Neural Fields

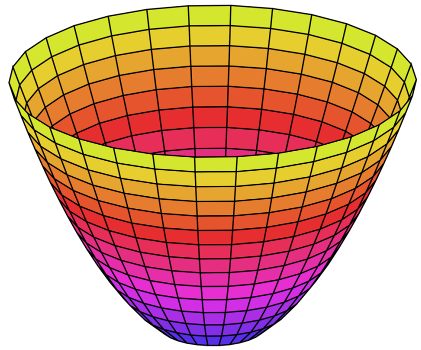
AMC SIGGRAPH 2023 Course on Neural Fields for Visual Computing

Towaki Takikawa, NVIDIA / University of Toronto

Adapted by Jernej Barbic, CSCI 520 Computer Animation, Univ. of Southern California

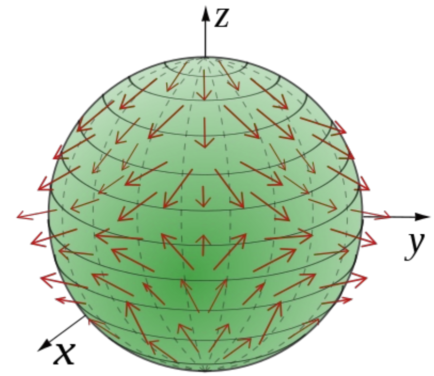
Definition
A *field* is a quantity defined for all spatial and / or temporal coordinates.

Examples of Fields

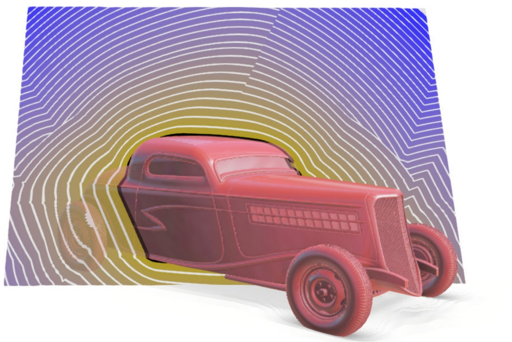


3D Parabola
(Explicit Surface)

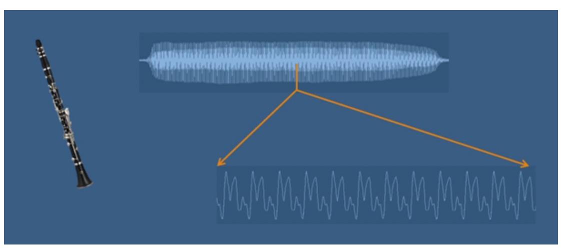
Image



Vector Field



3D Signed Distance Fields
(Implicit Surface)



Audio

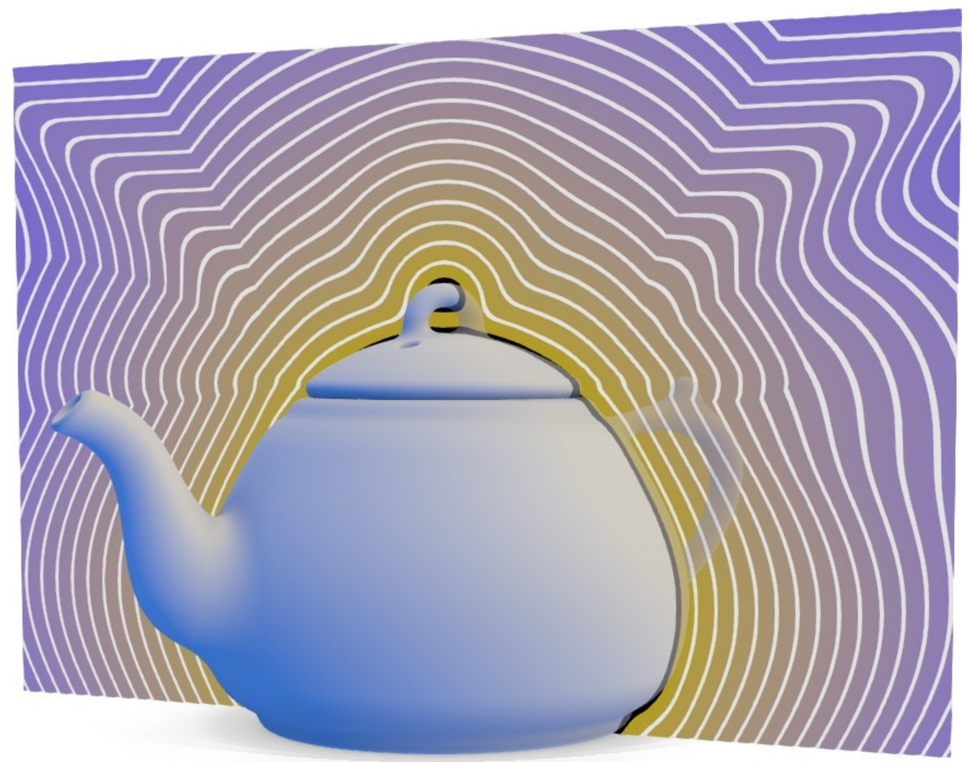
Fields

[Source: Wikipedia]

Geometry with Maths

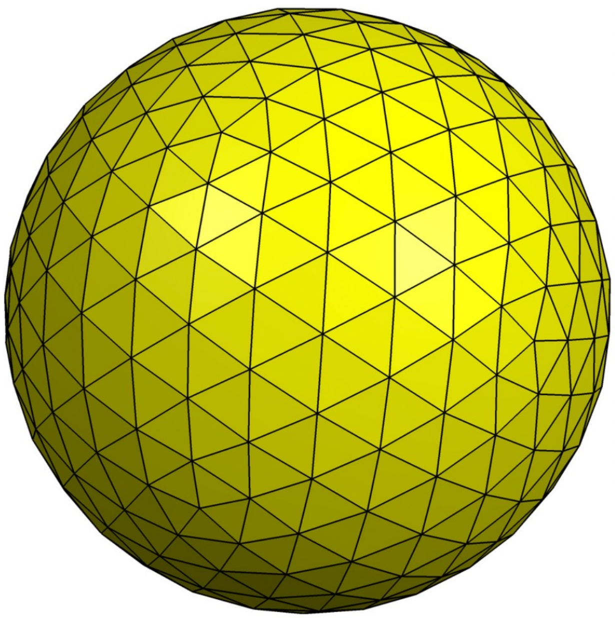
$$f(x, y, z) = d$$

Sometimes also referred to more generally as “Implicit Surfaces”

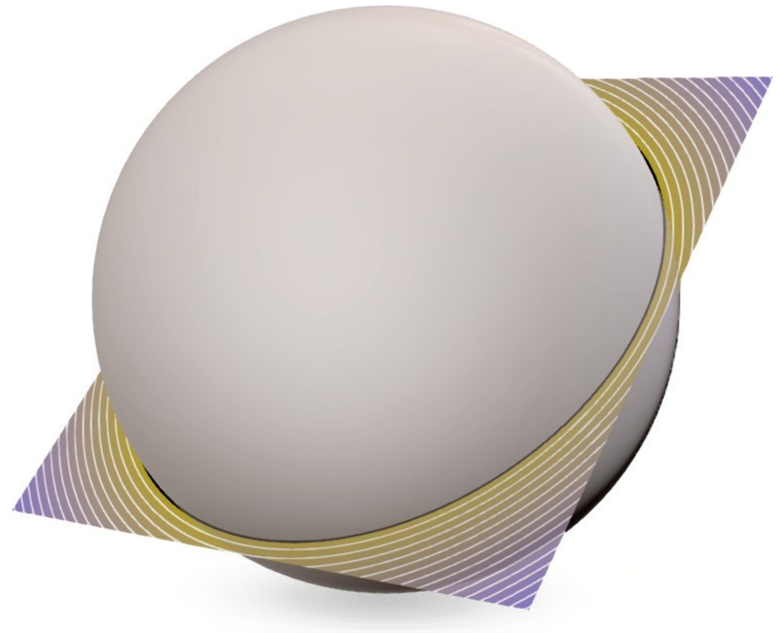


[Source: Takikawa et al]

Signed Distance Functions as Geometry



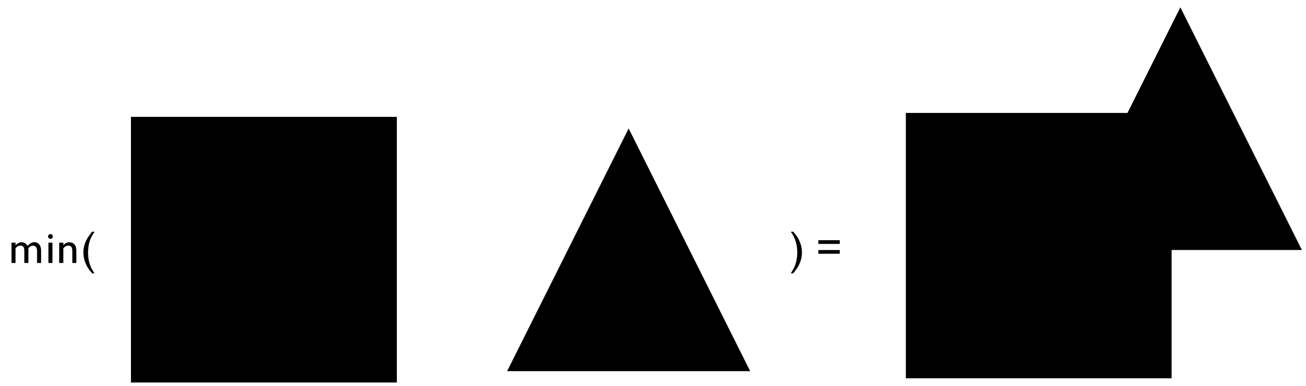
Polygon Mesh



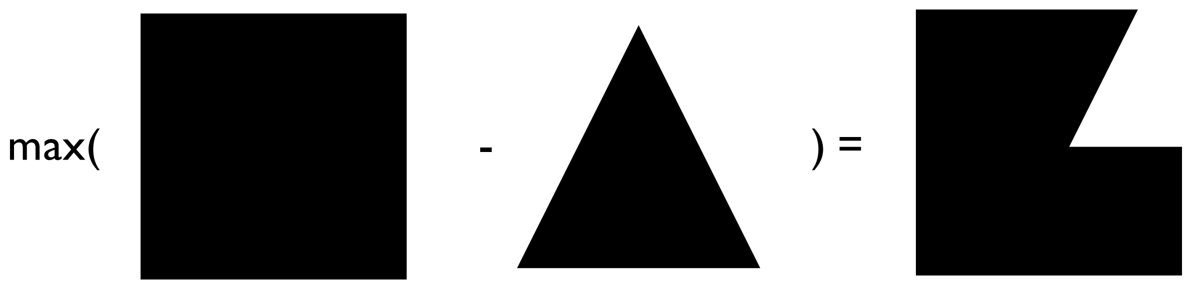
$$f(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 1$$

[Source: Wikipedia et al]

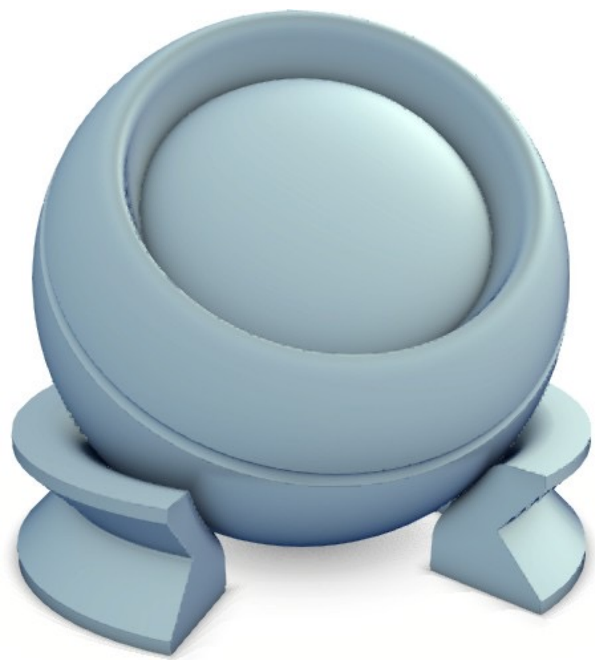
Boolean Operations



Boolean Operations

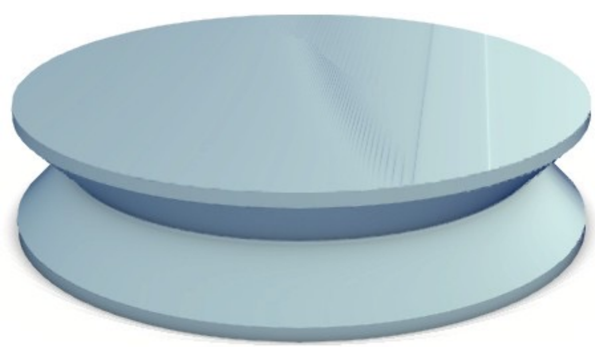


Constructive Solid Geometry



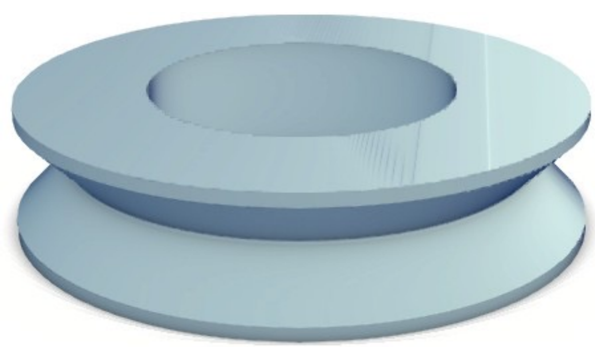
[Source: Takikawa et al]

Constructive Solid Geometry



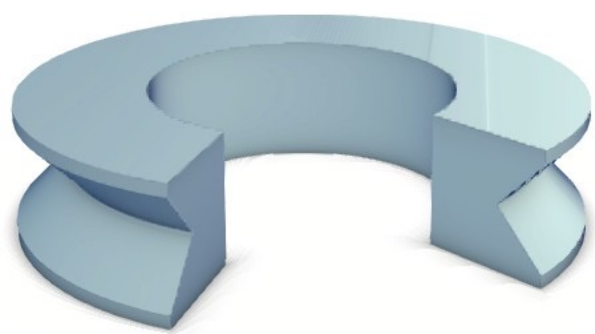
[Source: Takikawa et al]

Constructive Solid Geometry



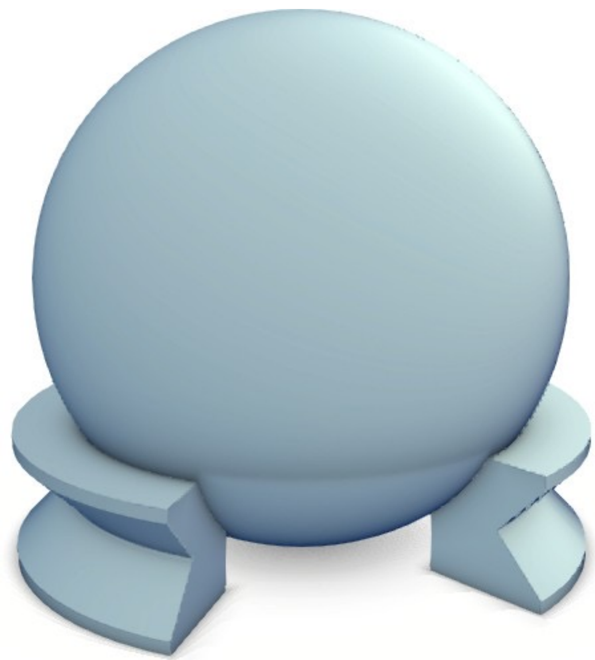
[Source: Takikawa et al]

Constructive Solid Geometry



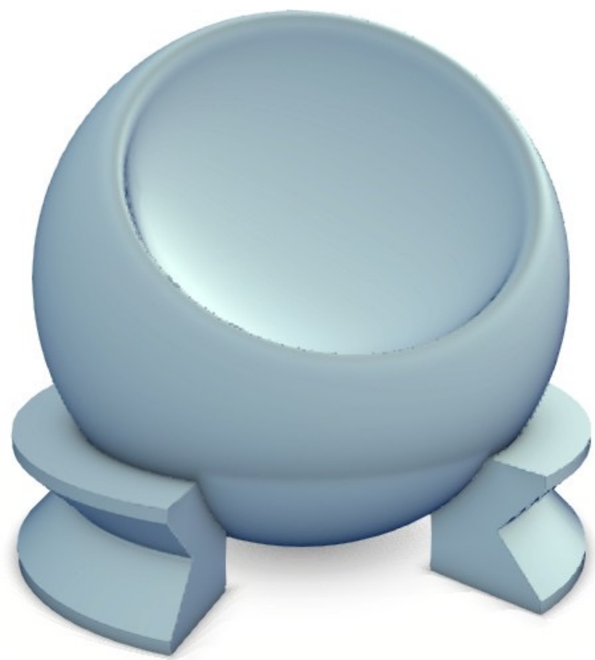
[Source: Takikawa et al]

Constructive Solid Geometry



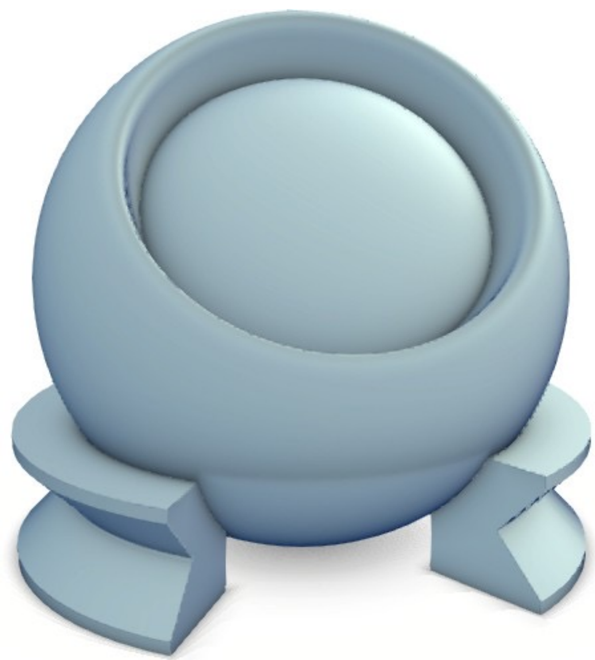
[Source: Takikawa et al]

Constructive Solid Geometry



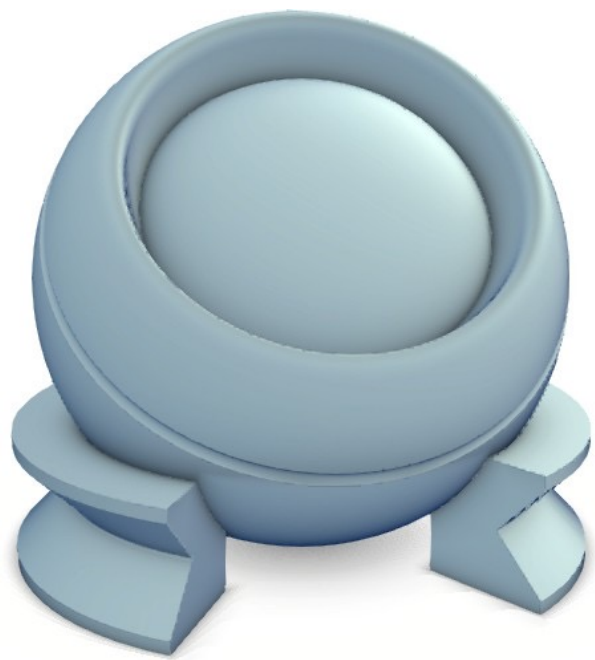
[Source: Takikawa et al]

Constructive Solid Geometry



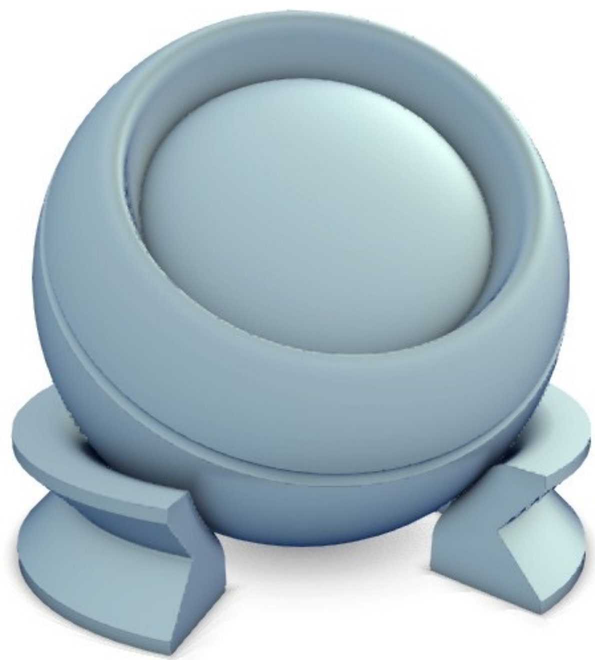
[Source: Takikawa et al]

Constructive Solid Geometry



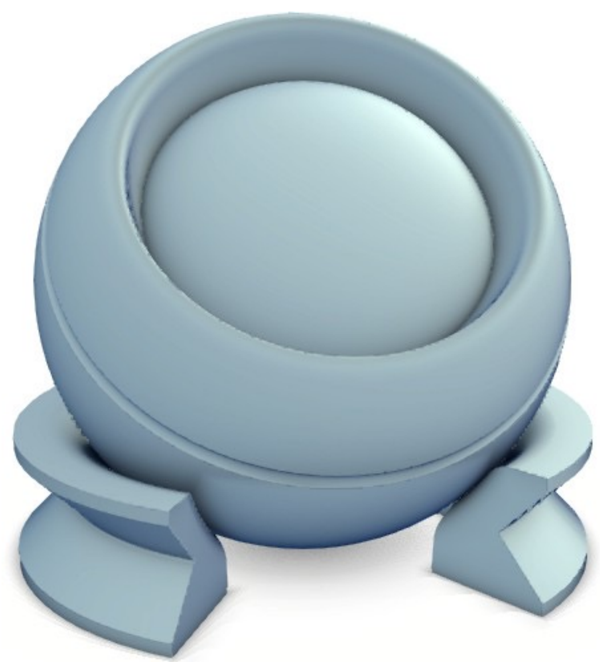
[Source: Takikawa et al]

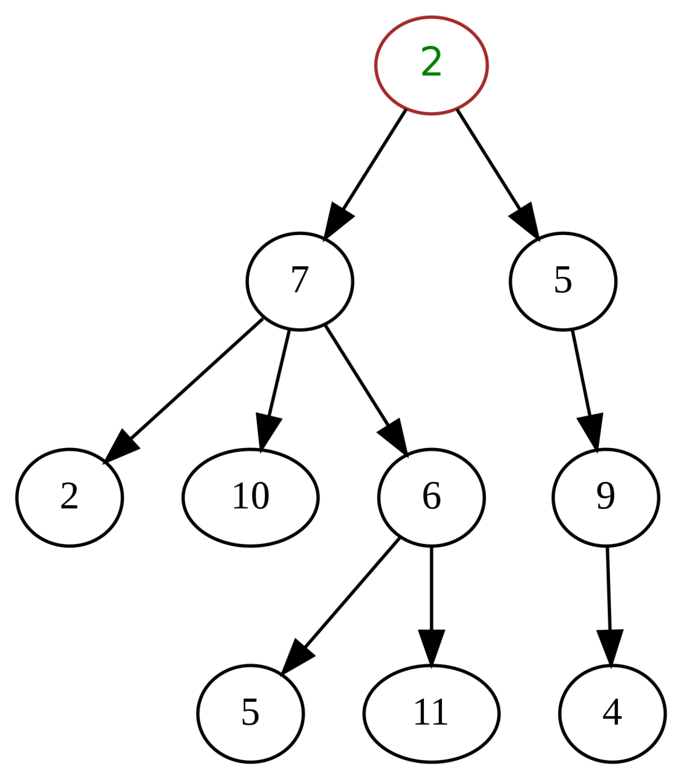
Constructive Solid Geometry



[Source: Takikawa et al]

Function Trees





Big trees are a nightmare to execute and manage!
(ever think about ‘why don’t we just always use linked lists?)

[Source: Wikipedia, Takikawa et al]

Voxel Grids

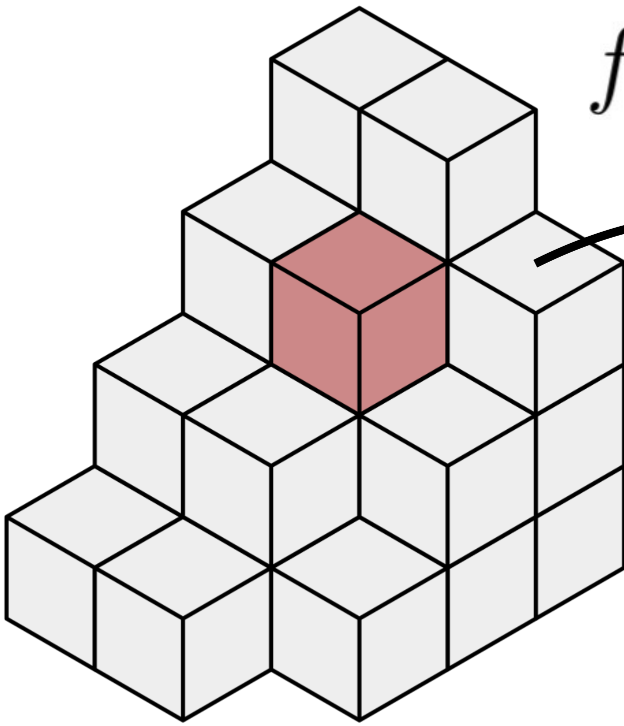
"What if we compute the function once and store the output?"

$$f(x, y, z) = d$$

```
voxel_grid = np.empty([100, 100, 100, N])

# f : R^3 -> R^N
for i, j, k in coordinates:
    voxel_grid[i,j,k] = f(i, j, k)
```

Voxel Grids



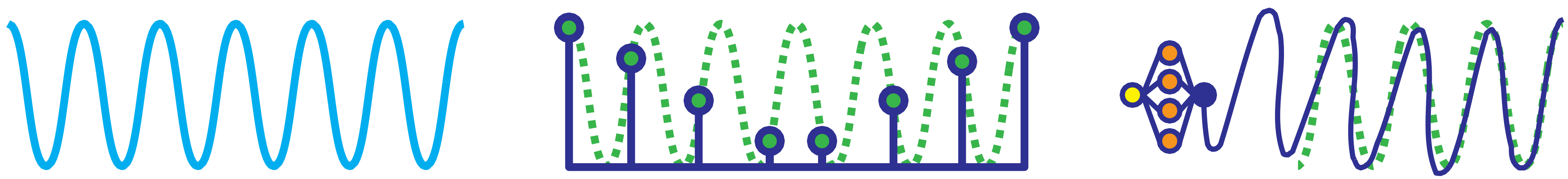
$$f(x, y, z) = d$$

Signed distance, color, density, etc...

[Source: Wikipedia]

Fields and Signals

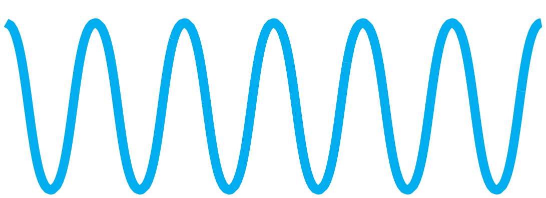
Fields / signals can be represented in many ways.



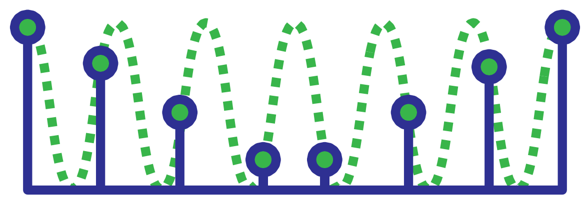
Continuous Discrete Neural

Fields and Signals

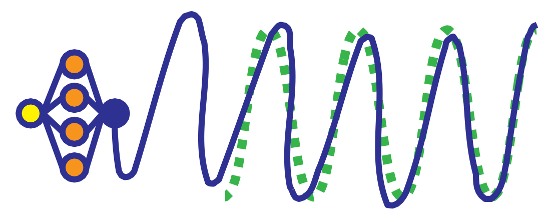
Fields / signals can be represented in many ways.



Continuous

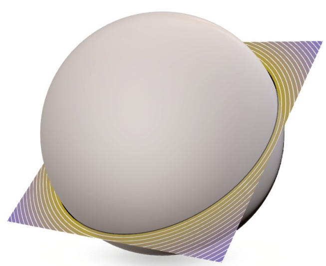


Discrete

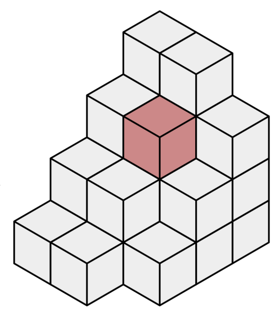


Neural

Smooth, compact, complex

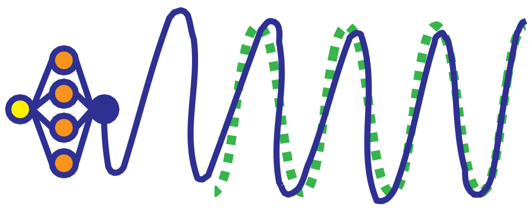


$$f(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 1$$



Simple, bulky, fast

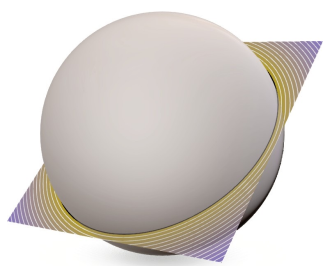
Fields and Signals



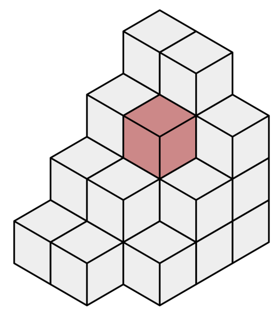
Neural (Fields)

Best of both worlds?

Smooth, compact, complex

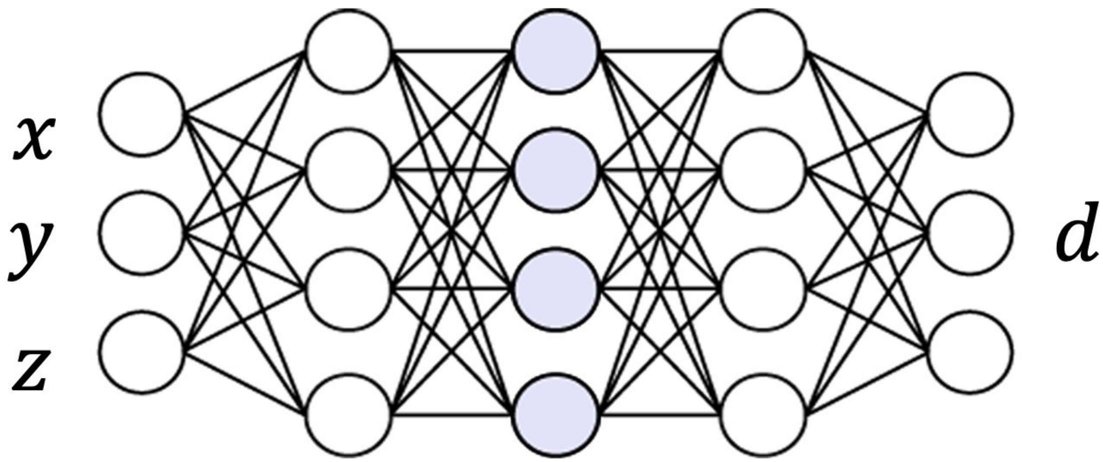


$$f(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 1$$



Simple, bulky, fast

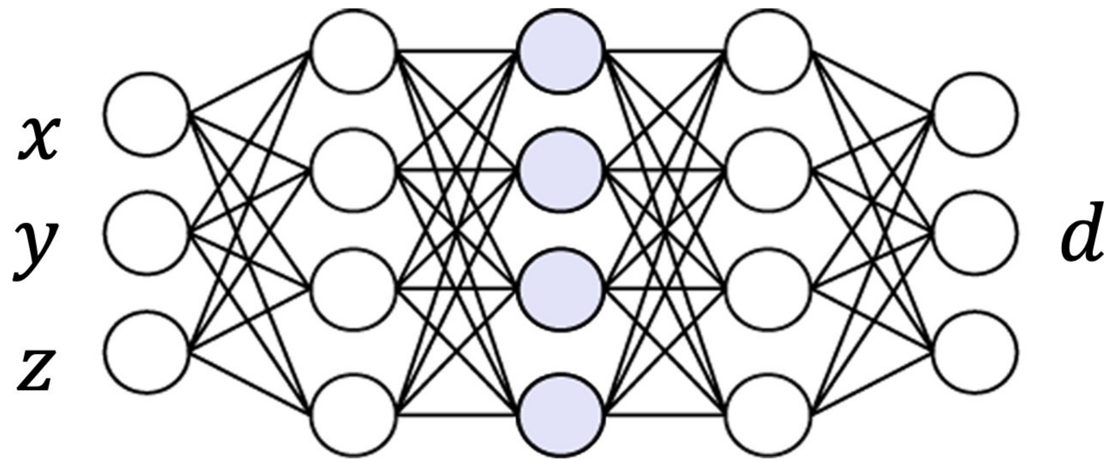
Multi-layer Perceptrons



$$f_\theta(x, y, z) = d$$

Multi-layer Perceptrons

How do you obtain the parameters???

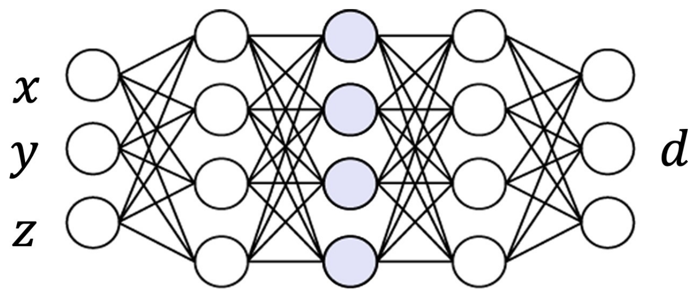


$$f_\theta(x, y, z) = d$$

In practice, can really be any parametric function!

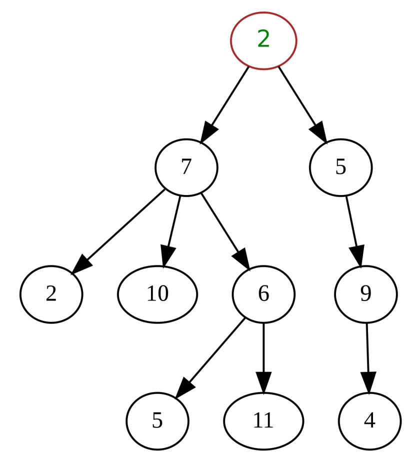
Multi-layer Perceptrons

$\min_{\theta} \Big\|$



$f_{\theta}(x, y, z) = d$

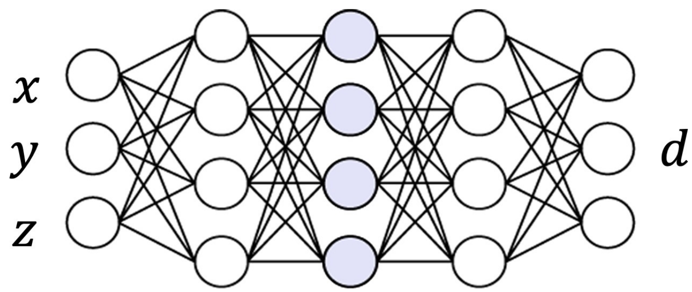
$-$



$\Big\|^{2}$

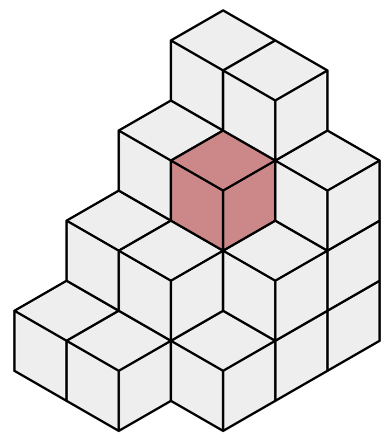
Multi-layer Perceptrons

$\min_{\theta} \Bigg\|$



$f_{\theta}(x, y, z) = d$

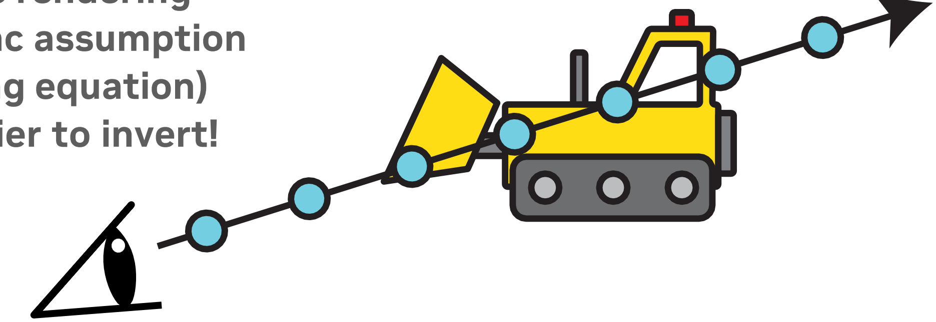
$-$



$\Bigg\|^{2}$

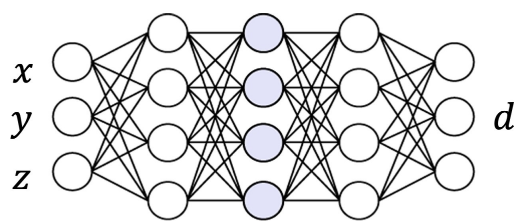
Neural Radiance Fields

NeRF uses volume rendering (i.e. removes the Dirac assumption in Kajiya's rendering equation) to make things easier to invert!



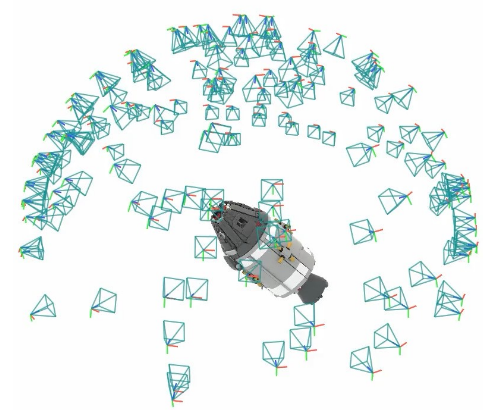
Forward Map (e.g. differentiable rendering)

$$\min_{\theta} \Big\| F\Big($$



$$f_{\theta}(x,y,z) = d$$

$$\Big) - $$

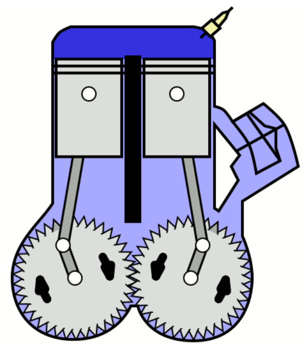


$$\Big\|^{2}$$

Baking via solving inverse problems!

Why Neural Fields? (as a representation)

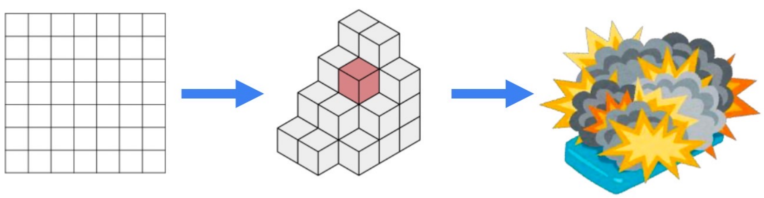
1. Compactness



2. Regularization

$$\mathrm{argmin}_x \|y - F(x)\| + \lambda P(x).$$

3. Domain Agnostic



[Source: Wikipedia]

Neural Fields

Neural Geometric Level of Detail (CVPR 2021):

Sparse Grids + CUDA + NNs = Real-Time Neural Field Rendering