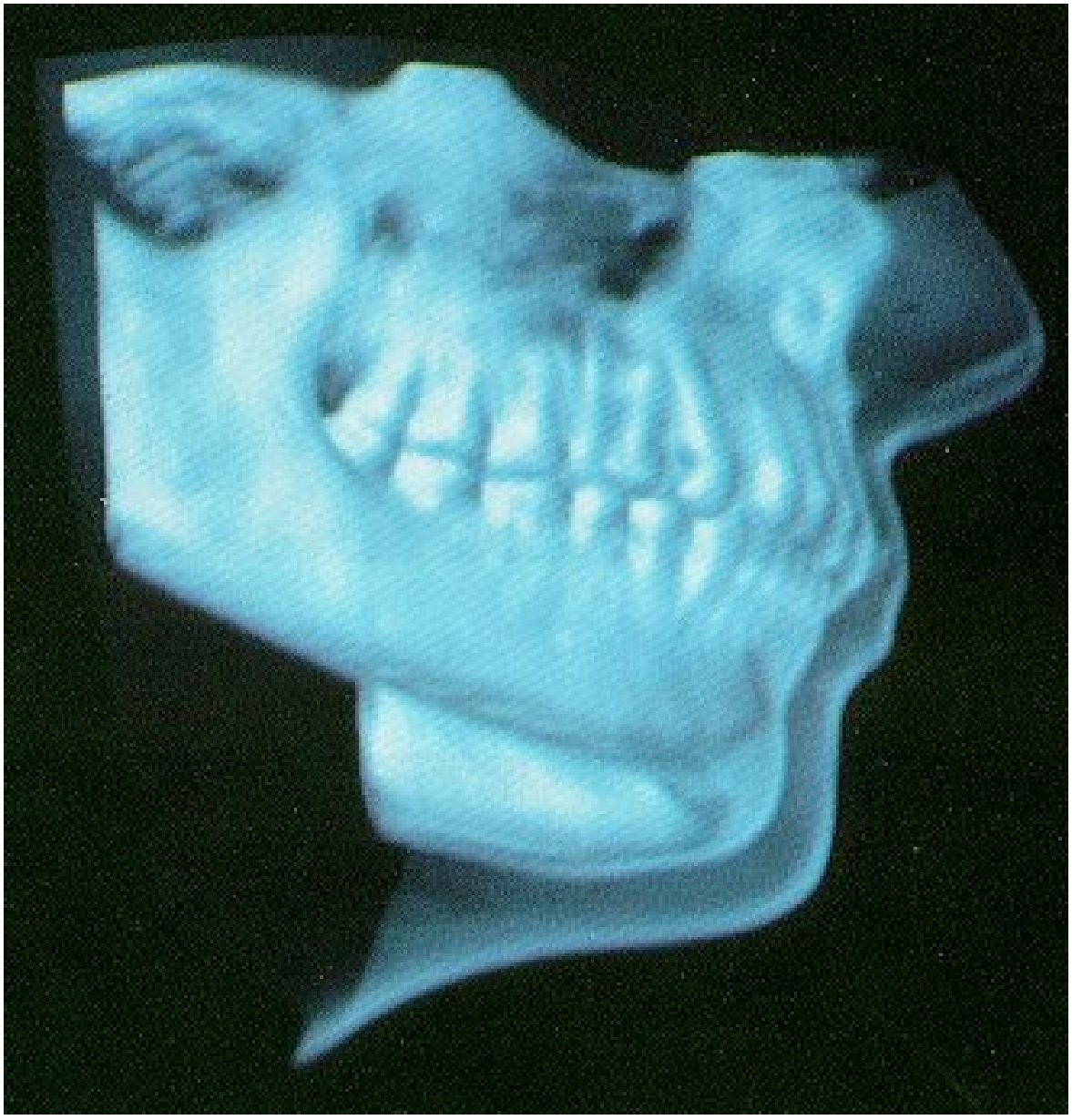
Lorenson

Ray Casting Volumes

- project rays and count

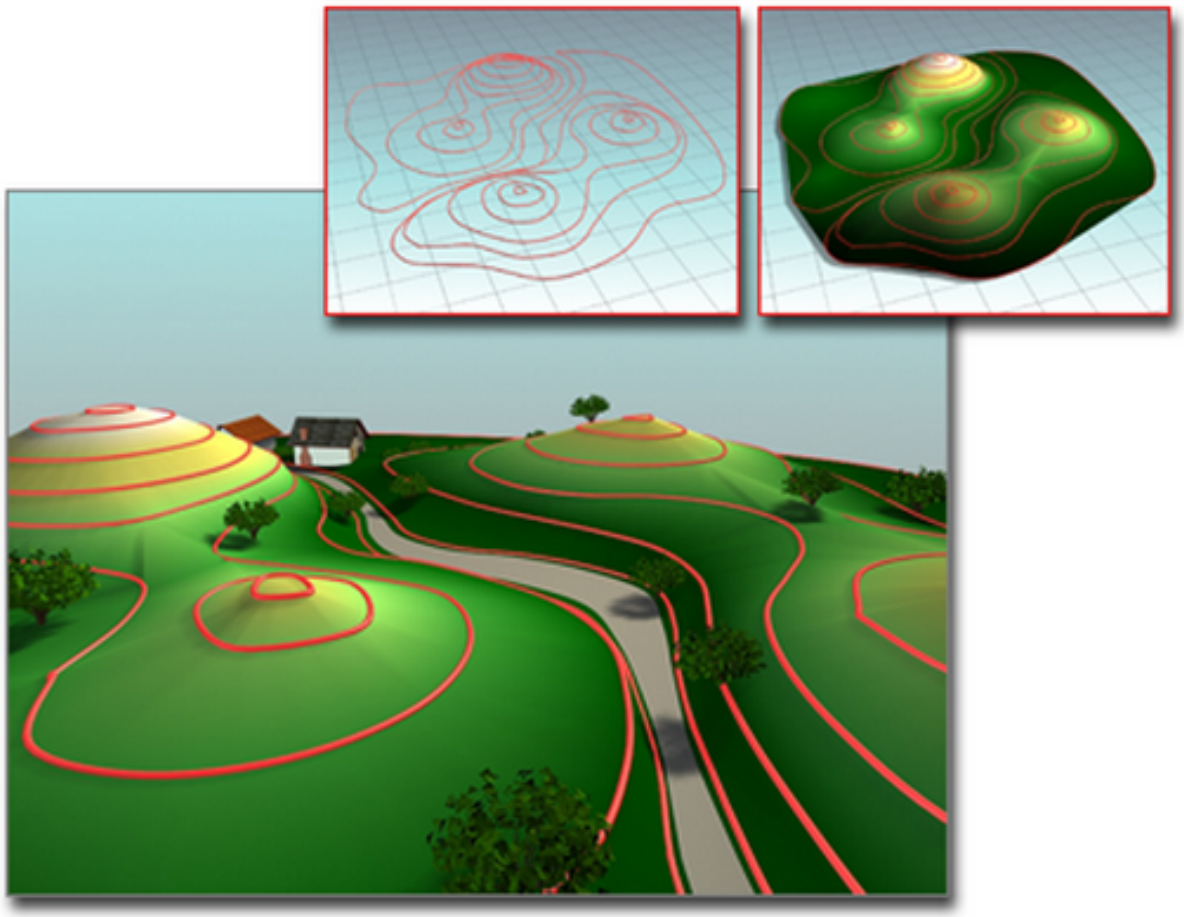


Ray Casting Jaw



John Pawasauskas

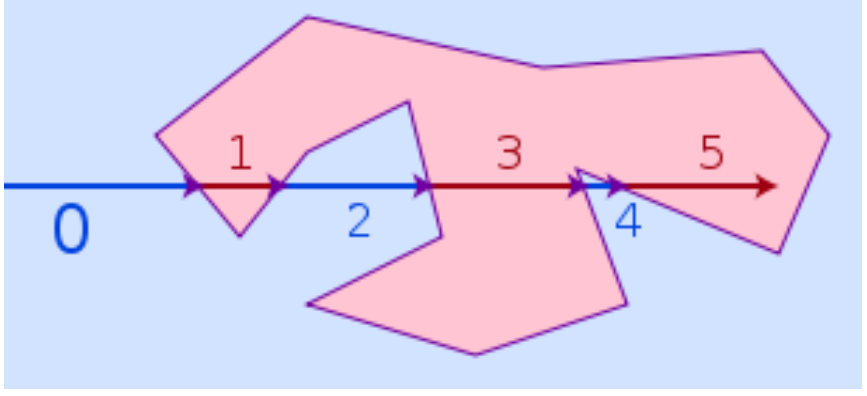
- slices at equal values

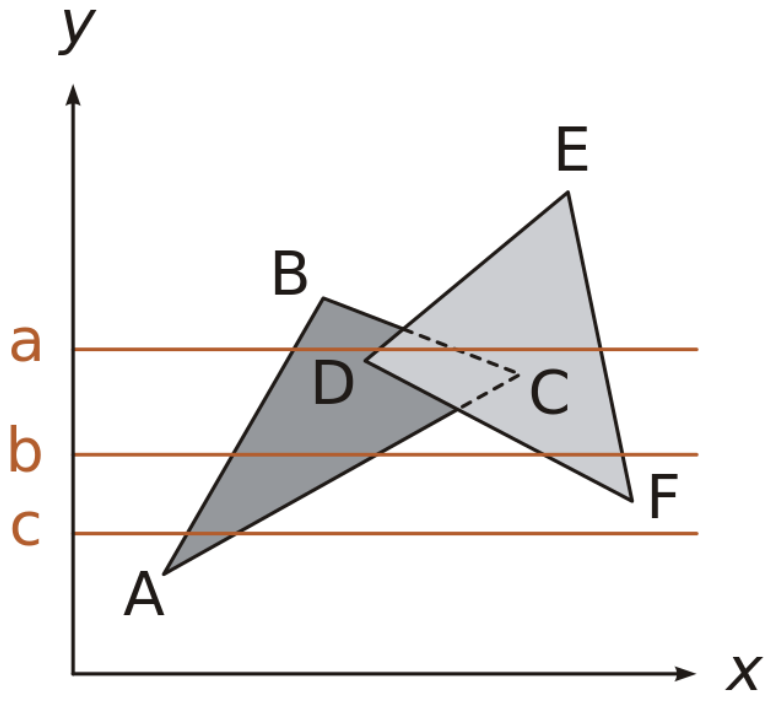


3ds max

Conversion from Surfaces to Voxels

- ray casting
- rasterization





Voxels Pros / Cons

pros

- simple and intuitive
- easy acquisition

cons

- approximate
- not affine invariant
- large
- slow to display

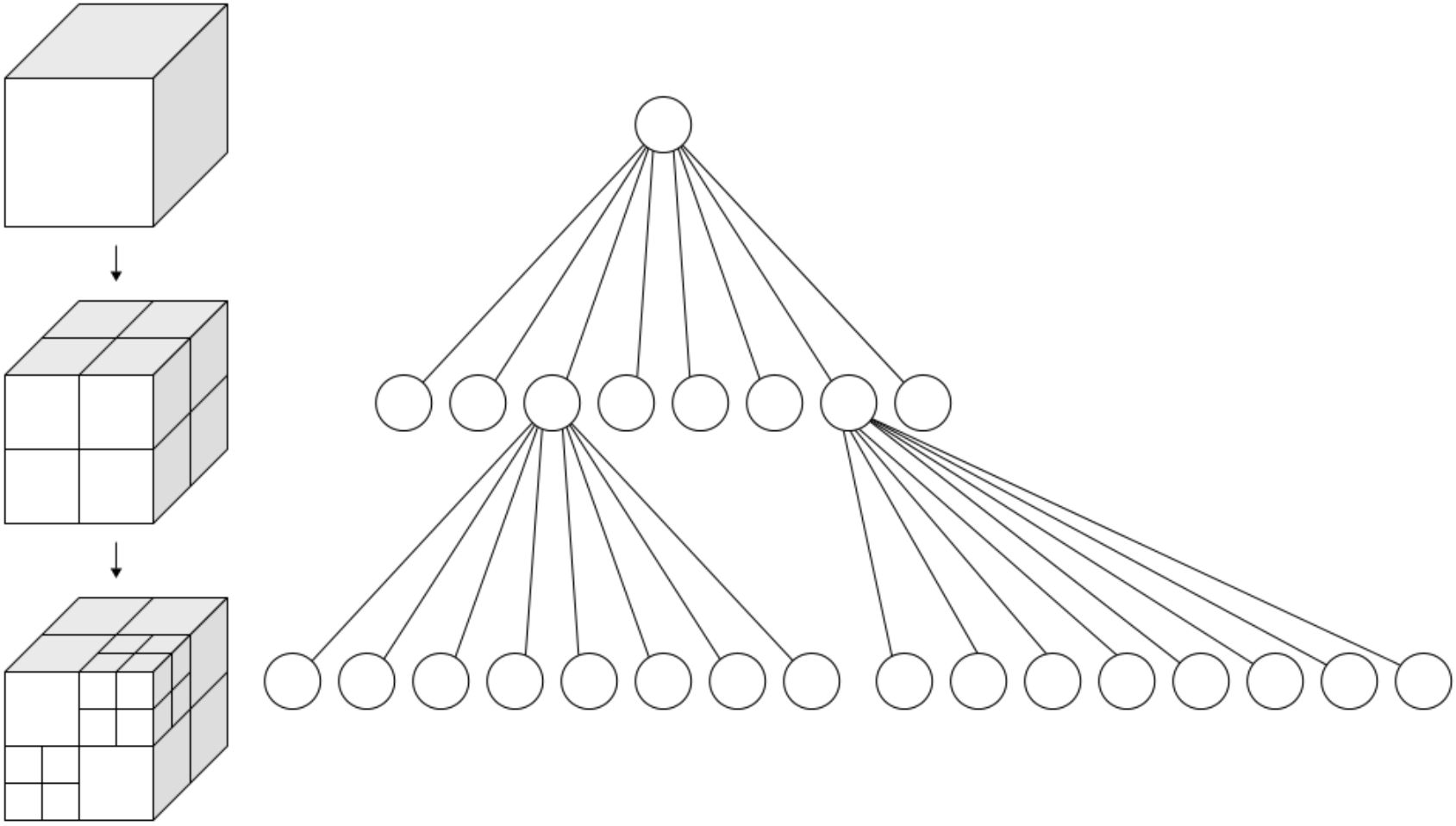
Voxels Resolution

- what is appropriate resolution?
- like photoshop

cnc design

Octree Representation

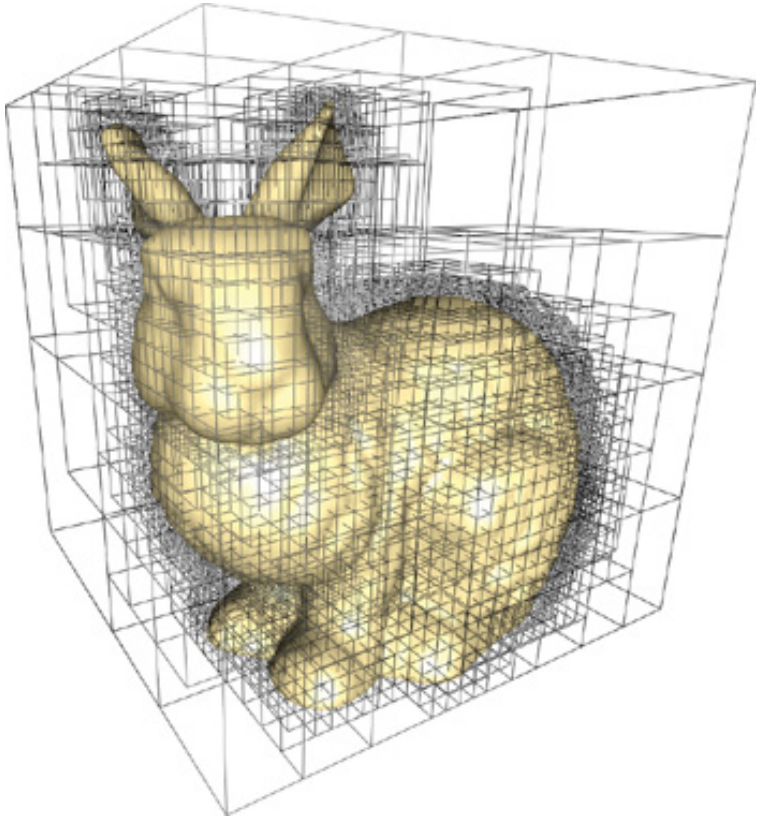
- hierarchical representation
- use detail where needed
- smaller



blah

Octree Results

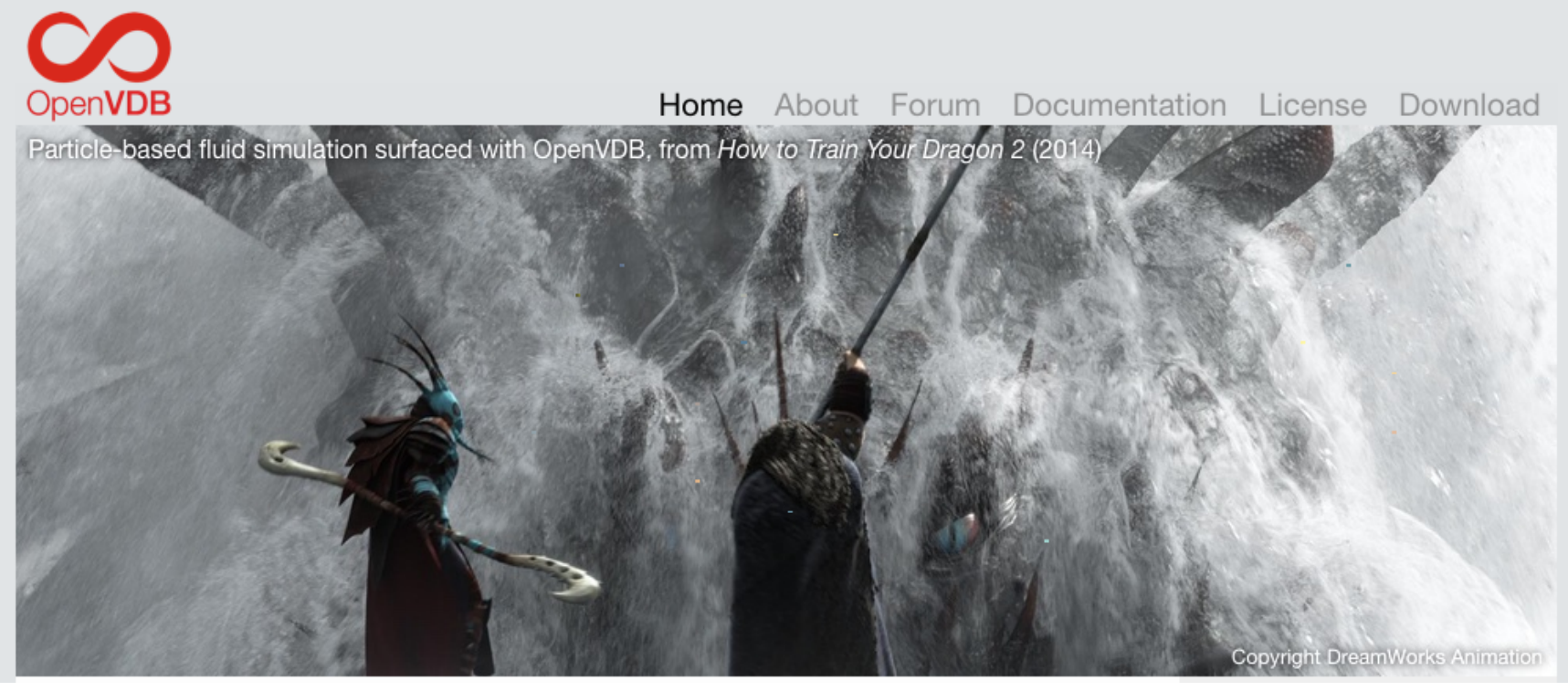
- construct tree based on uniformity
- voxel techniques apply
- operations, display



blah

Open VDB

- time varying
- space efficient
- simulation



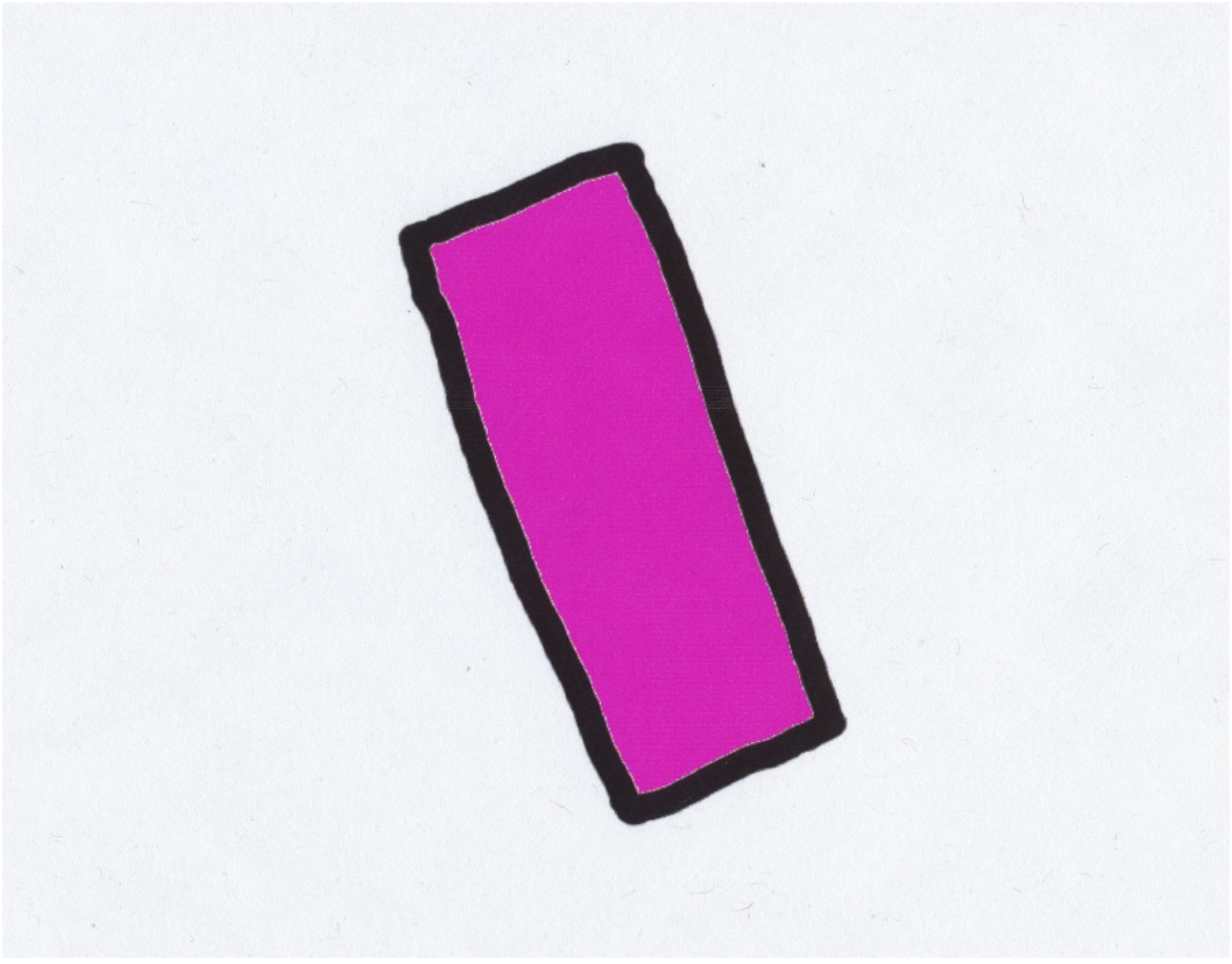
www.openvdb.org

function takes point says

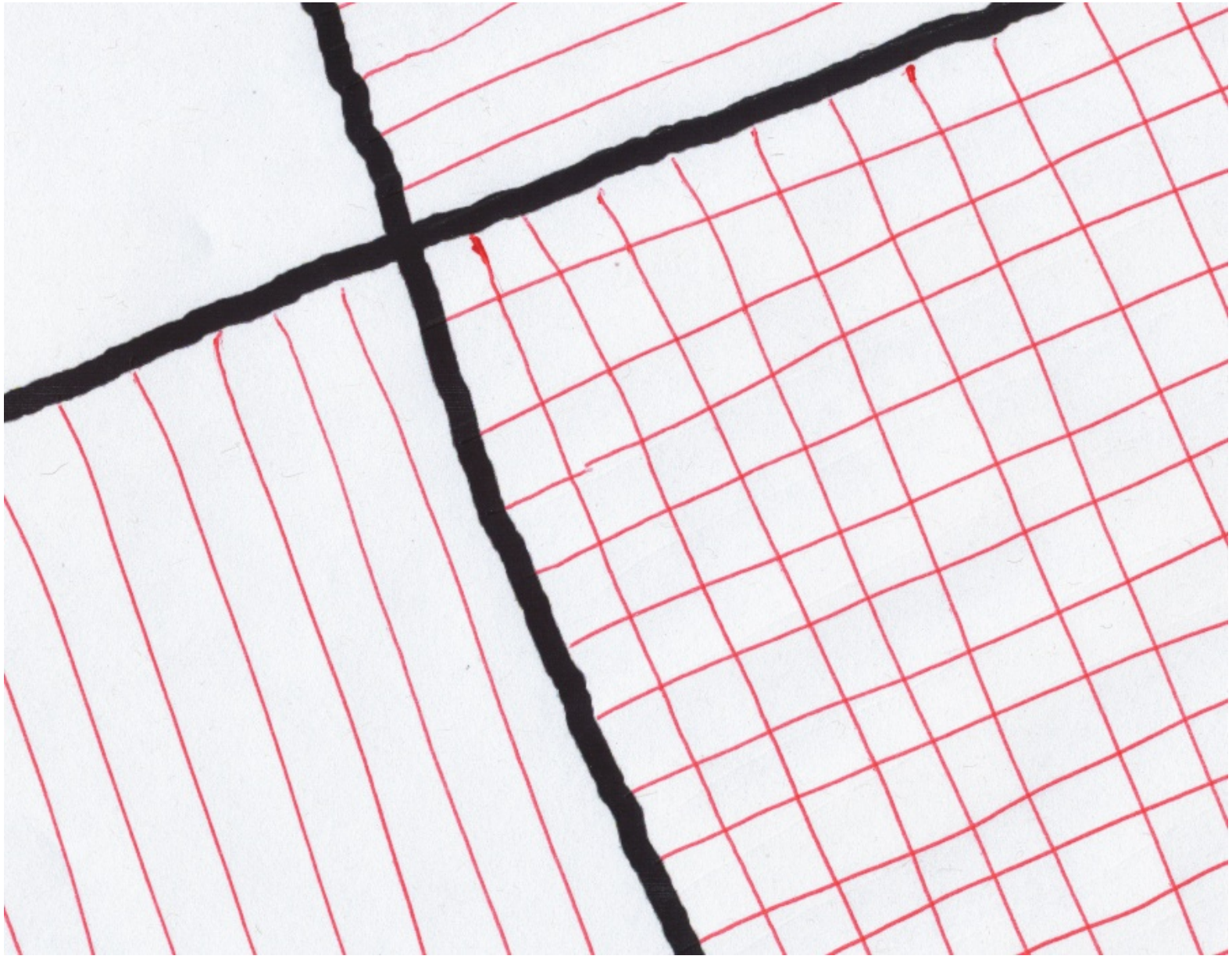
- in or out
- distance to closest point

Half Spaces

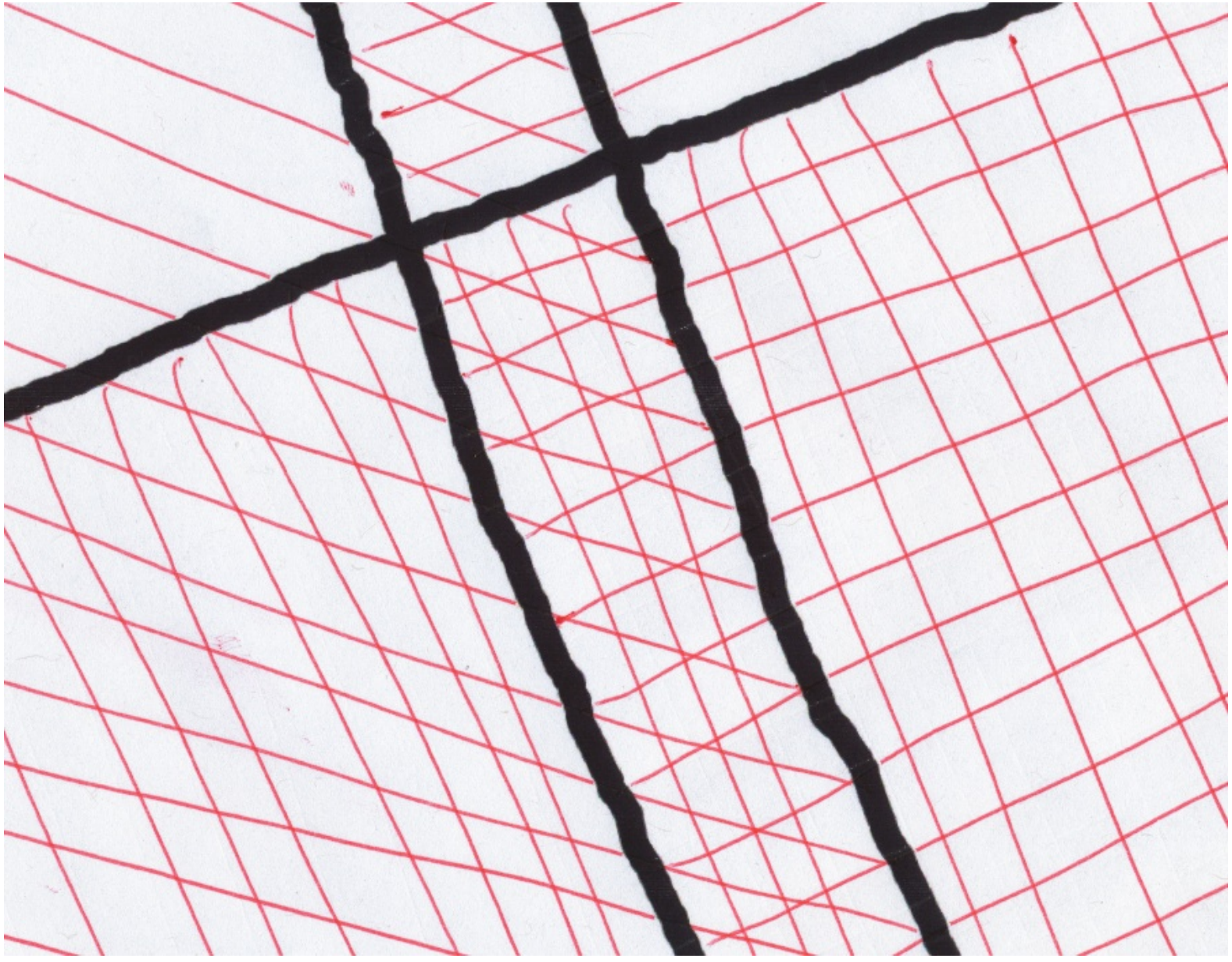
- primitives are half spaces
- solids are operations on half spaces



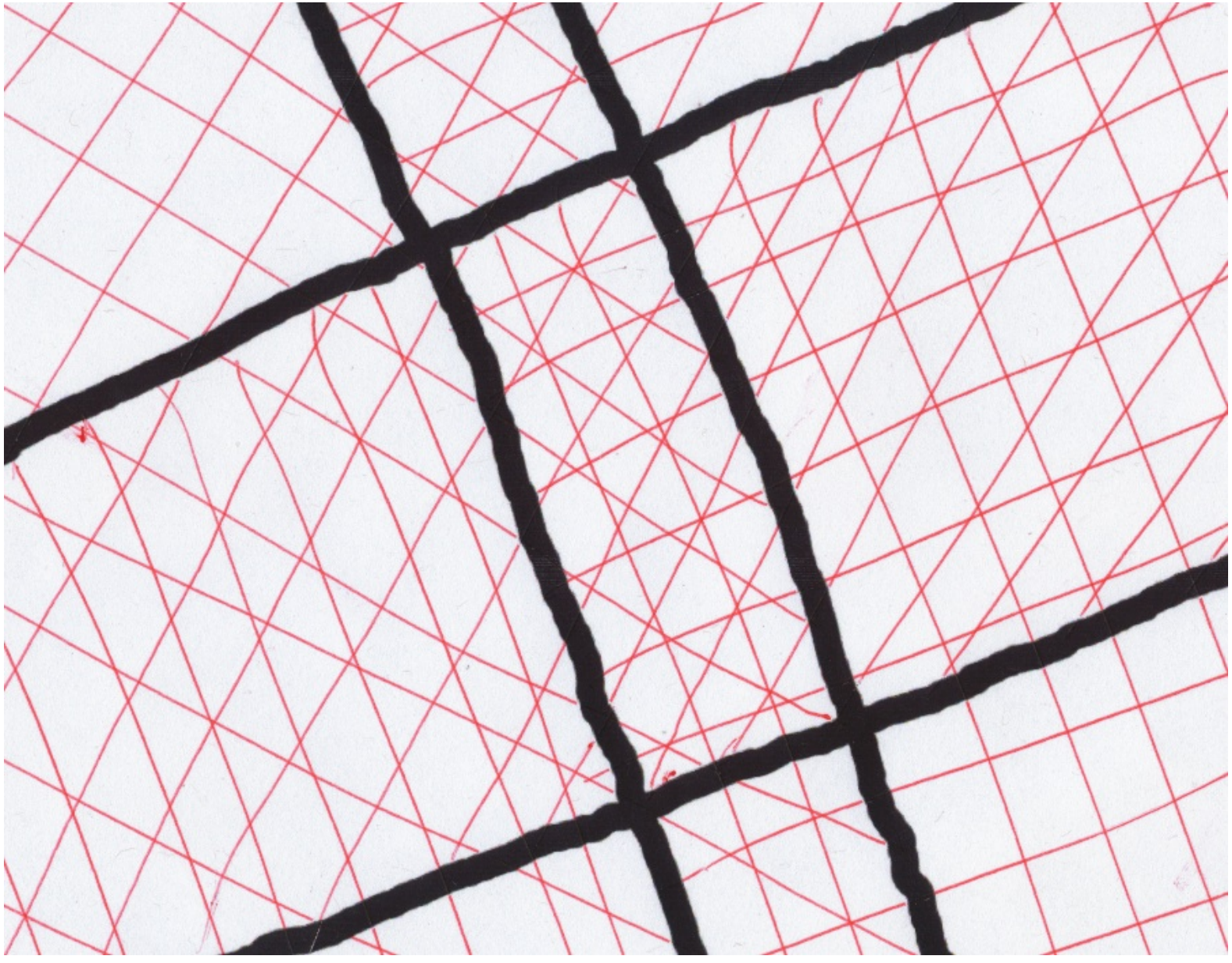
Half Space Composition



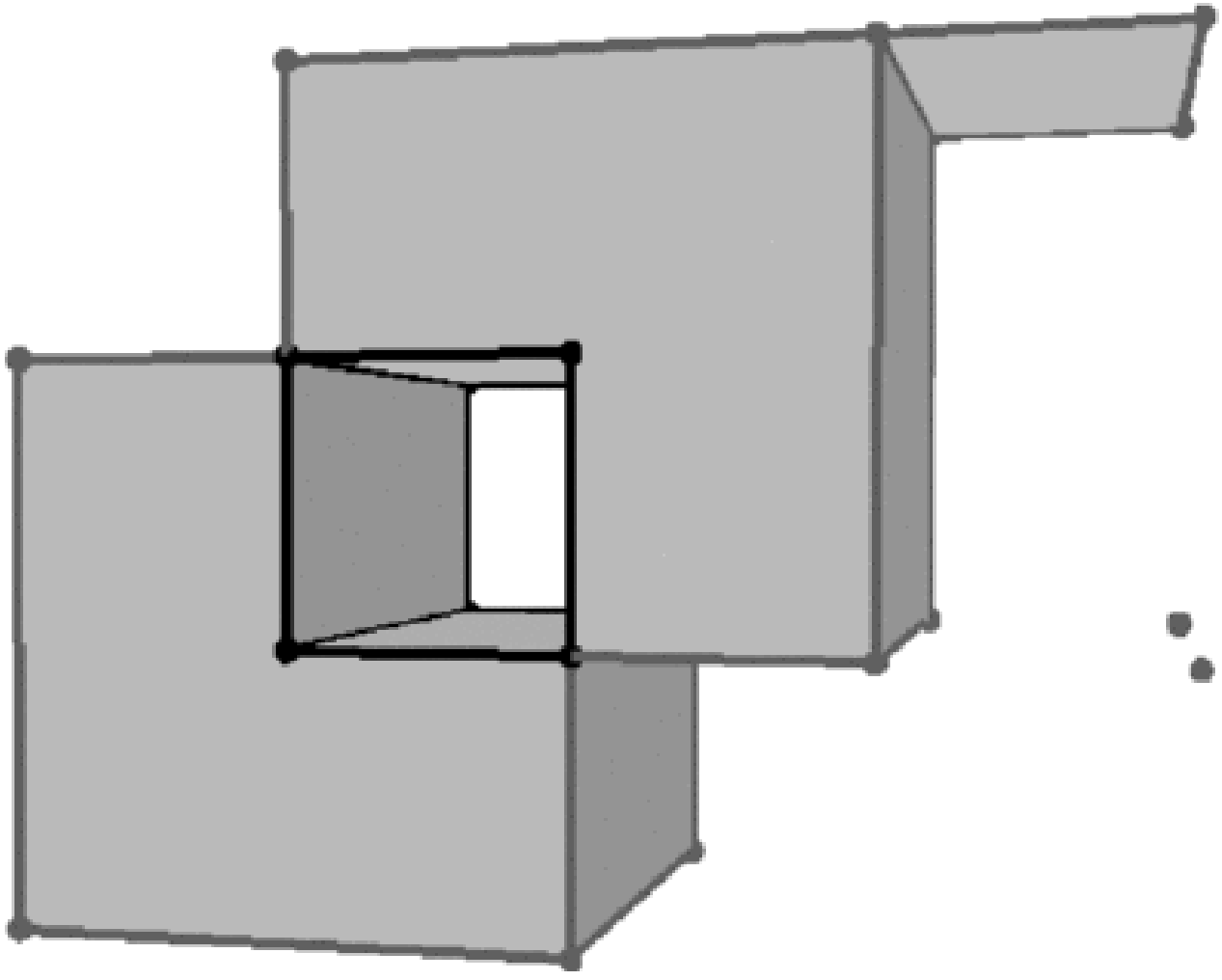
Half Space Composition



Half Space Composition



- C++ library
- B-Rep also

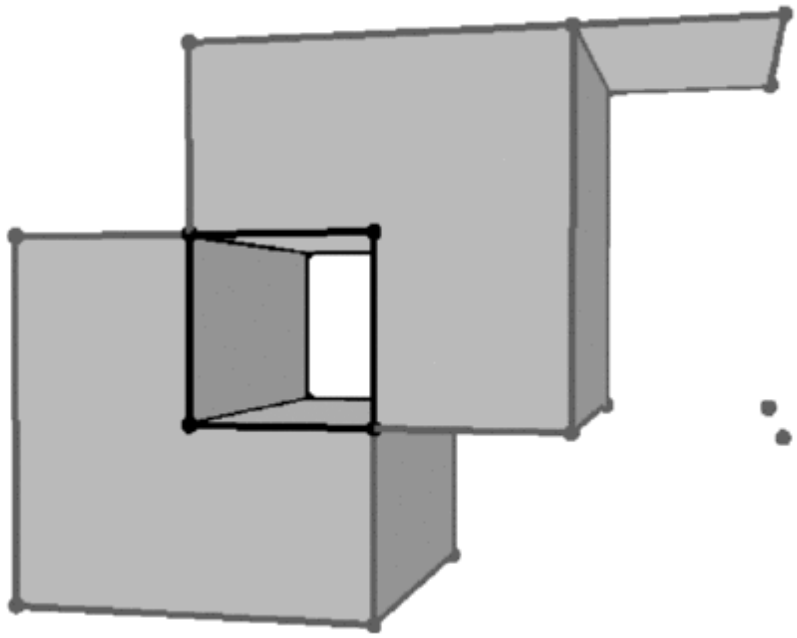


pros

- closed under set operations
- builds boundary

cons

- curves approximated with lots of half spaces

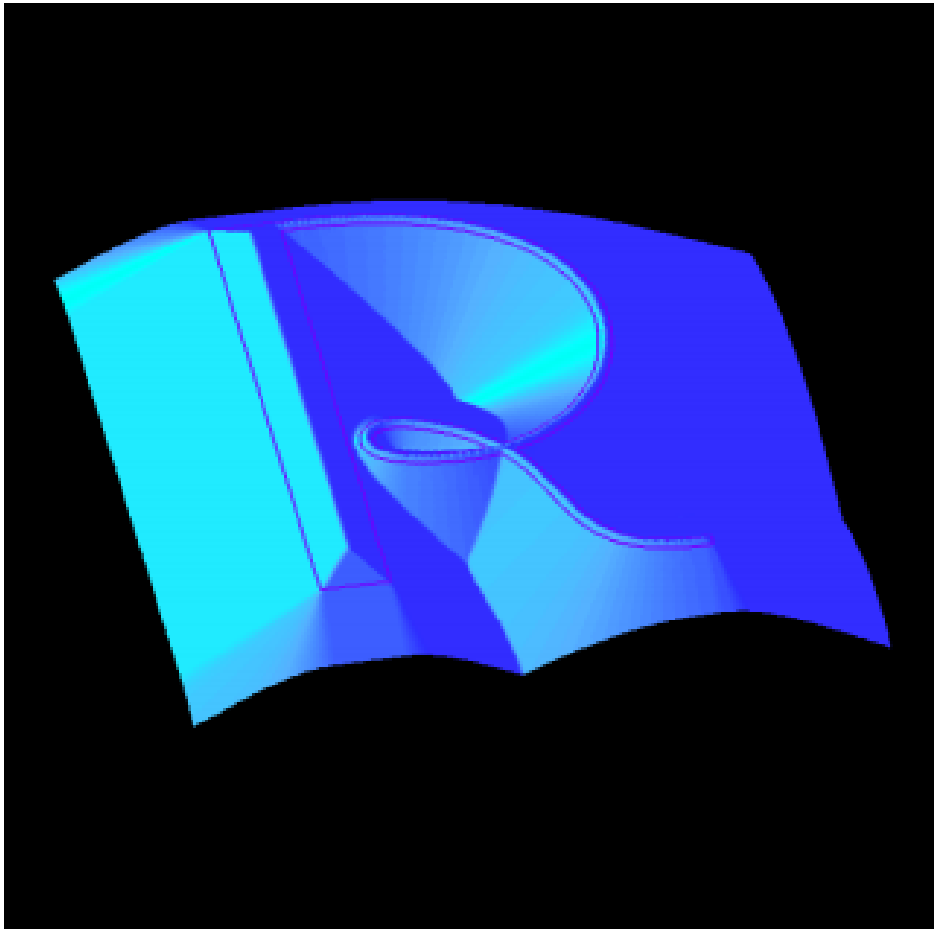


- distance to closest object for every point in space
- zero on boundary
- negative inside
- positive outside

R shape

Distance field of R

perry + frisken

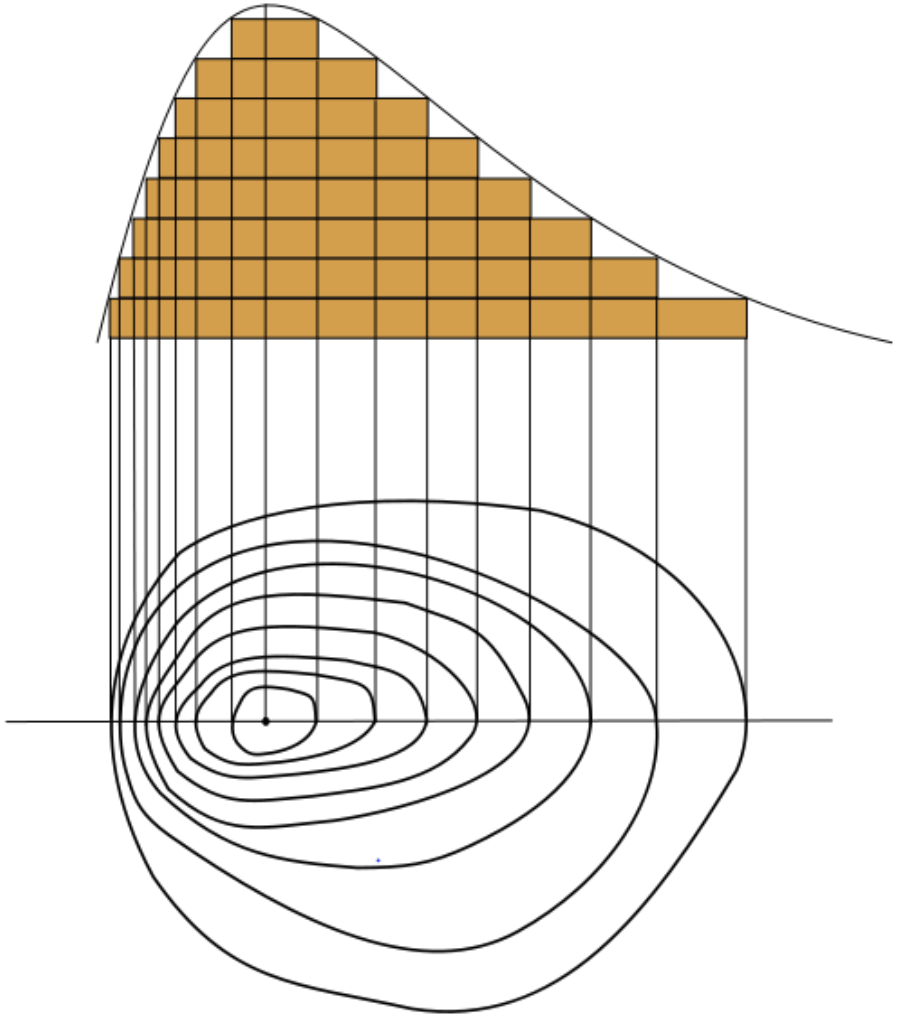


3D visualization of distance field of R

perry + frisken

Iso Surfaces

- surfaces with equal distances
- zero iso surface is boundary



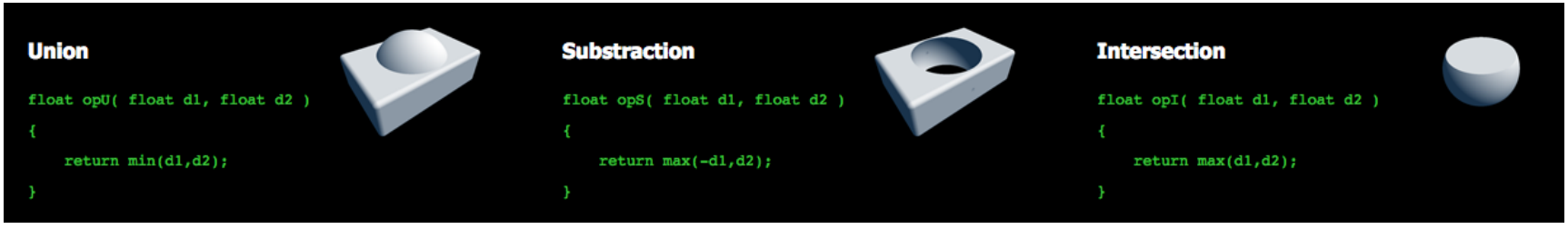
Distance Field Properties

- defined everywhere
- easy inside/outside test
- gradients of field provide useful information
  - on boundary gradient is surface normal
  - off boundary gradient is direction to closest boundary

Distance Field Operations

fast and simple operations:

- $dist(A \cap B) = min(dist(A), dist(B))$
- $dist(A \cup B) = max(dist(A), dist(B))$
- $dist(A - B) = min(dist(A), -dist(B))$



iquilezles

- sphere $dist(p) = sqrt((p - c)^2)$

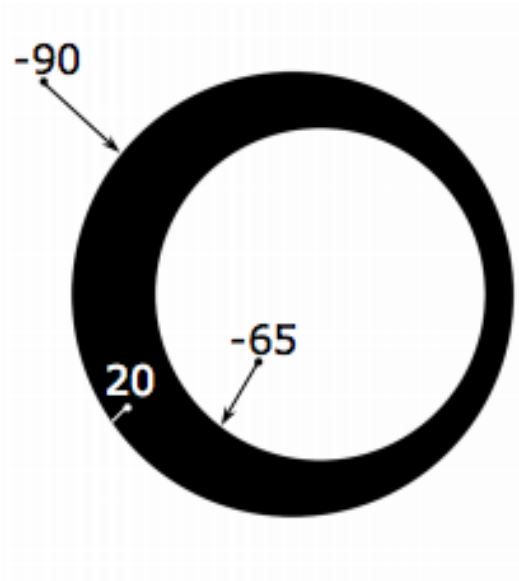
Sphere - signed

```
float sdSphere( vec3 p, float s )
{
  return length(p)-s;
}
```

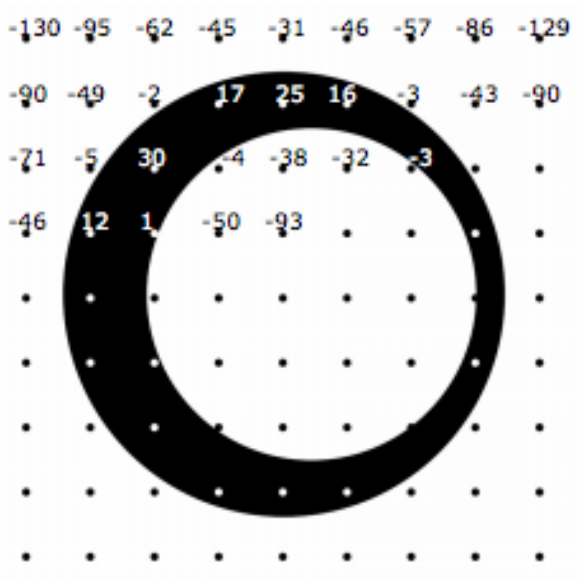
iquilezles

- regularly sampled
- octree
- adaptively sampled

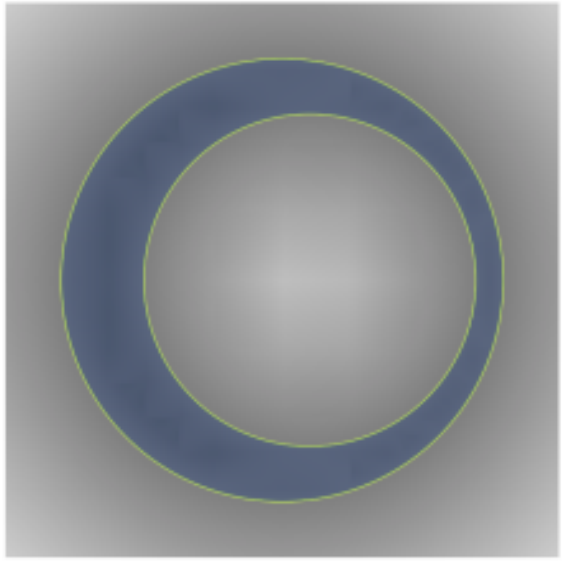
- use trilinear reconstruction



2D shape with sampled distances to the surface



Regularly sampled distance values



2D distance field

perry + frisken

Trilinear Interpolation

$$P_1(x) = p_{000}(1-x)(1-y)(1-z) + p_{100}x(1-y)(1-z) + p_{010}(1-x)y(1-z) + p_{001}(1-x)(1-y)z + p_{101}x(1-y)z + p_{011}(1-x)yz + p_{110}xy(1-z) + p_{111}xyz$$

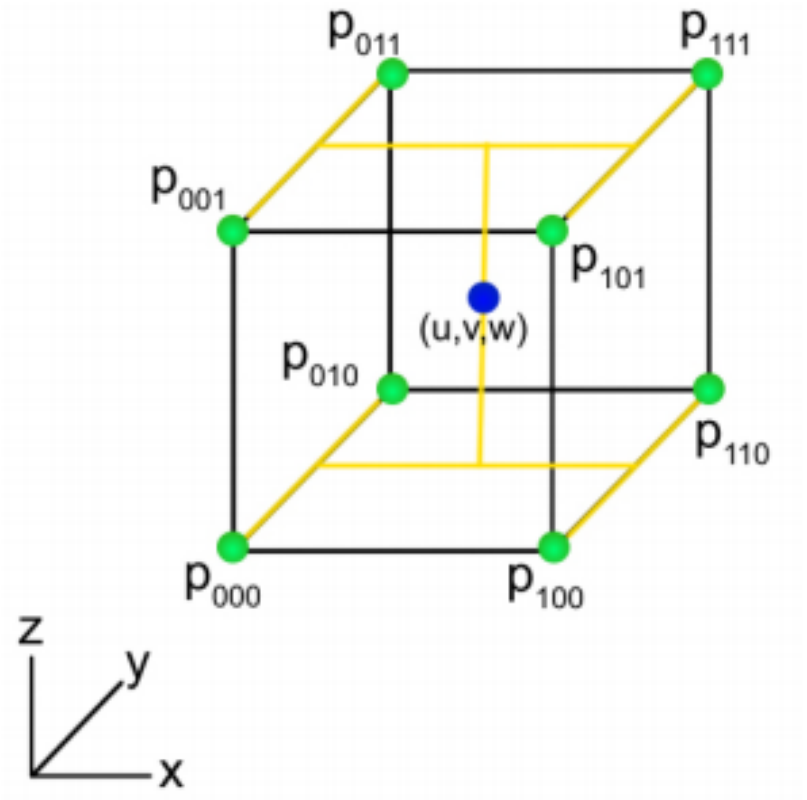
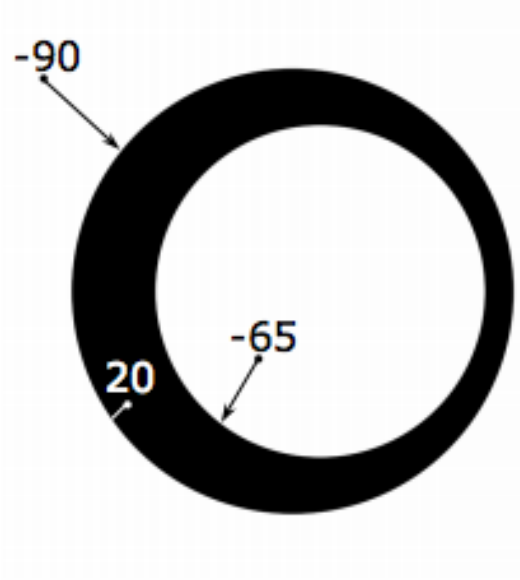


Figure 2. Trilinear interpolation

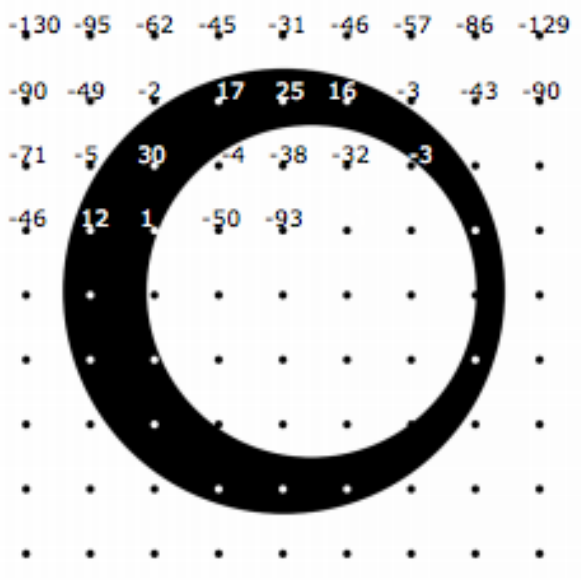
hagan + braley

Regularly Sampled

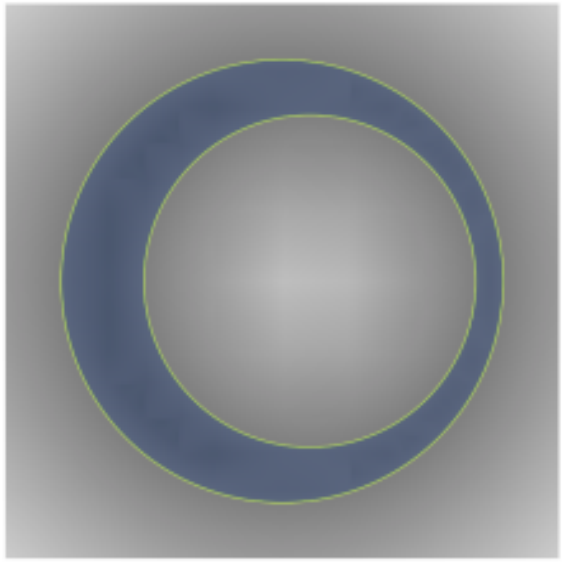
- insufficient sampling results in aliasing
- excess sampling requires excess memory



2D shape with sampled distances to the surface



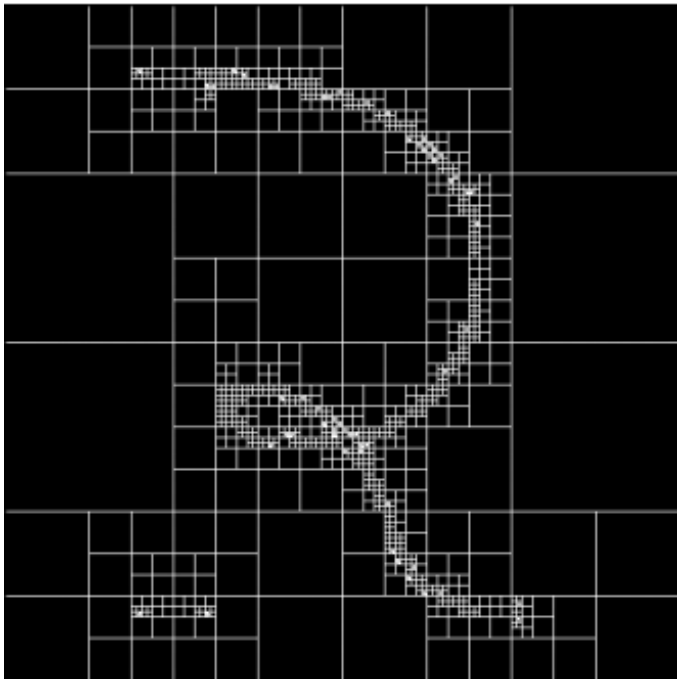
Regularly sampled distance values



2D distance field

Octree Sampled

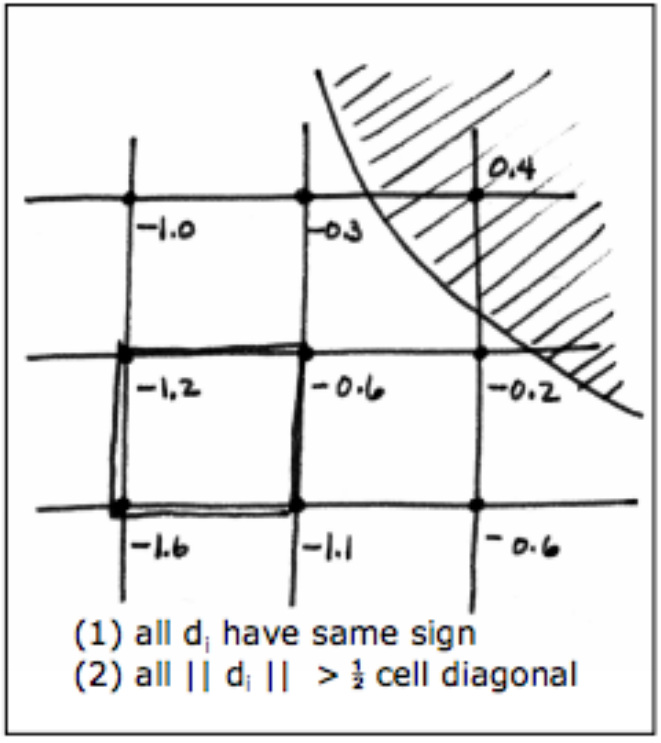
- still have to decide leaf resolution



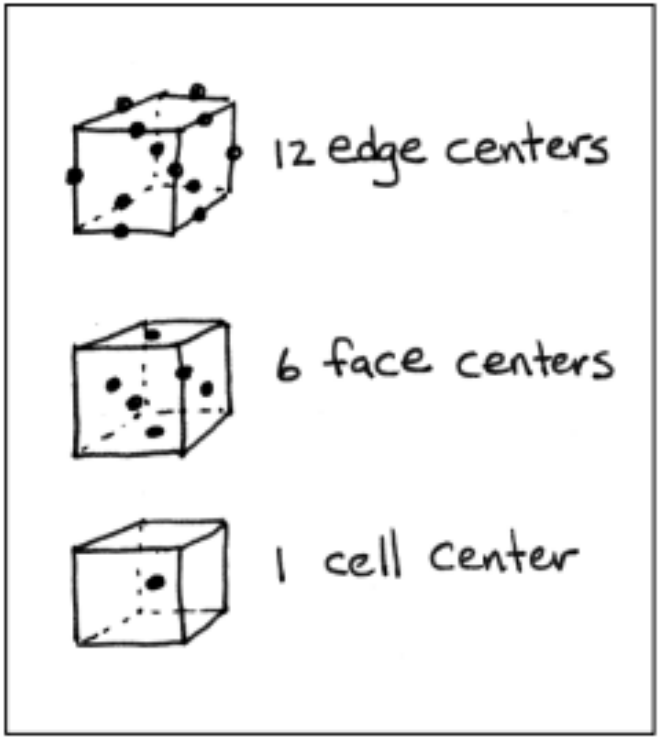
perry + frisken

Adaptively Sampled

- stop when can reconstruct with sufficient accuracy
- use test candidates

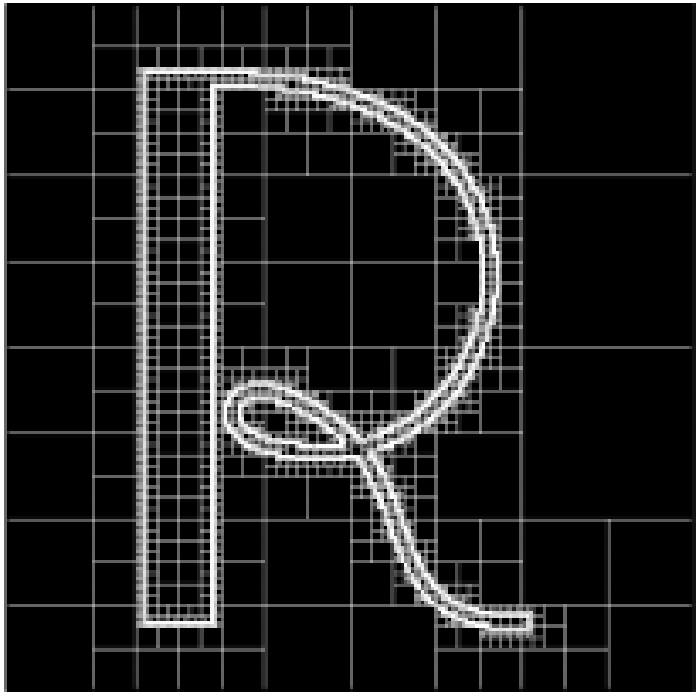


Test to trivially determine if a cell is interior or exterior

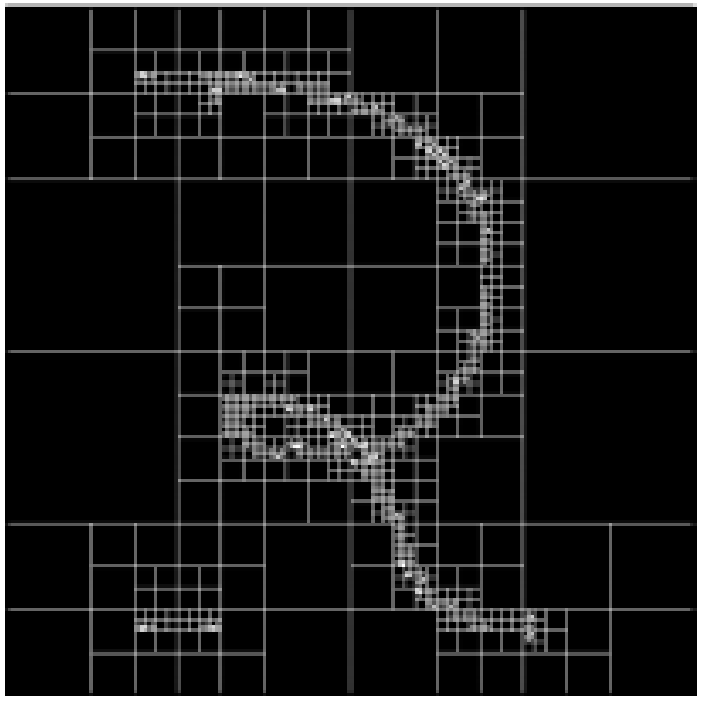


19 test points to determine cell error

Adaptively Sampled Comparison



23,573 cells (3-color)



1713 cells (ADF)

perry + frisken

Distance Field Fab

- distance fields in c++
- optimized with octree and interval analysis

Distance Field Comparison

advantages

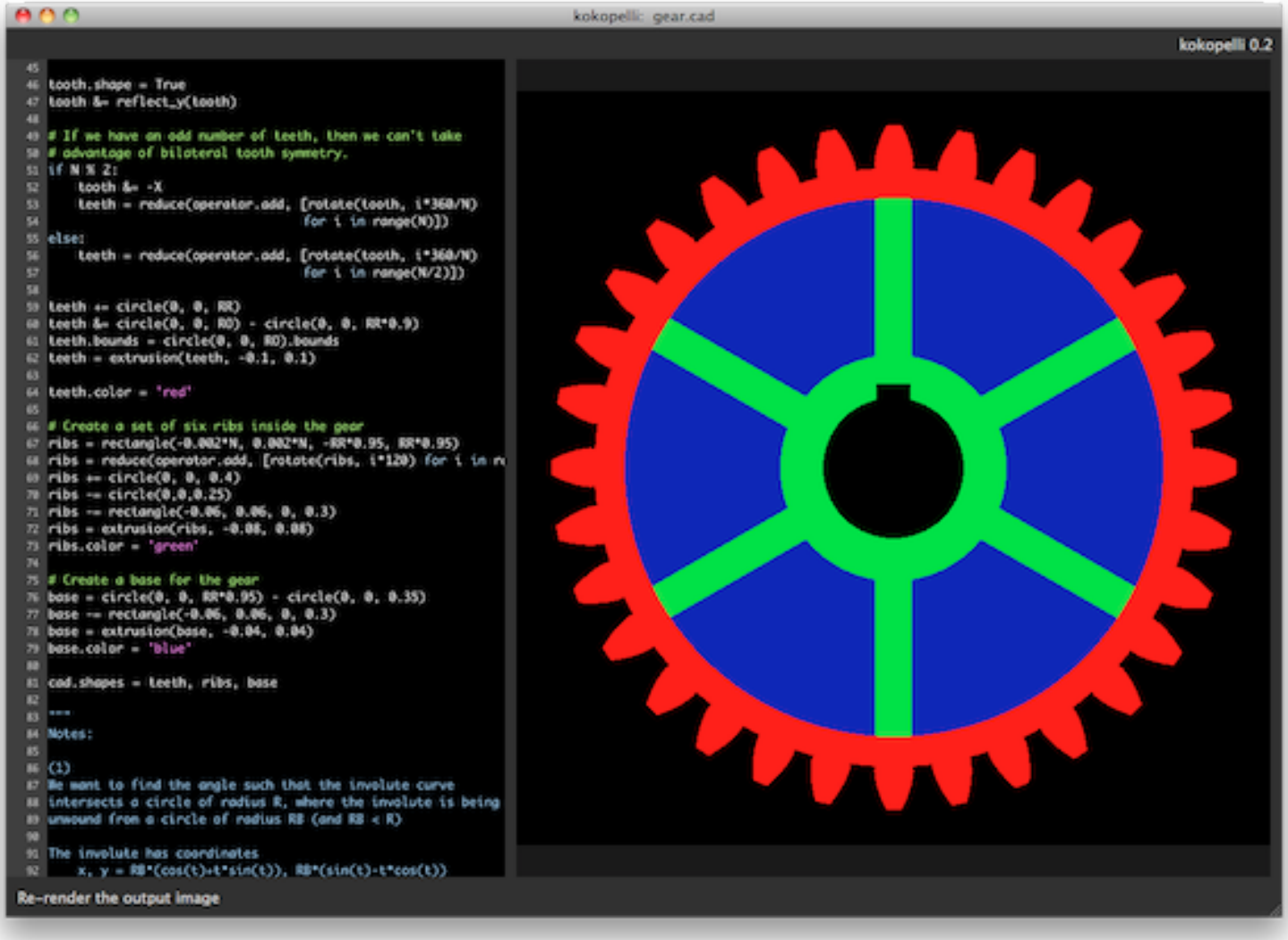
- functional + compositional
- blending
- powerful transformations
- unlimited precision
- iso surfaces – milling
- gradients
- scanning

disadvantages

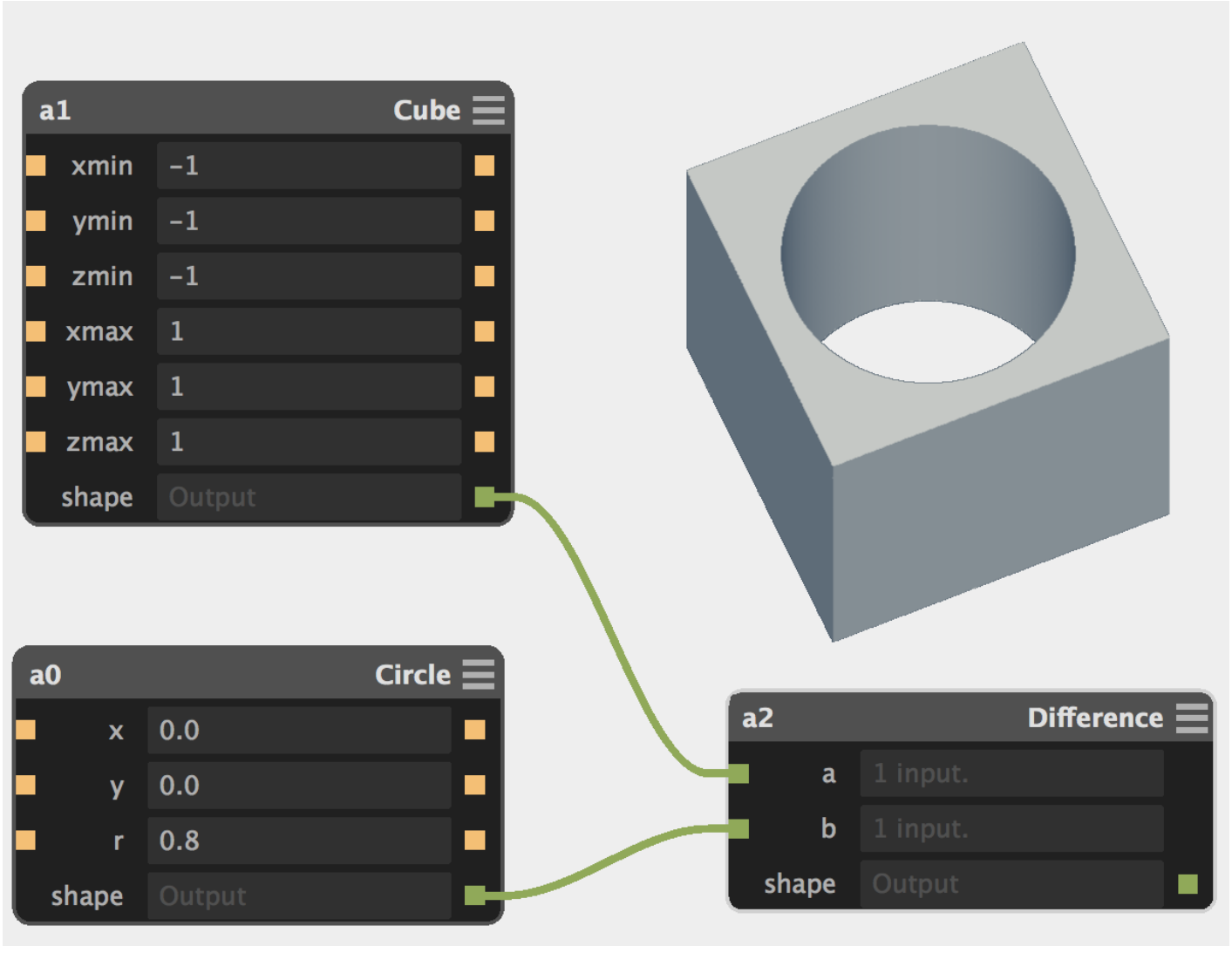
- slow to render
- may be hard to get mesh out
- too mathematical

Distance Field Fab Demos


```
demo "cube(4)"
demo "blend(cube(4),sphere(4),3.5)"
demo "(0.1*(sin(4*x)+sin(4*y)+sin(4*z)))+cube(3)"
demo "xrevolve(square(4))"
demo "zrot(z,pyramid(-4,4,-4,4,-4,4))"
demo "mag1(6,lettercube(4))"
```



by matt keeter



by matt keeter

If Using 2d Cutters Why Volumes?

- more general and abstract
- ready for any fabrication method
- pretty easy to convert

- PolyMesh
- Transformations
- Quaternions
- 3D -> 2D -> 3D

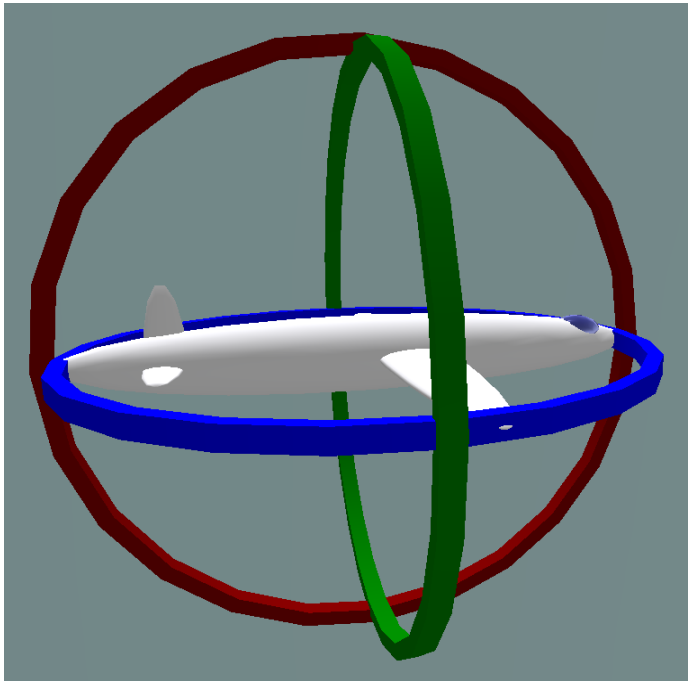
- collection of meshes (or sea of polygons)
  - points
  - indices (faces)
- standard geometry operations
  - transformations
  - hull, bounds, ...
- constructors
  - load from stl
  - points + indices
  - generator from solidpython

- create from numpy arrays
- constructors in transformation.py for basics
- composition can use compose or *= per docs

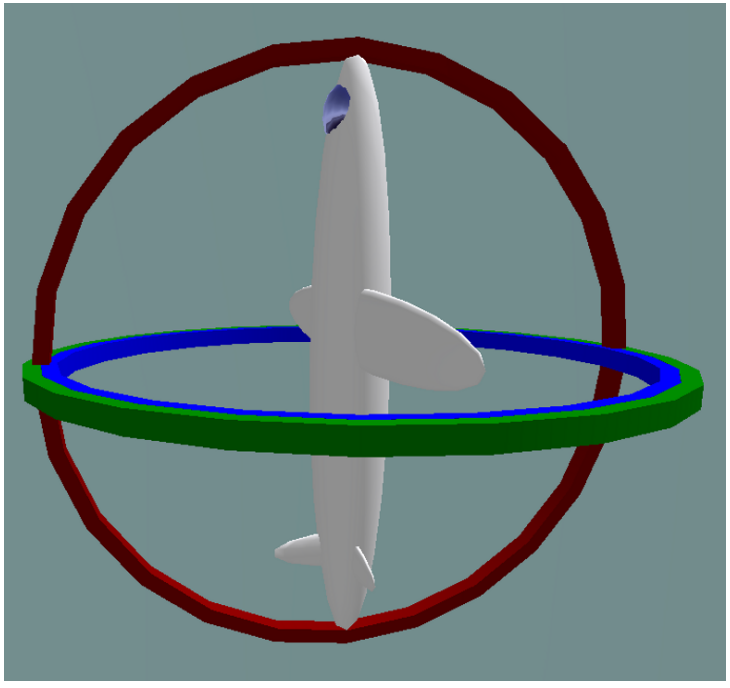
- euler angles – [rx,ry,rz]
- 3x3 matrix
- 4x4 matrix
- quaternion – [s,x,y,z]

Gimbal Lock Problem

- 3 euler angle rep can get locked when moving between angles
- two out of three gimbals are in same plane
- lose one degree of freedom



no gimbal lock



gimbal lock

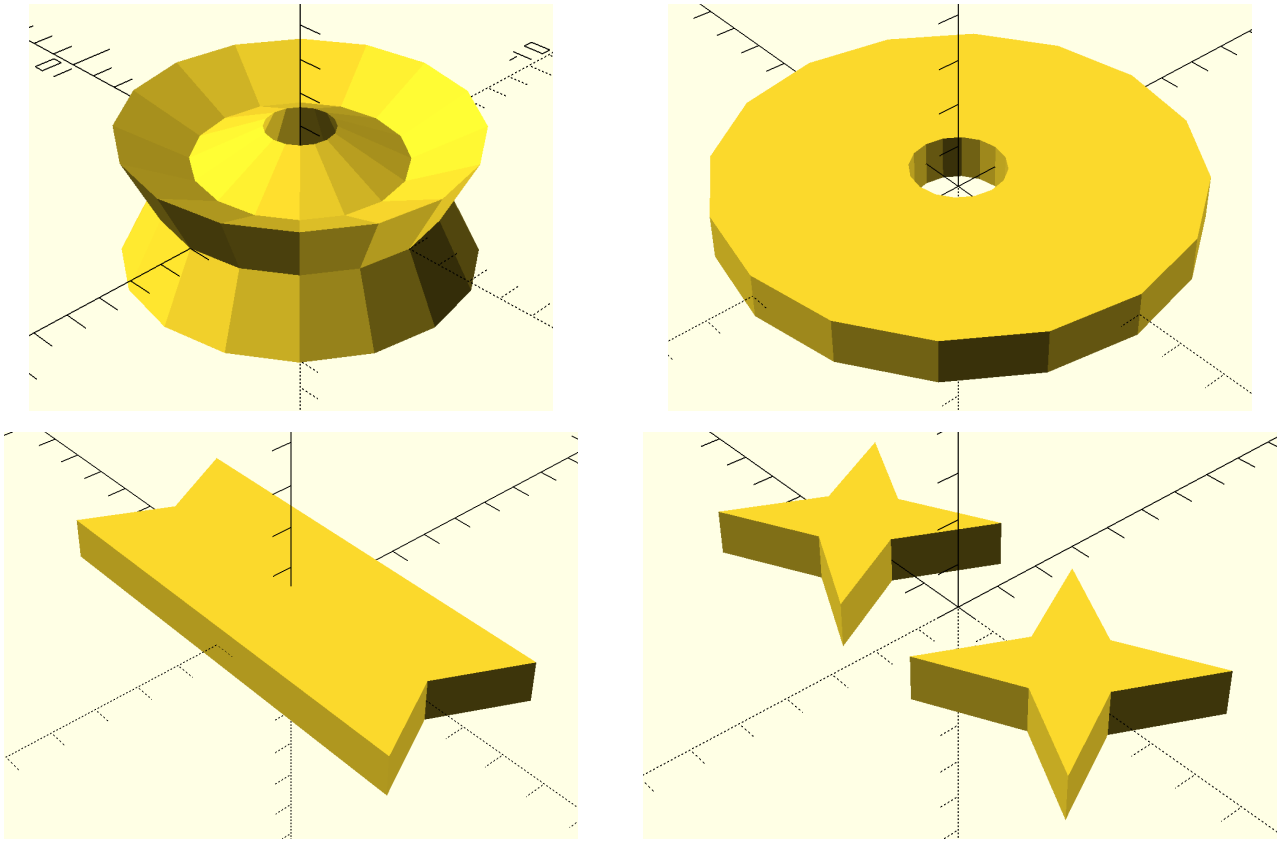
by MathsPoetry

- introduced by Hamilton in 1843
- succinct representation for rotations
- good for interpolation with no gimbal lock problem
- 4x4/3x3 matrix <-> quaternions

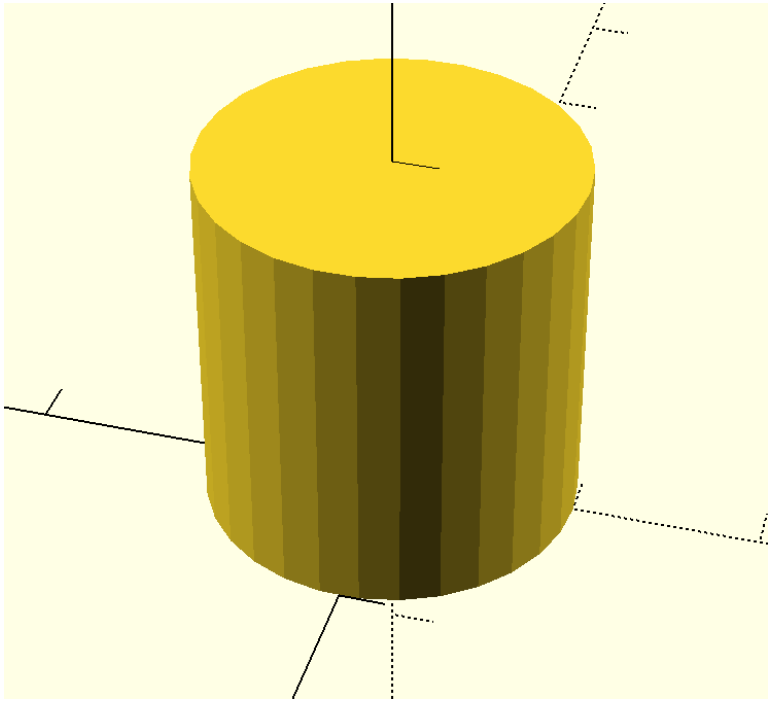
- provide extra degree of freedom
- avoids gimbal lock
- can smoothly and straightforwardly move between any rotations

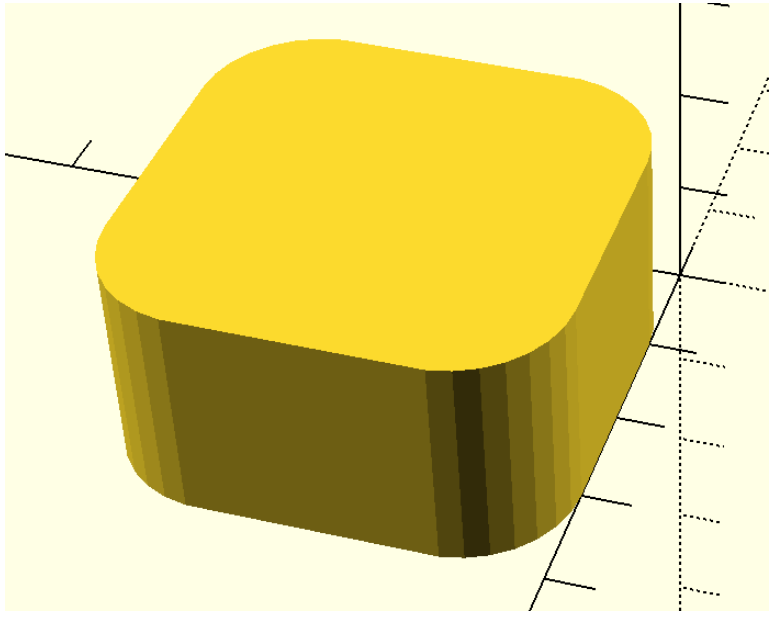
- multiplication
- conjugate
- inverse
- interpolation – slerp
- conversions to/from other rotation representations

- shadow
- slice



- height
- scale
- rotate





next time

- 3D CNC
- 3D Rationalization
- 3D Joinery
- 3D Validation

- *Mesh Basics* by Dr. Ching-Kuang Shene
  `http://www.cs.mtu.edu/~shene/COURSES/cs3621/SLIDES/Mesh.pdf`
- *Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics* by Frisken + Perry + Rockwood + Jones
- *Quaternions* by Ken Shoemake
  `http://www.cs.ucr.edu/~vbz/resources/quatut.pdf`
- *Kokopelli* by Matt Keeter
  `http://www.mattkeeter.com/projects/kokopelli/`
- *Antimony* by Matt Keeter
  `http://www.mattkeeter.com/projects/antimony/`