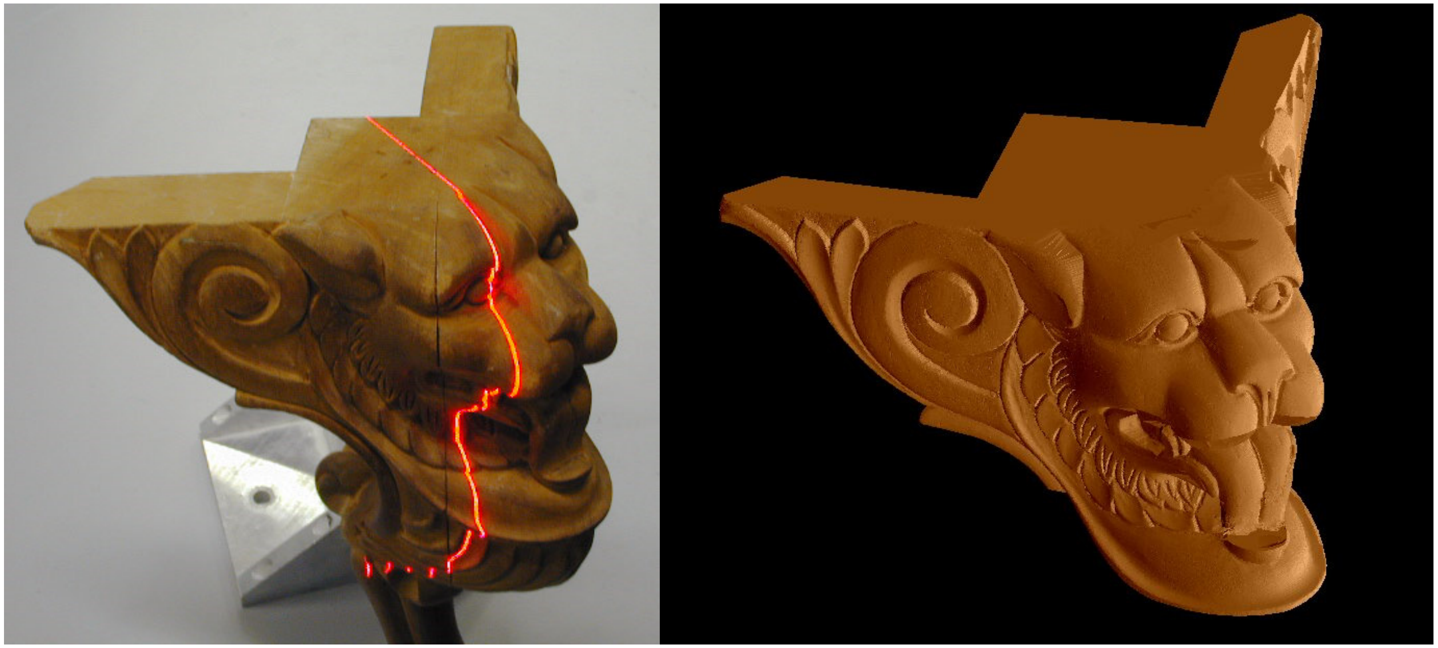
Implicit Surfaces & Reconstruction



Evangelos Kalogerakis – 574/674

Course overview so far

We have seen four families of 3D DL methods for **shape/scene analysis:**

- Multi-View approach
- Volumetric approach
- Point approach
- Graph approach

Course overview so far

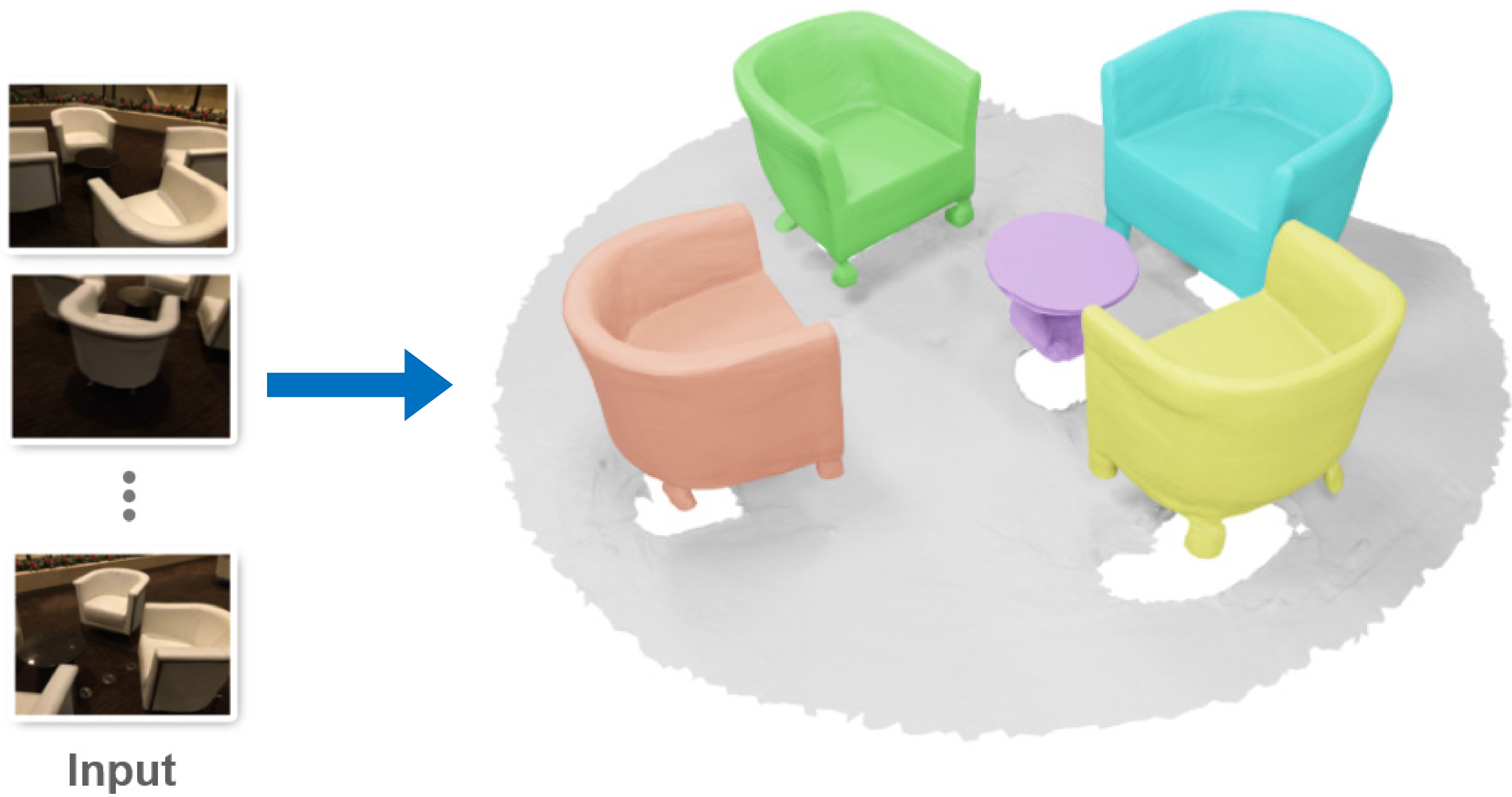
We have seen four families of 3D DL methods for **shape/scene analysis:**

- Multi-View approach
- Volumetric approach
- Point approach
- Graph approach

Let's start talking about **shape/scene generation...**

Examples of Generative tasks

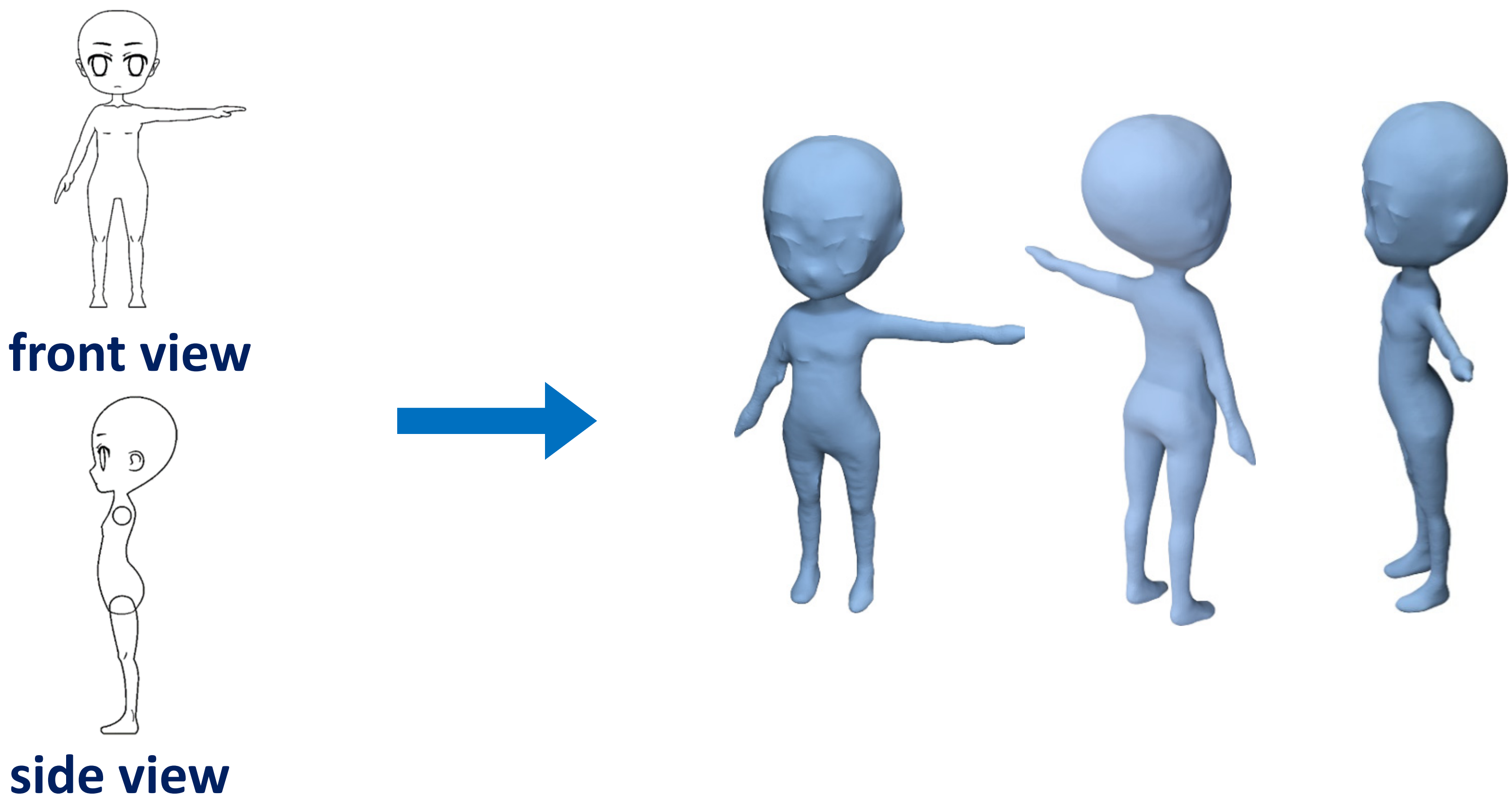
From one or more 2D images to 3D shapes/scenes:



FroDO: From Detections to 3D Objects, CVPR 2020

Examples of Generative tasks

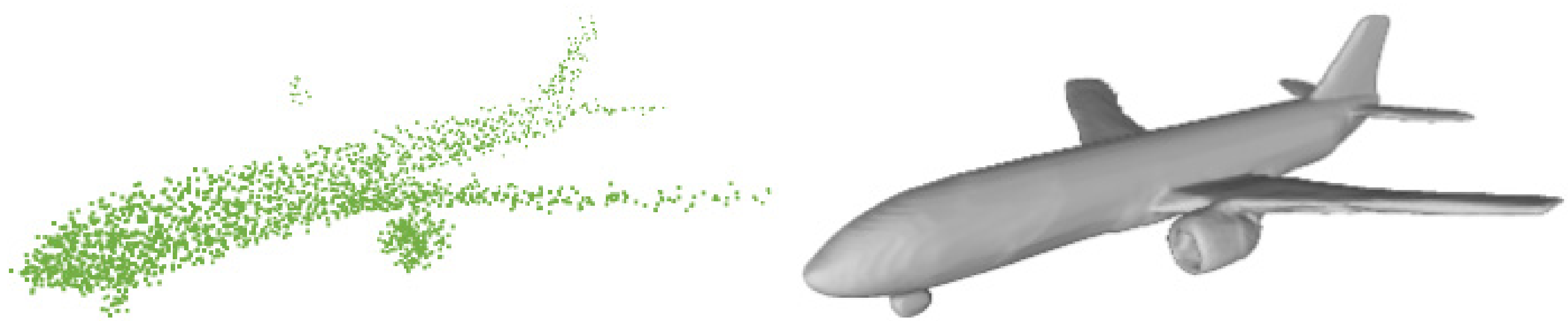
From one or more 2D sketches to 3D shapes/scenes:



3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks, 3DV 2017

Examples of Generative tasks

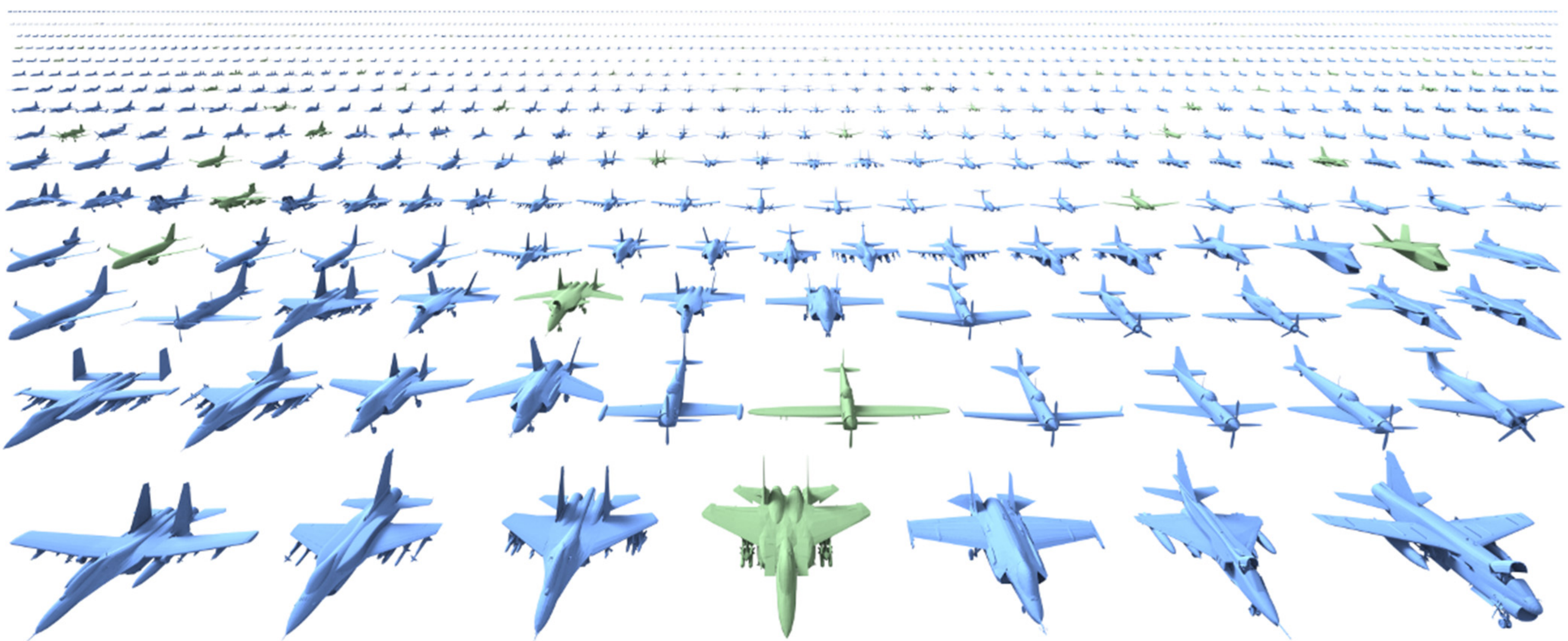
From (noisy) 3D point clouds to 3D shapes/scenes:



DeepSDF, CVPR 2019

Examples of Generative tasks

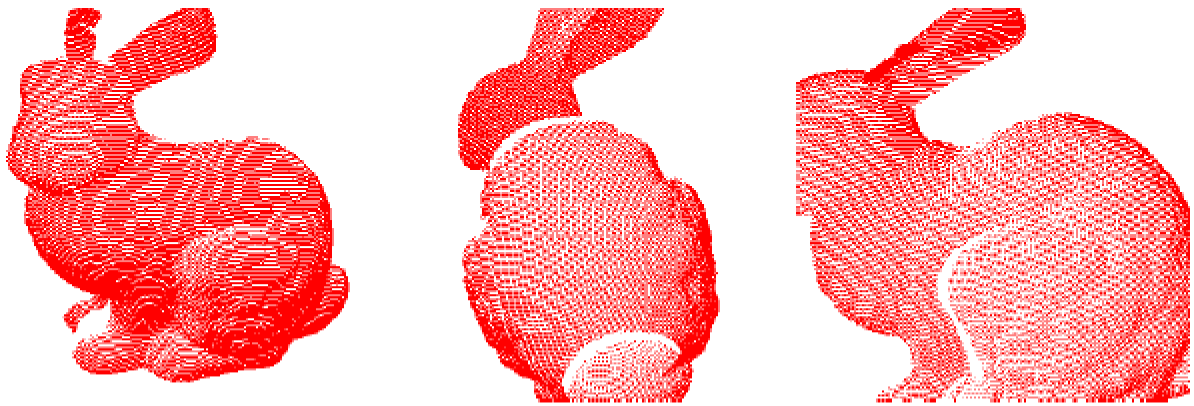
Randomly generate 3D shapes/scenes from a class:



A Probabilistic Model for Component-Based Shape Synthesis , SIGGRAPH 2012

Output Shape Representation

- **Point cloud?**

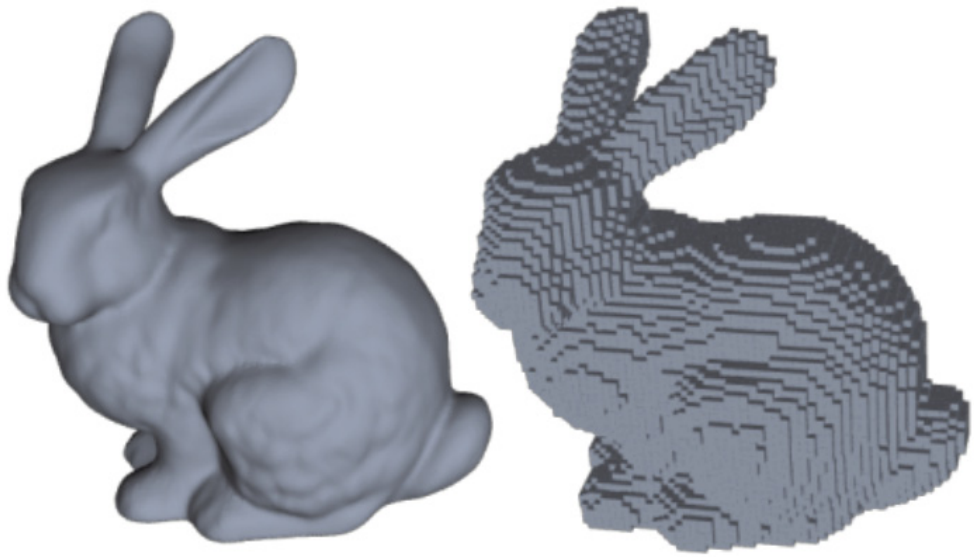


Simplest representation... Raw scanner output, yet:

- Implausible to display in low-resolutions
- No smooth rendering
- No topological information
- Deformation, animation, subdivision are not straightforward

Output Shape Representation

- **Voxels?**

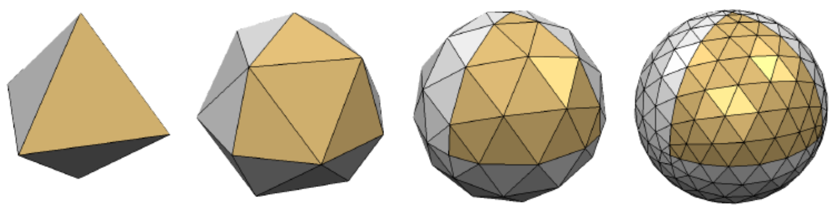


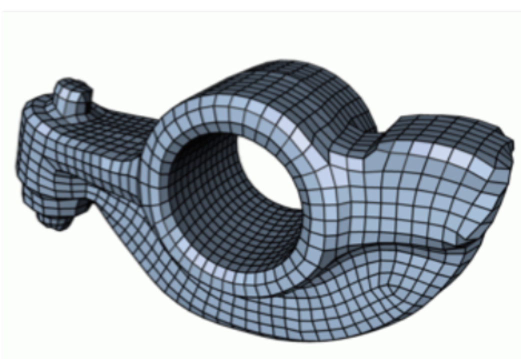
Popular with 3D convolutional networks, yet:

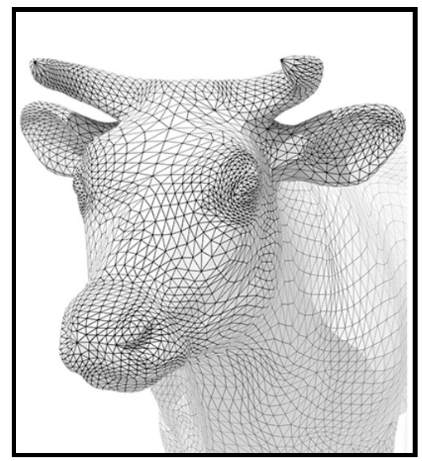
- Implausible to display in low-resolutions
- No smooth rendering
- Provides topological information based on voxel connectivity
- Deformation, animation, subdivision… can be done (unusual in CG)

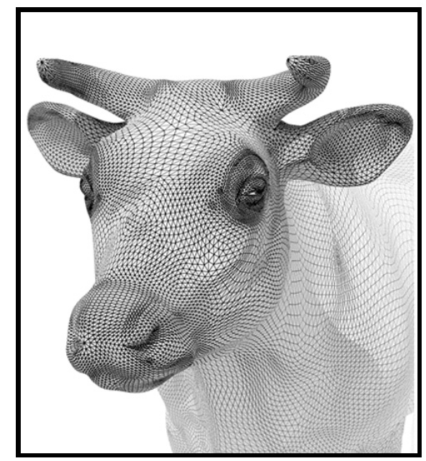
Output Shape Representation

- **Polygon Mesh!**





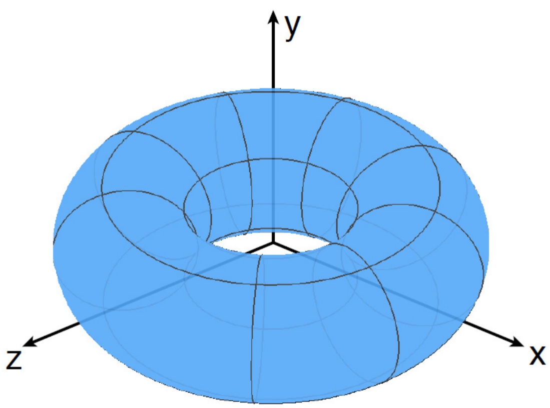




- Surface is piecewise continuous
- Rendering is plausible
- Provides topological information
- Many techniques for mesh deformation, animation, subdivision...

Output Shape Representation

- **Parametric surfaces!**
  A smooth surface created as a map from a 2D set of parameters to 3D space

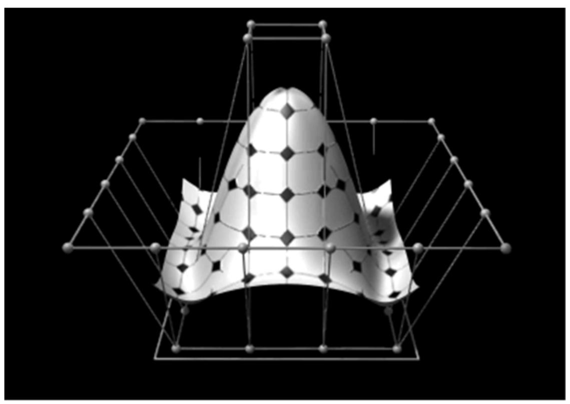


$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$

- Surface is continuous
- Rendering is plausible
- Specific topology
- Can be easily converted to a mesh

Output Shape Representation

- **Parametric surfaces!**
  A smooth surface created as a map from a 2D set of parameters to 3D space

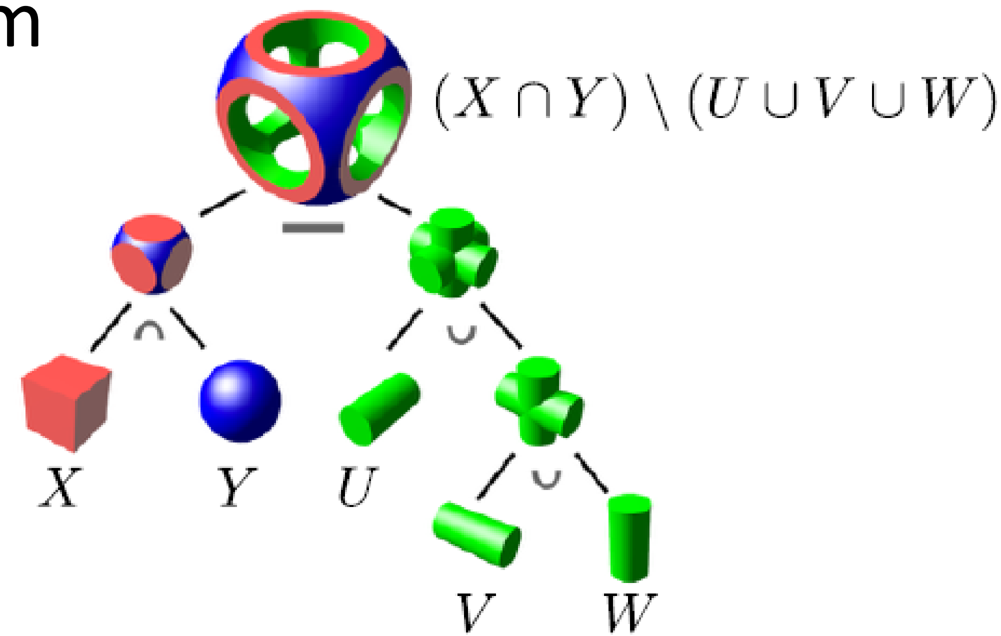


NURBS parametric surface patches
(Non-uniform Rational B-splines)
can be created / edited from a set of control points multiplied by basis functions

- Surface is continuous
- Rendering is plausible
- Specific topology
- Can be easily converted to a mesh

Output Shape Representation

- **Constructive solid geometry!**
  Primitives (spheres, cylinders etc) and boolean combinations of them



- Surface is continuous
- Rendering is plausible
- Topology can be modified by Boolean ops on primitives
- Can be easily converted to a mesh

Output Shape Representation

- **Implicit surfaces!**
  Surface is the zero set of a function in x,y,z:
  $f(x,y,z) = 0$

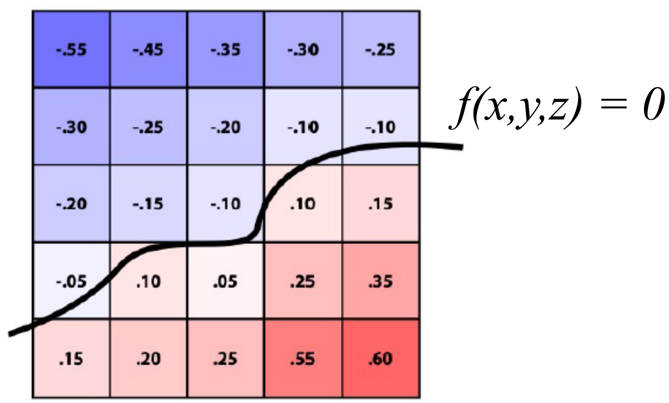
$x^2+y^2+z^2=1$ $\quad$ $(R-\sqrt{x^2+y^2})^2+z^2=r^2$ $\quad$ $(x^2+\frac{9y^2}{4}+z^2-1)^3 = x^2z^3+\frac{9y^2z^3}{80}$

- Surface is continuous
- Rendering is plausible
- Topology can be modified by Boolean ops on implicits
- Can be easily converted to a mesh

Output Shape Representation

- **Implicit surfaces!**
  Surface is the zero set of a function in x,y,z:

$$f(x,y,z) = 0$$

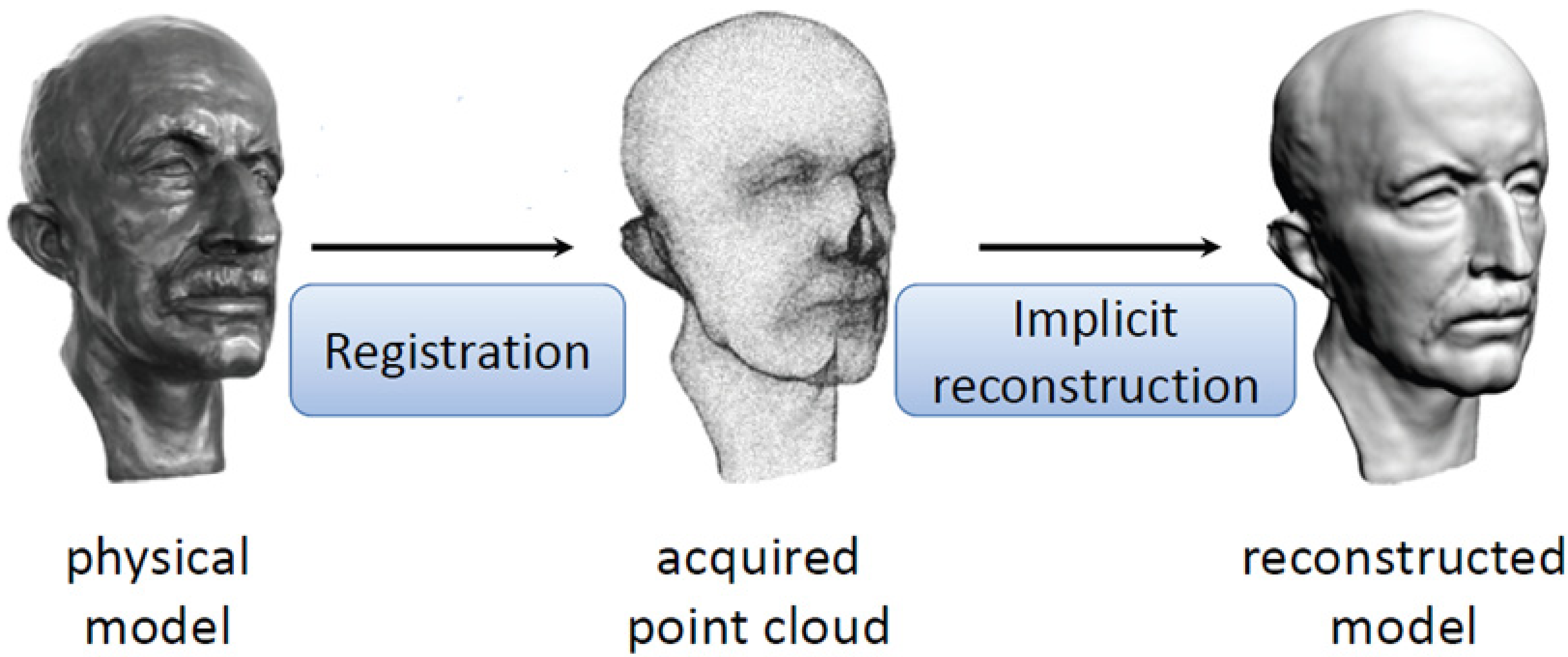


Store a grid of values approximating the function

Surface is defined where interpolated values of this function is 0.

Compared to other representations, it can be easily produced by a neural network operating on 3D grids!

Implicit surfaces are useful for **robust reconstruction of point clouds**

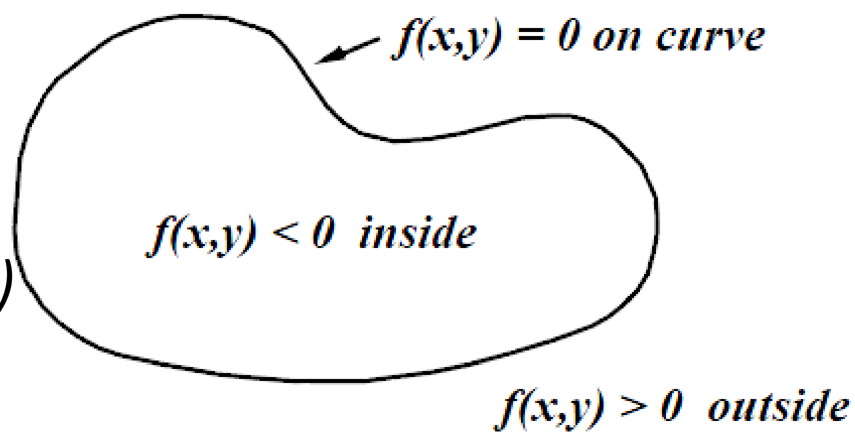


Implicit surfaces in more detail

Curve (in 2D) / surface (in 3D) is defined by **implicit** function:

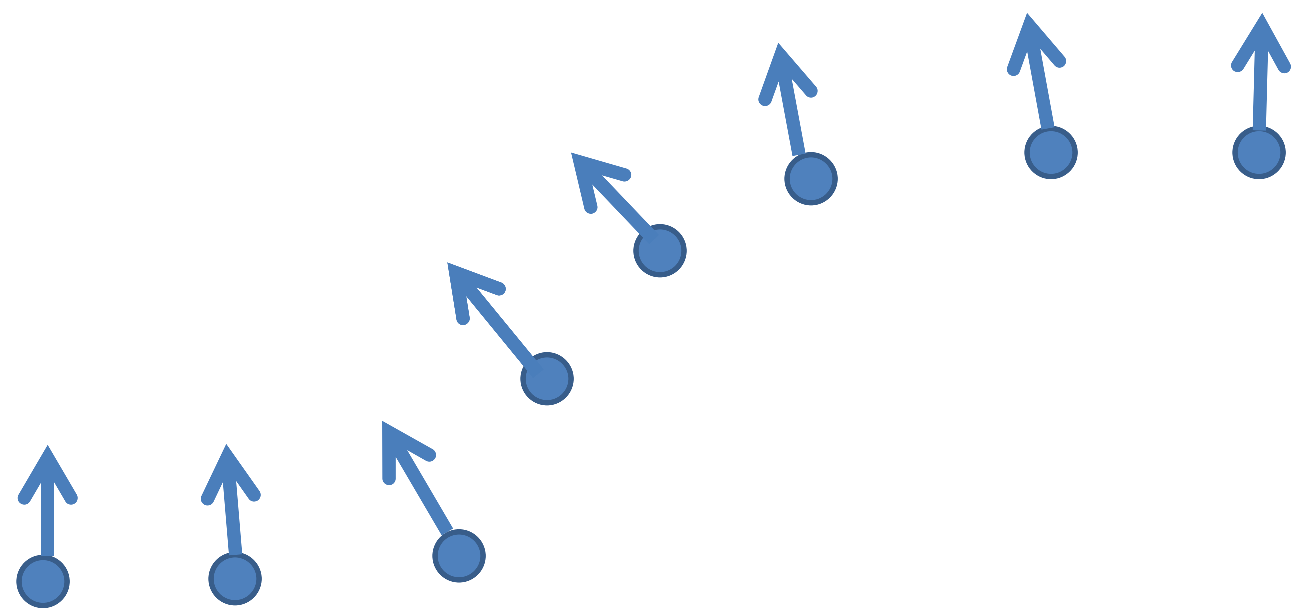
- $f(x,y,z) = 0$ (on surface)
- $f(x,y,z) > 0$ (outside)
- $f(x,y,z) < 0$ (inside)

2-D example (implicit curve)



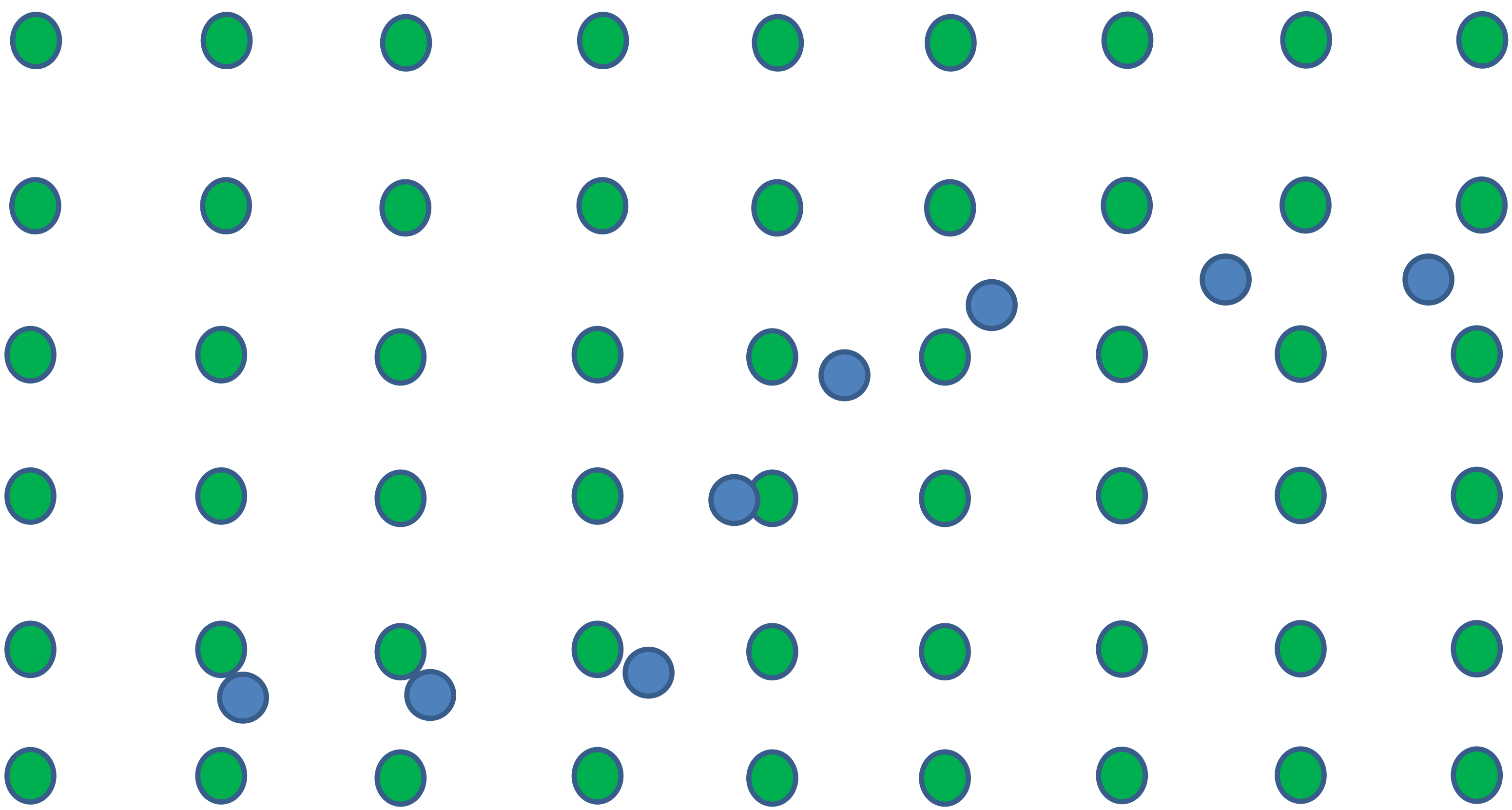
How to define implicit surface for reconstruction?

One way is to represent **signed** distance to the unknown surface represented by points (in blue), their outward-pointing normals are shown as vectors



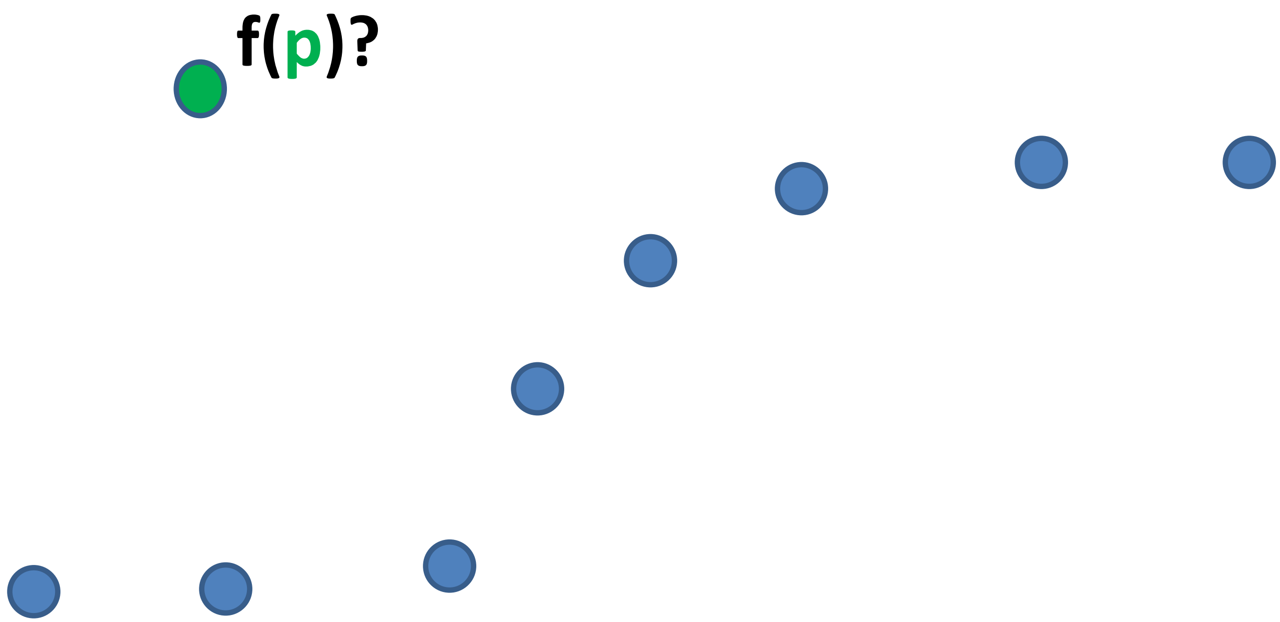
How to define implicit surface for reconstruction?

We will measure the **signed** distance to the unknown surface points in a uniform grid of points (green points)



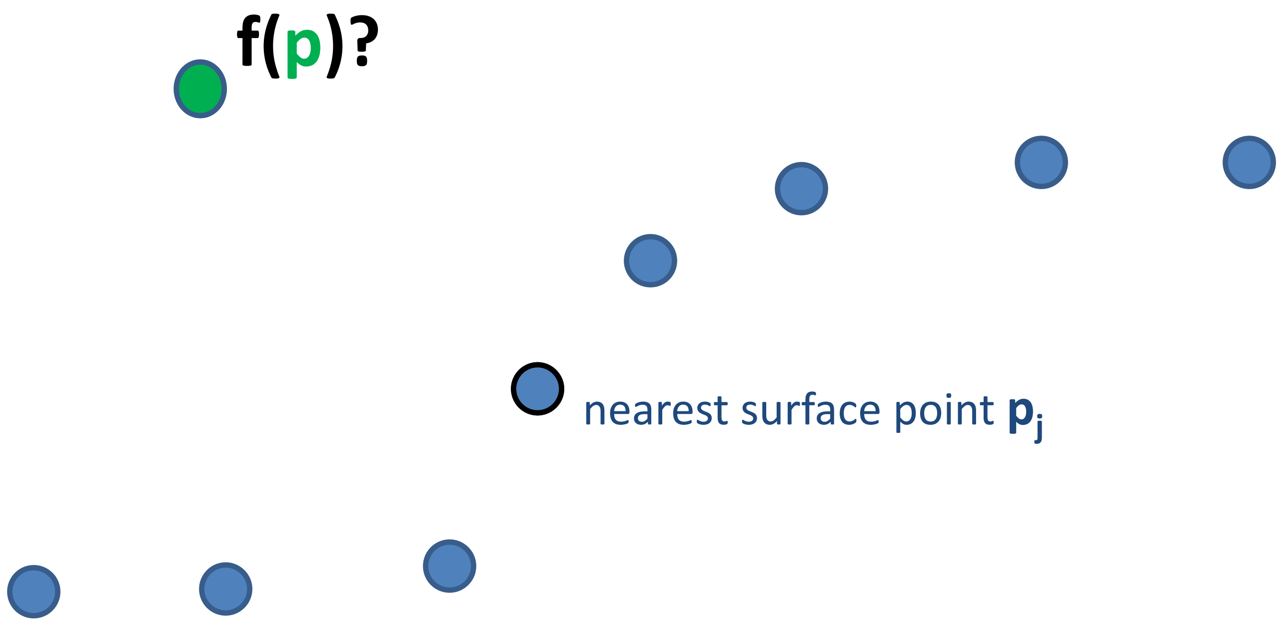
How to define implicit surface for reconstruction?

Let’s focus on computing the **signed** distance f(x,y,z) for a single grid point **p={x,y,z}**:



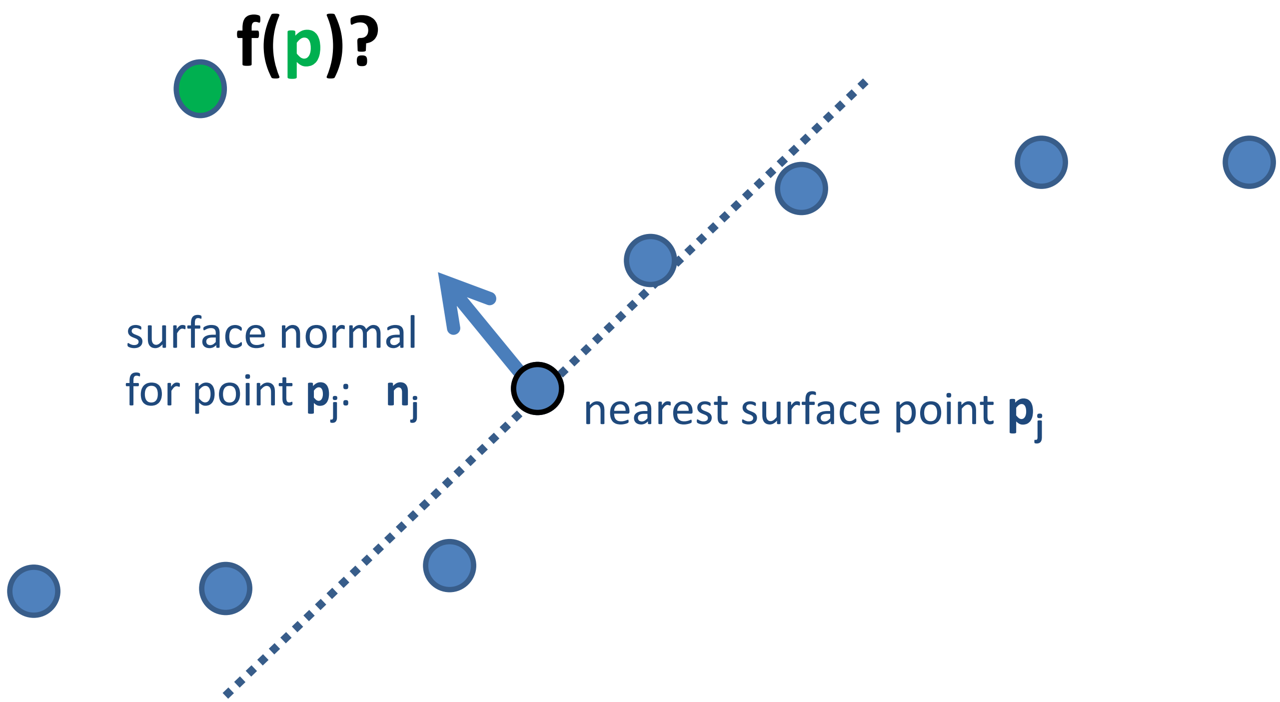
How to define implicit surface for reconstruction?

One way is to find the nearest surface point to the grid point.
Then compute distance of the grid point to the tangent plane of that surface point



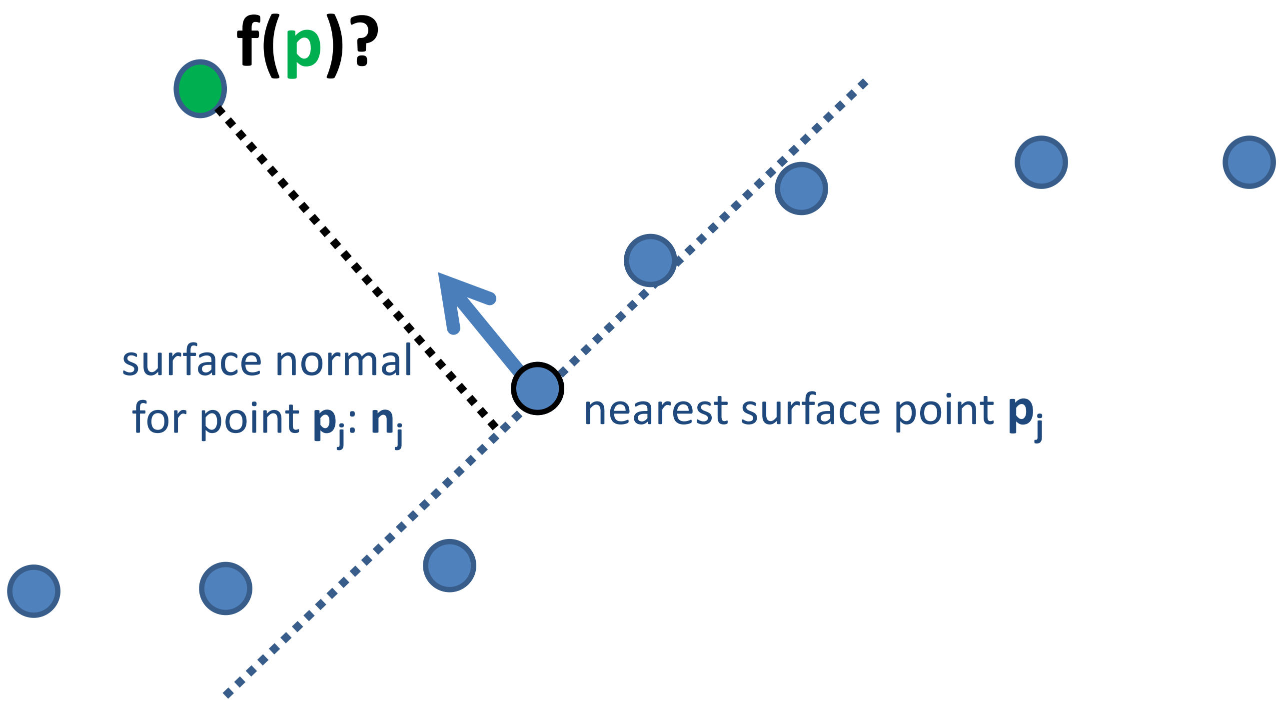
How to define implicit surface for reconstruction?

One way is to find the nearest surface point to the grid point.
Then compute distance of the grid point to the tangent plane of that surface point



How to define implicit surface for reconstruction?

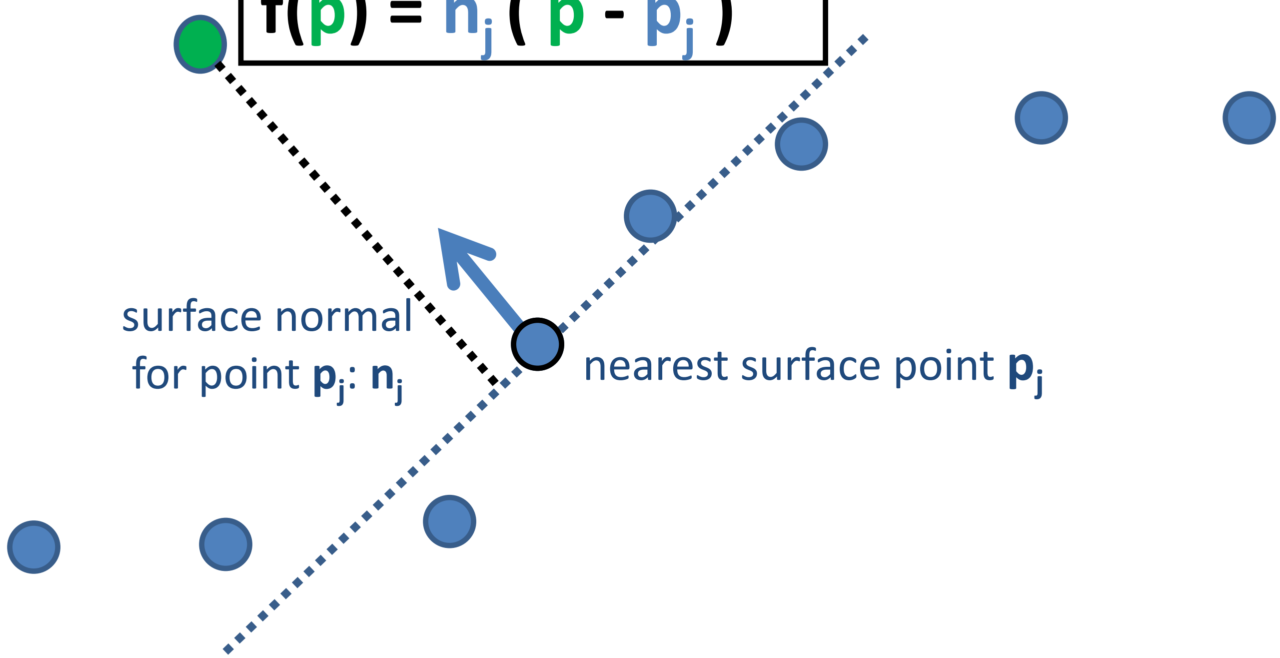
One way is to find the nearest surface point to the grid point.
Then compute distance of the grid point to the tangent plane of that surface point



How to define implicit surface for reconstruction?

One way is to find the nearest surface point to the grid point.
Then compute distance of the grid point to the tangent plane of that surface point

$$f(\mathbf{p}) = \mathbf{n}_j \, (\, \mathbf{p} - \mathbf{p}_j \,)$$



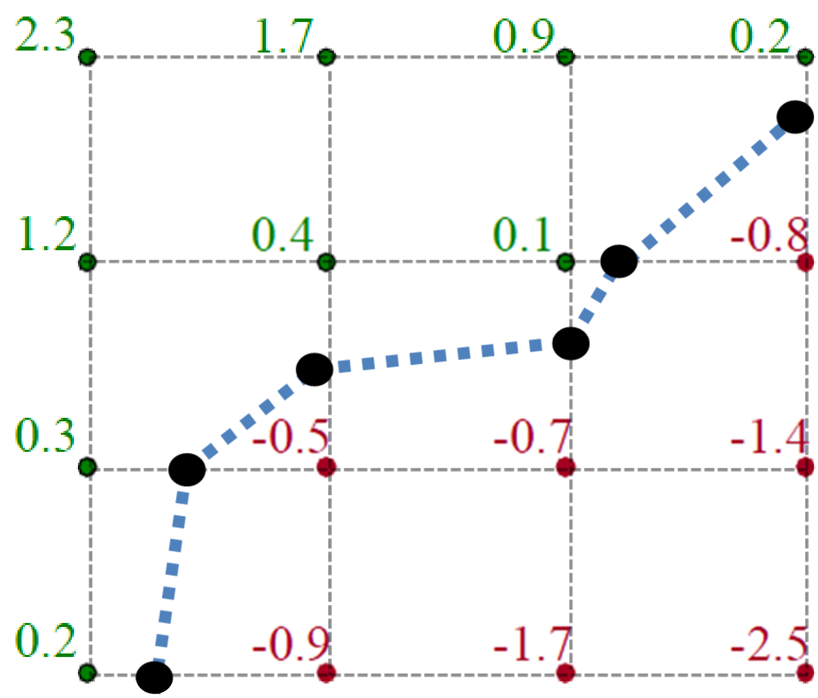
How to define implicit surface for reconstruction?

As a result, we get the implicit function measuring distance at a uniform grid

2.3	1.7	0.9	0.2
1.2	0.4	0.1	-0.8
0.3	-0.5	-0.7	-1.4
0.2	-0.9	-1.7	-2.5

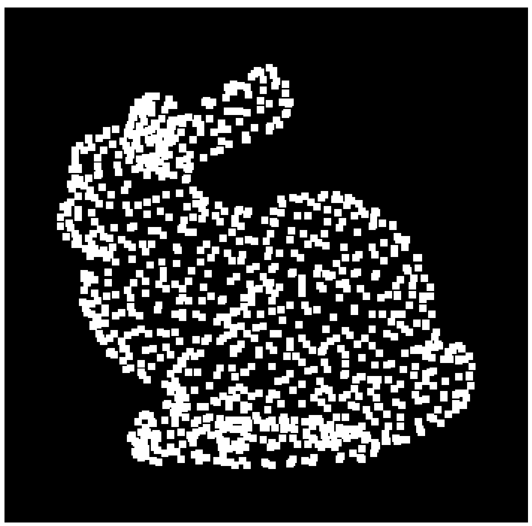
How to define implicit surface for reconstruction?

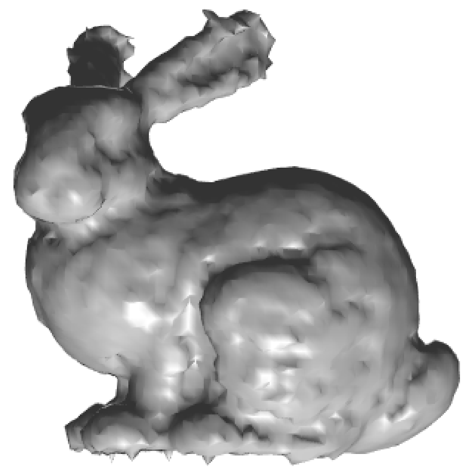
Linear interpolation to get a piecewise linear approximation of the surface



How to define implicit surface for reconstruction?

However, this implicit function definition is not good – the function changes abruptly along the query points of the grid (when their nearest points change)



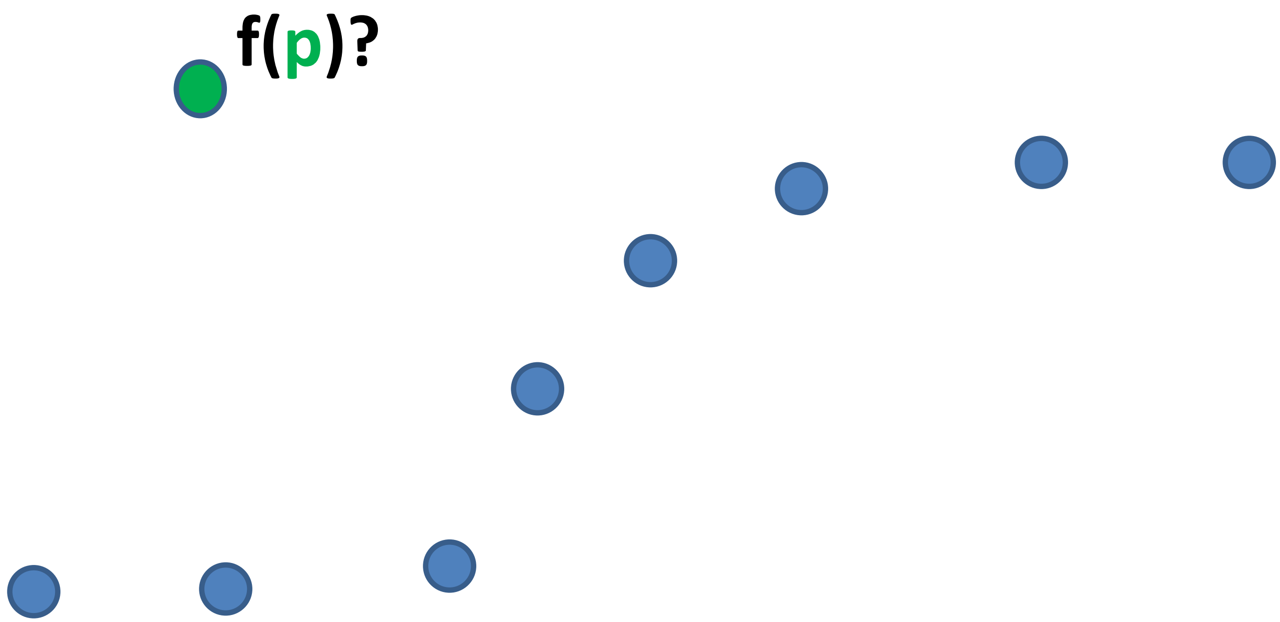


Better Implicit Reconstruction

- **Moving Least Squares**
- RBF/NN interpolation (next time)
- Isosurface extraction (next time)

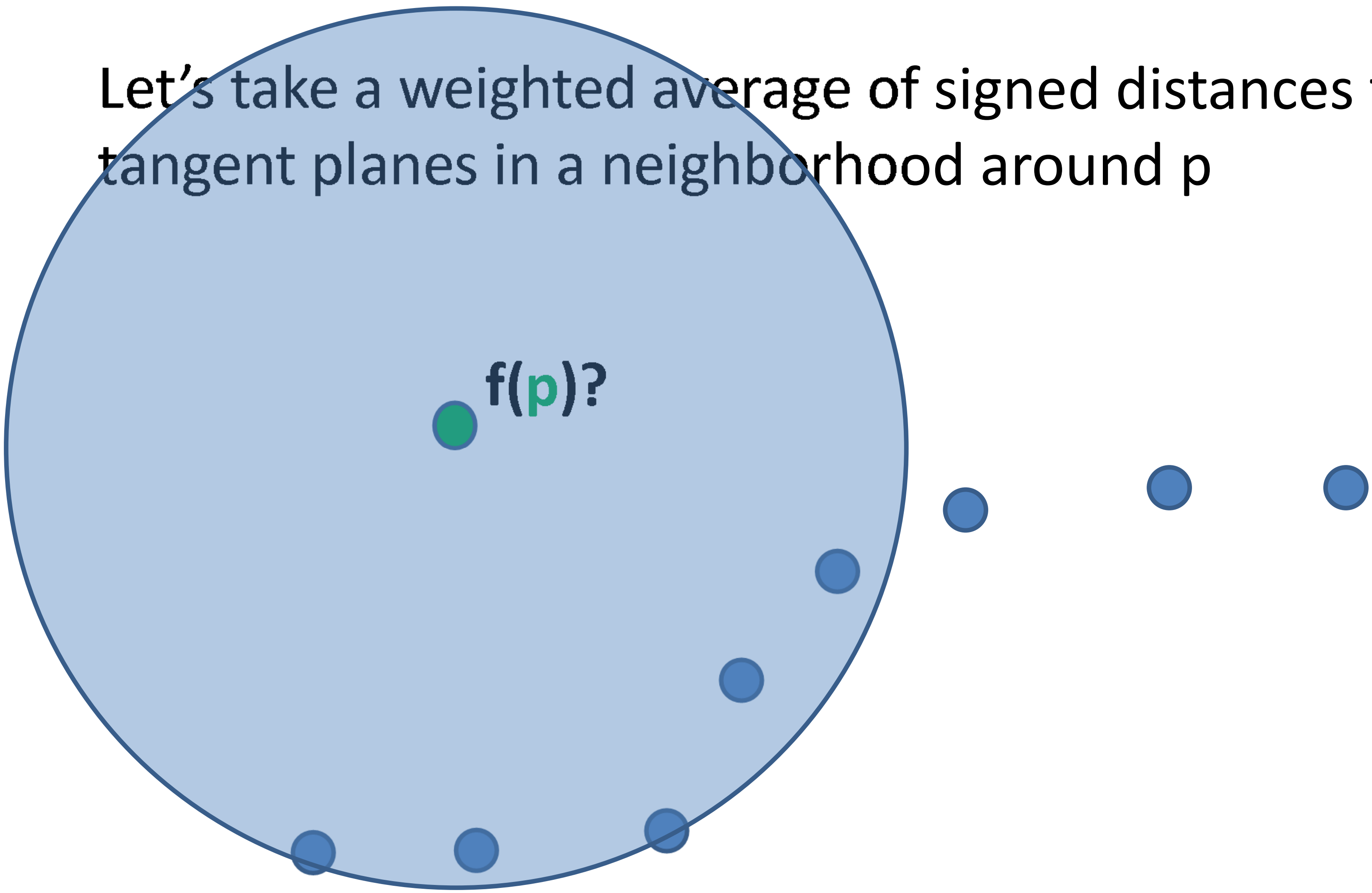
Moving Least Squares

Let's take a weighted average of signed distances to tangent planes in a neighborhood around p



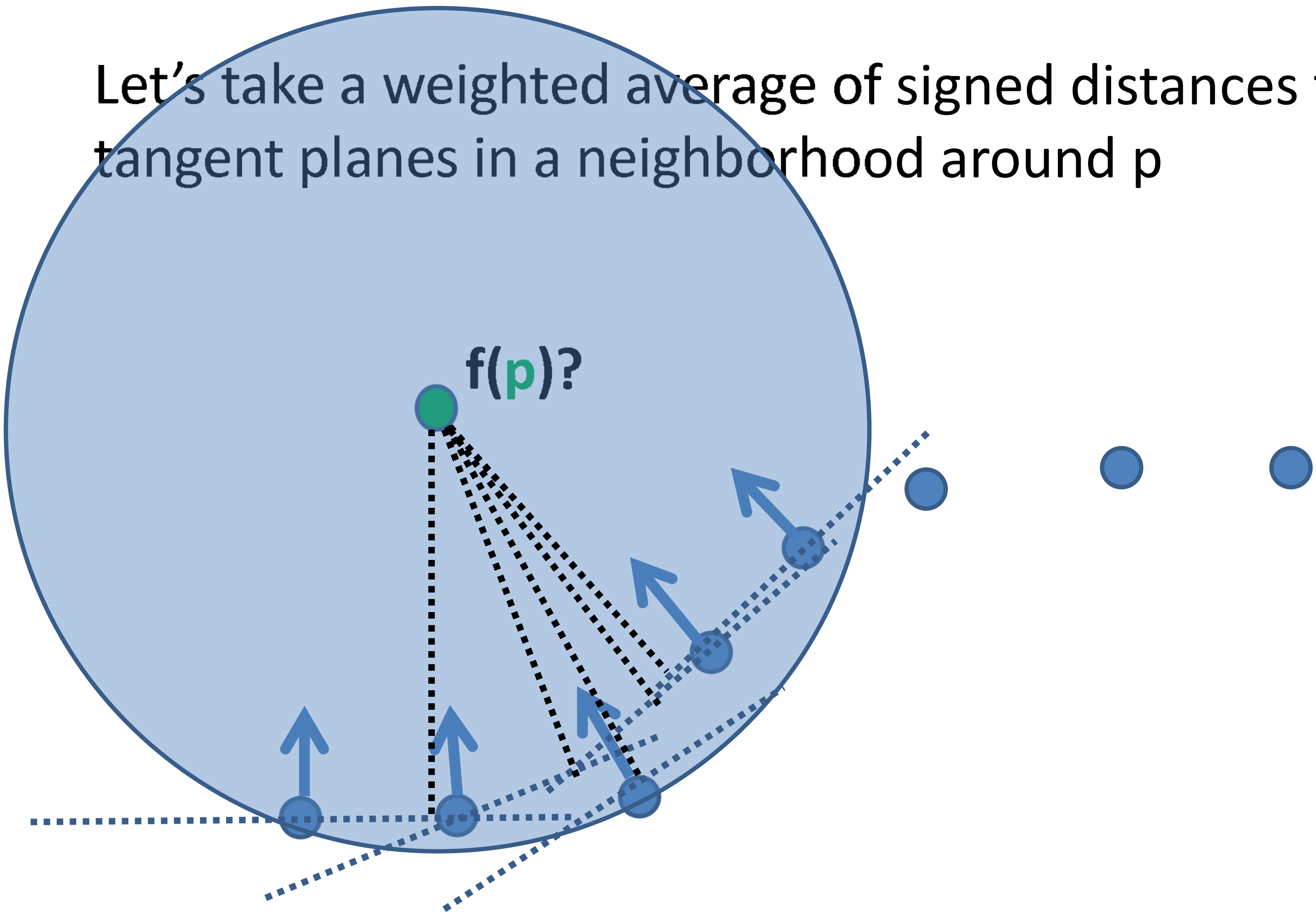
Moving Least Squares

Let’s take a weighted average of signed distances to tangent planes in a neighborhood around p



Moving Least Squares

Let's take a weighted average of signed distances to tangent planes in a neighborhood around p



Moving Least Squares

Take a weighted average of all signed distances to tangent planes

$$f(\mathbf{p}) = \frac{\sum_i \mathbf{n_i}(\mathbf{p} - \mathbf{p_i})\varphi(\| \mathbf{p} - \mathbf{p_i} \|)}{\sum_i \varphi(\| \mathbf{p} - \mathbf{p_i} \|)}$$

$$\varphi(\| \mathbf{p} - \mathbf{p_i} \|) = \exp(- \| \mathbf{p} - \mathbf{p_i} \|^2 / \sigma^2)$$

Mesh-independent surface interpolation, 2003

Moving Least Squares

Take a weighted average of all signed distances to tangent planes

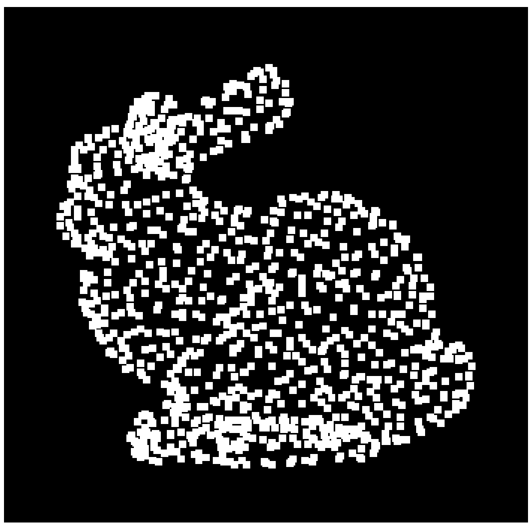
$$f(\mathbf{p}) = \frac{\sum_i \mathbf{n_i}(\mathbf{p} - \mathbf{p_i})\varphi(\|\mathbf{p} - \mathbf{p_i}\|)}{\sum_i \varphi(\|\mathbf{p} - \mathbf{p_i}\|)}$$

$$\varphi(\|\mathbf{p} - \mathbf{p_i}\|) = \exp(-\|\mathbf{p} - \mathbf{p_i}\|^2 / \boxed{\sigma^2})$$

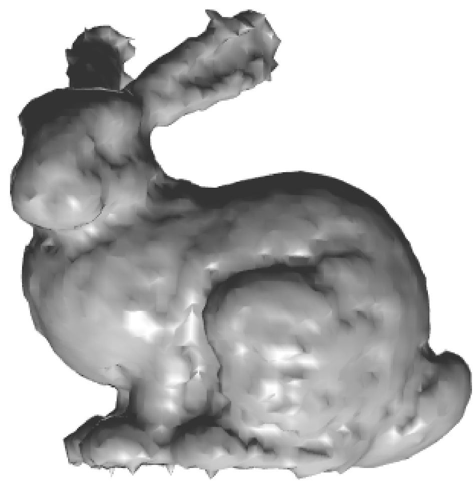
controls weight decay:
Set σ to average distance of each surface point to its nearest surface point multiplied by a small factor (like 2.0)

Moving Least Squares

Point cloud (with normals)



SDF based on tangent plane of nearest point



SDF based on MLS