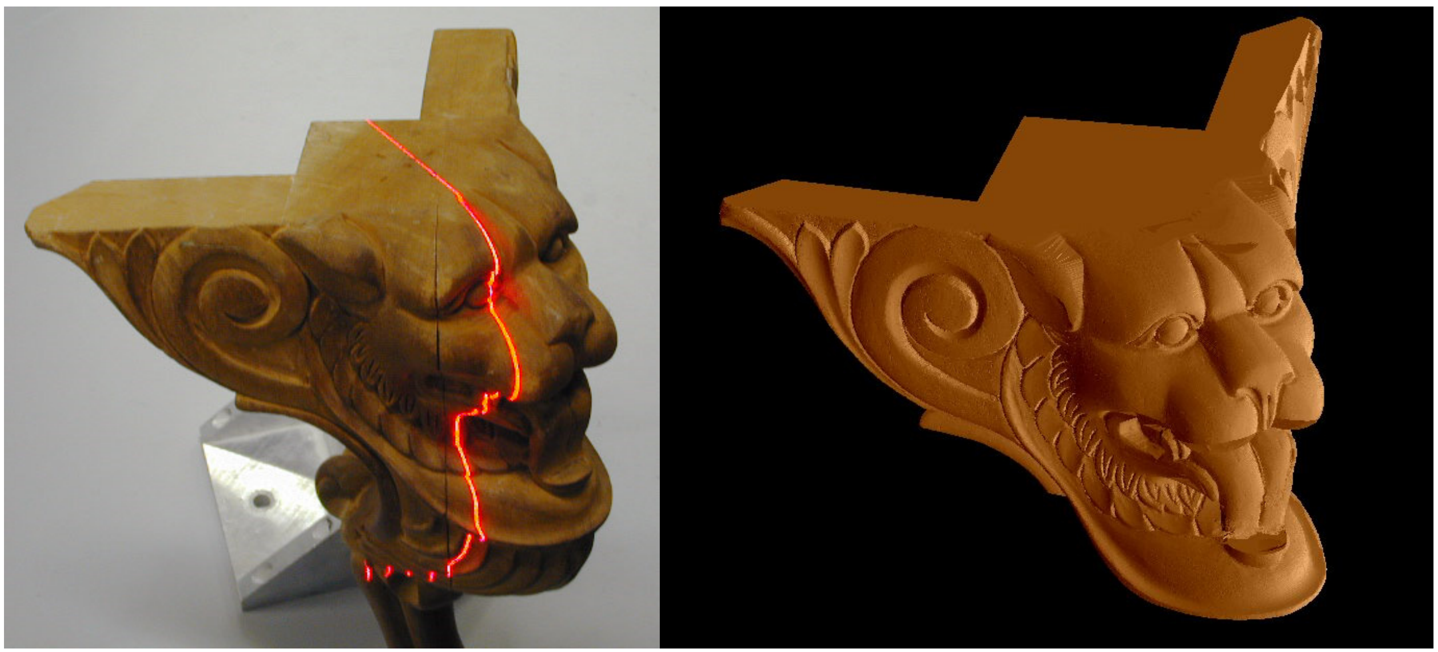
Implicit Surfaces
& Reconstruction



Evangelos Kalogerakis –
574/674

Implicit Reconstruction – what we'll see today

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

RBF Reconstruction

“Fit” an implicit function expressing the distance to the underlying surface. Let’s define an implicit function as:

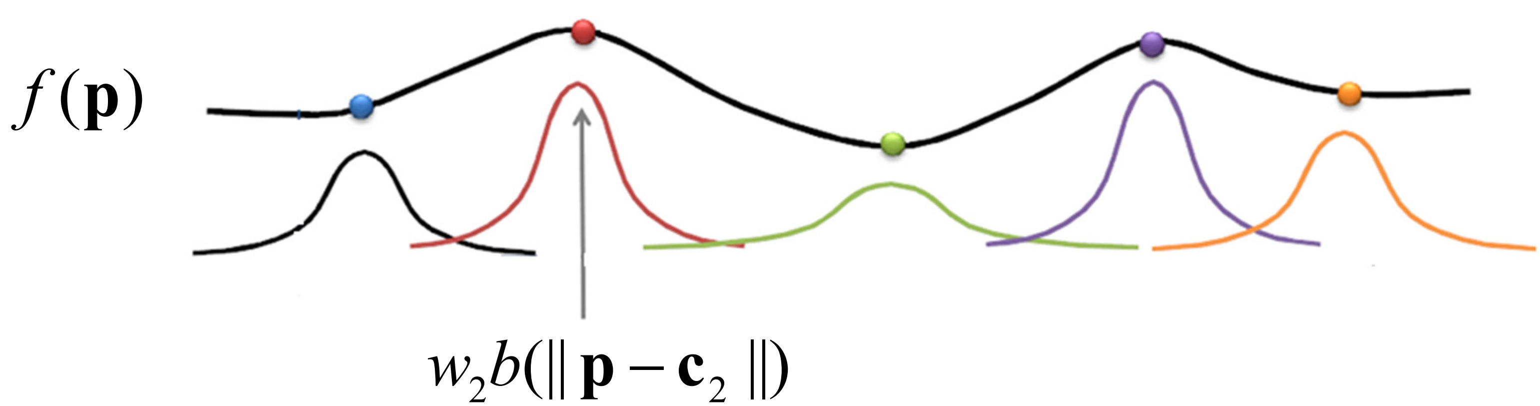
$$f(\mathbf{p}) = \sum_k w_{\mathbf{k}} b(\| \mathbf{p} - \mathbf{c}_{\mathbf{k}} \|)$$

“Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001

RBF Reconstruction

"Fit" an implicit function expressing the distance to the underlying surface. Let's define an implicit function as:

$$f(\mathbf{p}) = \sum_k w_\mathbf{k} b(\| \mathbf{p} - \mathbf{c_k} \|)$$

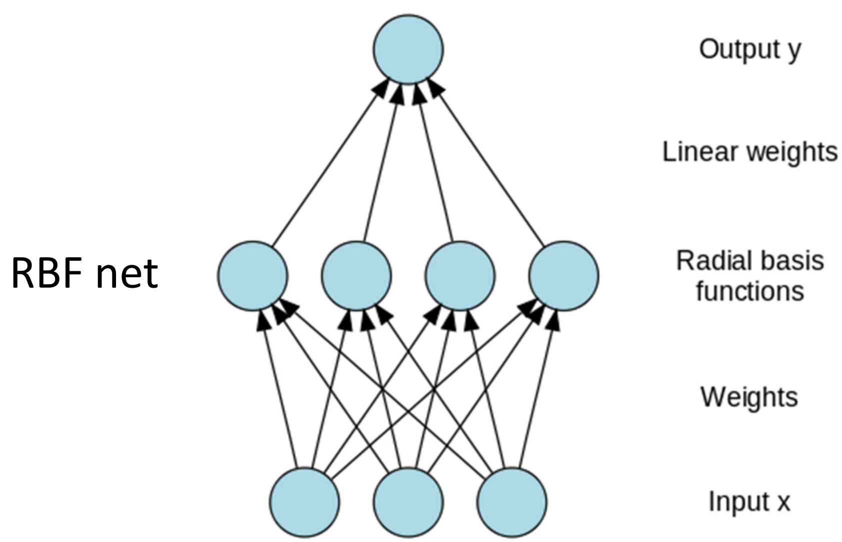


"Reconstruction and Representation of 3D Objects with Radial Basis Functions", SIGGRAPH 2001

RBF Reconstruction

“Fit” an implicit function expressing the distance to the underlying surface. Let’s define an implicit function as:

$$f(\mathbf{p}) = \sum_k w_{\mathbf{k}} b(\| \mathbf{p} - \mathbf{c}_{\mathbf{k}} \|)$$

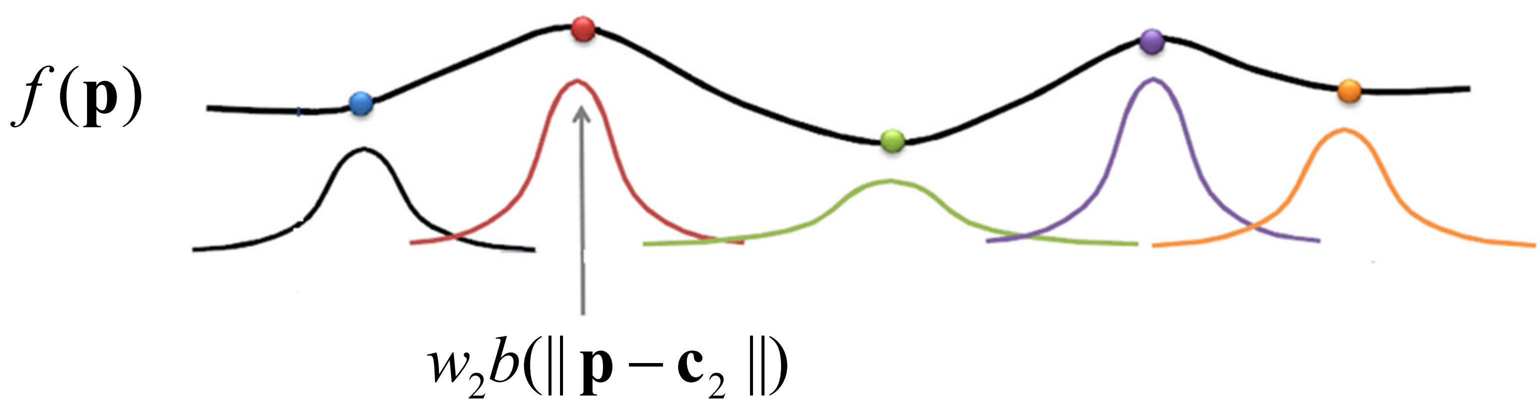


“Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001

RBF Reconstruction

How do we define weights $\mathbf{w_k}$? How do we define centers $\mathbf{c_k}$? How do we define basis functions $\mathbf{b(\cdot)}$?

$$f(\mathbf{p}) = \sum_k w_{\mathbf{k}} b(\| \mathbf{p} - \mathbf{c_k} \|)$$



“Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001

Basis functions

In general, basis functions are functions that increase or decrease **smoothly** with distance to centers

$r = ||\mathbf{p} - \mathbf{c}_k||$

- Gaussian:

$$b(r) = e^{-(\varepsilon r)^2}$$

- Multiquadric:

$$b(r) = \sqrt{1 + (\varepsilon r)^2}$$

- Inverse quadratic:

$$b(r) = \frac{1}{1 + (\varepsilon r)^2}$$

- Inverse multiquadric:

$$b(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}$$

- Polyharmonic spline:

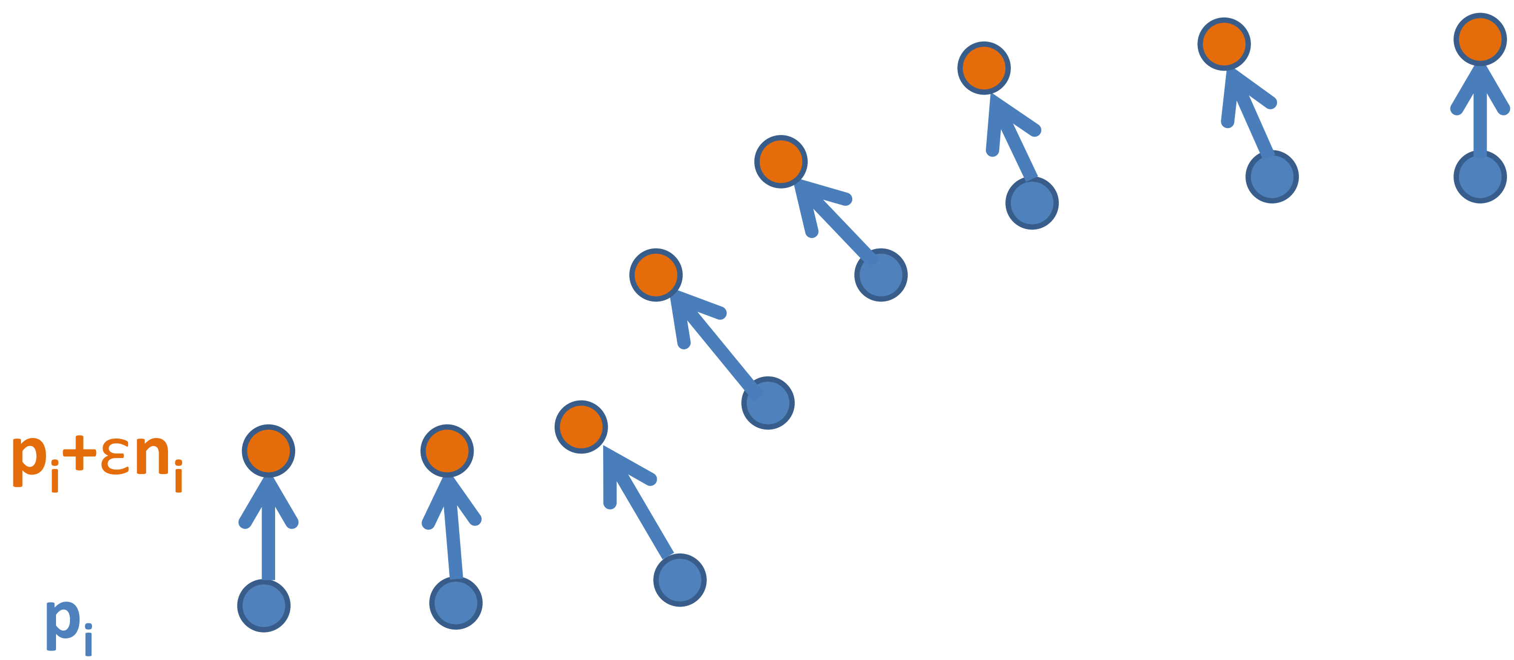
$$b(r) = r^k,\ k = 1, 3, 5, \dots$$
$$b(r) = r^k \ln(r),\ k = 2, 4, 6, \dots$$

- Thin plate spline (a special polyharmonic spline):

$$b(r) = r^2 \ln(r)$$

Centers $\mathbf{c}_k$

Centers are data points (**on-surface data points**) and offsets along their normals (**off-surface data points**)



"Reconstruction and Representation of 3D Objects with Radial Basis Functions", SIGGRAPH 2001

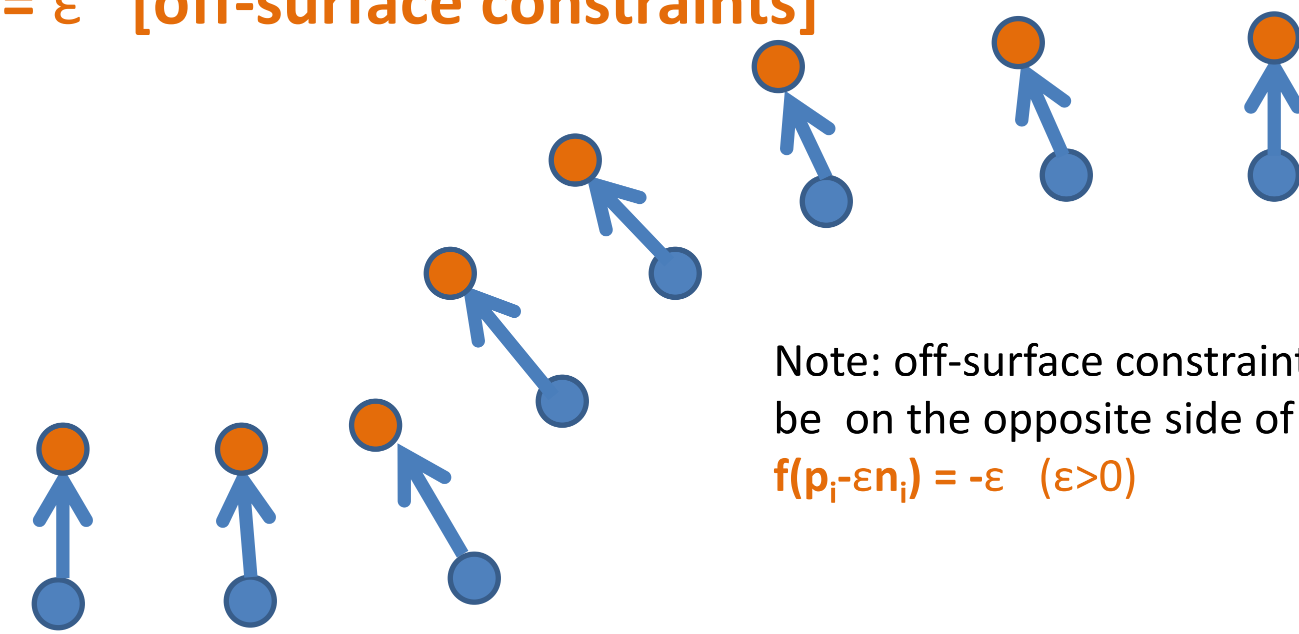
Weights $\mathbf{w_k}$

Note: function is linear on the unknown weights!

$$f(\mathbf{p}) = \sum_k w_{\mathbf{k}} b(\| \mathbf{p} - \mathbf{c_k} \|)$$

$f(p_i) = 0$ [on-surface constraints]

$f(p_i+\varepsilon n_i) = \varepsilon$ [off-surface constraints]



Note: off-surface constraints can also be on the opposite side of the surface: $f(p_i-\varepsilon n_i) = -\varepsilon$ $(\varepsilon>0)$

"Reconstruction and Representation of 3D Objects with Radial Basis Functions", SIGGRAPH 2001

Weights $\mathbf{w_k}$

$f(p_i) = 0$ [on-surface constraints]

$f(p_i+\varepsilon n_i) = \varepsilon$ [off-surface constraints]

2N equations where N is the number of points

$$\begin{bmatrix} b(\|\mathbf{p}_1-\mathbf{p}_1\|) & \cdots & b(\|\mathbf{p}_1-\mathbf{p}_N\|) & \cdots & b(\|\mathbf{p}_1-(\mathbf{p}_N+\varepsilon\mathbf{n}_N\|) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b(\|\mathbf{p}_N-\mathbf{p}_1\|) & \cdots & b(\|\mathbf{p}_N-\mathbf{p}_N\|) & \cdots & b(\|\mathbf{p}_N-(\mathbf{p}_N+\varepsilon\mathbf{n}_N\|) \\ b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-\mathbf{p}_1\|) & \cdots & b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-\mathbf{p}_N\|) & \cdots & b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-(\mathbf{p}_N+\varepsilon\mathbf{n}_N\|) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-\mathbf{p}_1\|) & \cdots & b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-\mathbf{p}_N\|) & \cdots & b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-(\mathbf{p}_N+\varepsilon\mathbf{n}_N)\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \cdots \\ w_N \\ w_{N+1} \\ \cdots \\ w_{2N} \end{bmatrix} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ \varepsilon \\ \cdots \\ \varepsilon \end{bmatrix}$$

"Reconstruction and Representation of 3D Objects with Radial Basis Functions", SIGGRAPH 2001

Weights $\mathbf{w}_k$

$f(p_i) = 0$ [on-surface constraints]

$f(p_i+\varepsilon n_i) = \varepsilon$ [off-surface constraints]

2N equations where N is the number of points

$$\begin{bmatrix} b(\|\mathbf{p}_1-\mathbf{p}_1\|) & \cdots & b(\|\mathbf{p}_1-\mathbf{p}_N\|) & \cdots & b(\|\mathbf{p}_1-(\mathbf{p}_N+\varepsilon\mathbf{n}_N\|) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b(\|\mathbf{p}_N-\mathbf{p}_1\|) & \cdots & b(\|\mathbf{p}_N-\mathbf{p}_N\|) & \cdots & b(\|\mathbf{p}_N-(\mathbf{p}_N+\varepsilon\mathbf{n}_N\|) \\ b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-\mathbf{p}_1\|) & \cdots & b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-\mathbf{p}_N\|) & \cdots & b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-(\mathbf{p}_N+\varepsilon\mathbf{n}_N\|) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-\mathbf{p}_1\|) & \cdots & b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-\mathbf{p}_N\|) & \cdots & b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-(\mathbf{p}_N+\varepsilon\mathbf{n}_N)\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \cdots \\ w_N \\ w_{N+1} \\ \cdots \\ w_{2N} \end{bmatrix} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ \varepsilon \\ \cdots \\ \varepsilon \end{bmatrix}$$

$$\mathbf{M}\cdot\mathbf{w}=\mathbf{d}$$

"Reconstruction and Representation of 3D Objects with Radial Basis Functions", SIGGRAPH 2001

Weights $\mathbf{w_k}$

$f(p_i) = 0$ [on-surface constraints]

$f(p_i + \varepsilon n_i) = \varepsilon$ [off-surface constraints]

2N equations where N is the number of points

$$
\begin{bmatrix}
b(\|\mathbf{p}_1 - \mathbf{p}_1\|) & \dots & b(\|\mathbf{p}_1 - \mathbf{p}_N\|) & \dots & b(\|\mathbf{p}_1 - (\mathbf{p}_N + \varepsilon\mathbf{n}_N)\|) \\
\dots & \dots & \dots & \dots & \dots \\
b(\|\mathbf{p}_N - \mathbf{p}_1\|) & \dots & b(\|\mathbf{p}_N - \mathbf{p}_N\|) & \dots & b(\|\mathbf{p}_N - (\mathbf{p}_N + \varepsilon\mathbf{n}_N)\|) \\
b(\|(\mathbf{p}_1 + \varepsilon\mathbf{n}_1) - \mathbf{p}_1\|) & \dots & b(\|(\mathbf{p}_1 + \varepsilon\mathbf{n}_1) - \mathbf{p}_N\|) & \dots & b(\|(\mathbf{p}_1 + \varepsilon\mathbf{n}_1) - (\mathbf{p}_N + \varepsilon\mathbf{n}_N)\|) \\
\dots & \dots & \dots & \dots & \dots \\
b(\|(\mathbf{p}_N + \varepsilon\mathbf{n}_N) - \mathbf{p}_1\|) & \dots & b(\|(\mathbf{p}_N + \varepsilon\mathbf{n}_N) - \mathbf{p}_N\|) & \dots & b(\|(\mathbf{p}_N + \varepsilon\mathbf{n}_N) - (\mathbf{p}_N + \varepsilon\mathbf{n}_N)\|)
\end{bmatrix}
\begin{bmatrix} w_1 \\ \dots \\ w_N \\ w_{N+1} \\ \dots \\ w_{2N} \end{bmatrix}
=
\begin{bmatrix} 0 \\ \dots \\ 0 \\ \varepsilon \\ \dots \\ \varepsilon \end{bmatrix}
$$

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{d}$$

sum of basis functions for point p_1, p_1 is on surface, thus $f(p_1)=0$

“Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001

Weights $\mathbf{w}_k$

$f(p_i) = 0$ [on-surface constraints]

$f(p_i+\varepsilon n_i) = \varepsilon$ [off-surface constraints]

2N equations where N is the number of points

$$
\begin{bmatrix}
b(\|\mathbf{p}_1-\mathbf{p}_1\|) & \cdots & b(\|\mathbf{p}_1-\mathbf{p}_N\|) & \cdots & b(\|\mathbf{p}_1-(\mathbf{p}_N+\varepsilon\mathbf{n}_N)\|) \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
b(\|\mathbf{p}_N-\mathbf{p}_1\|) & \cdots & b(\|\mathbf{p}_N-\mathbf{p}_N\|) & \cdots & b(\|\mathbf{p}_N-(\mathbf{p}_N+\varepsilon\mathbf{n}_N)\|) \\
b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-\mathbf{p}_1\|) & \cdots & b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-\mathbf{p}_N\|) & \cdots & b(\|(\mathbf{p}_1+\varepsilon\mathbf{n}_1)-(\mathbf{p}_N+\varepsilon\mathbf{n}_N)\|) \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-\mathbf{p}_1\|) & \cdots & b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-\mathbf{p}_N\|) & \cdots & b(\|(\mathbf{p}_N+\varepsilon\mathbf{n}_N)-(\mathbf{p}_N+\varepsilon\mathbf{n}_N)\|)
\end{bmatrix}
\begin{bmatrix} w_1 \\ \cdots \\ w_N \\ w_{N+1} \\ \cdots \\ w_{2N} \end{bmatrix}
=
\begin{bmatrix} 0 \\ \cdots \\ 0 \\ \varepsilon \\ \cdots \\ \varepsilon \end{bmatrix}
$$

$$\mathbf{M}\cdot\mathbf{w}=\mathbf{d}$$

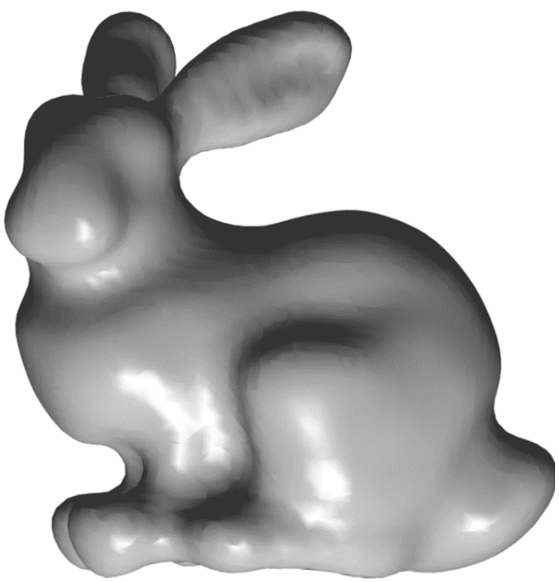
sum of basis functions for point $p_N+\varepsilon n_N$, $p_N+\varepsilon n_N$ is off surface, therefore $f(p_N+\varepsilon n_N)=\varepsilon$

“Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001

RBF Interpolation

Which do you prefer? MLS or RBF?

MLS



Tri-harmonic basis functions r^3

Thin plate spline basis functions $r^2 \log r$

- ➢Globally supported
- ➢Provably smooth, C2 smoothness
- ➢Works relatively well for irregular sampling

Extensions

- **Greedy RBFs:** start with a few basis functions (few centers c_i), then add more RBFs in the areas of large residual error

- **Augmented RBFs:** add a polynomial term q(x,y,z) in our implicit function to better interpolate planar (or low-order polynomial) patches:

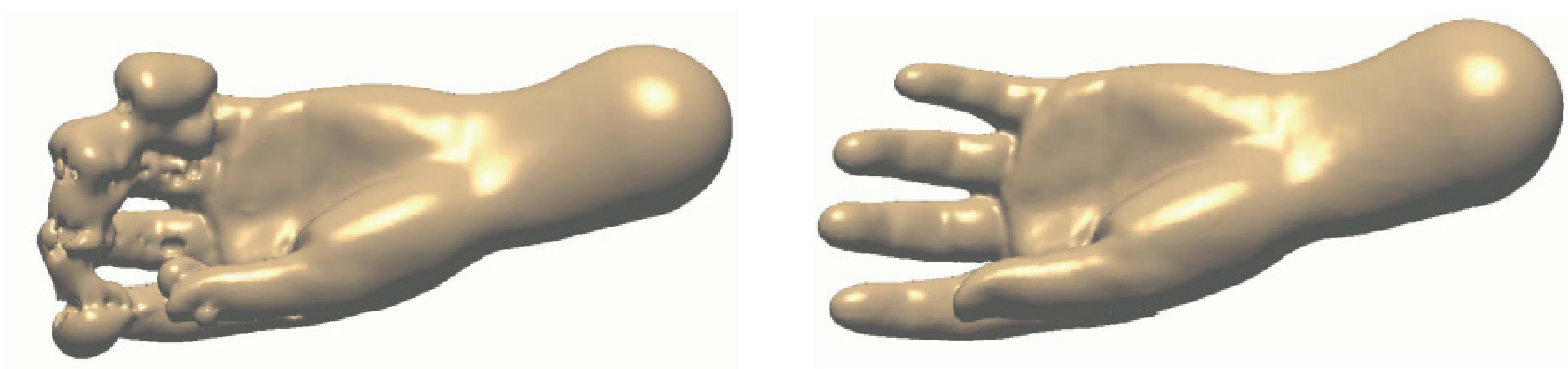
$$f(\mathbf{p}) = q(\mathbf{p}) + \sum_k w_\mathbf{k} b(\| \mathbf{p} - \mathbf{c_k} \|)$$

"Reconstruction and Representation of 3D Objects with Radial Basis Functions", SIGGRAPH 2001

How big should ε be?

Each offset point should be constructed so that **its closest surface point is the surface point that generated it**!

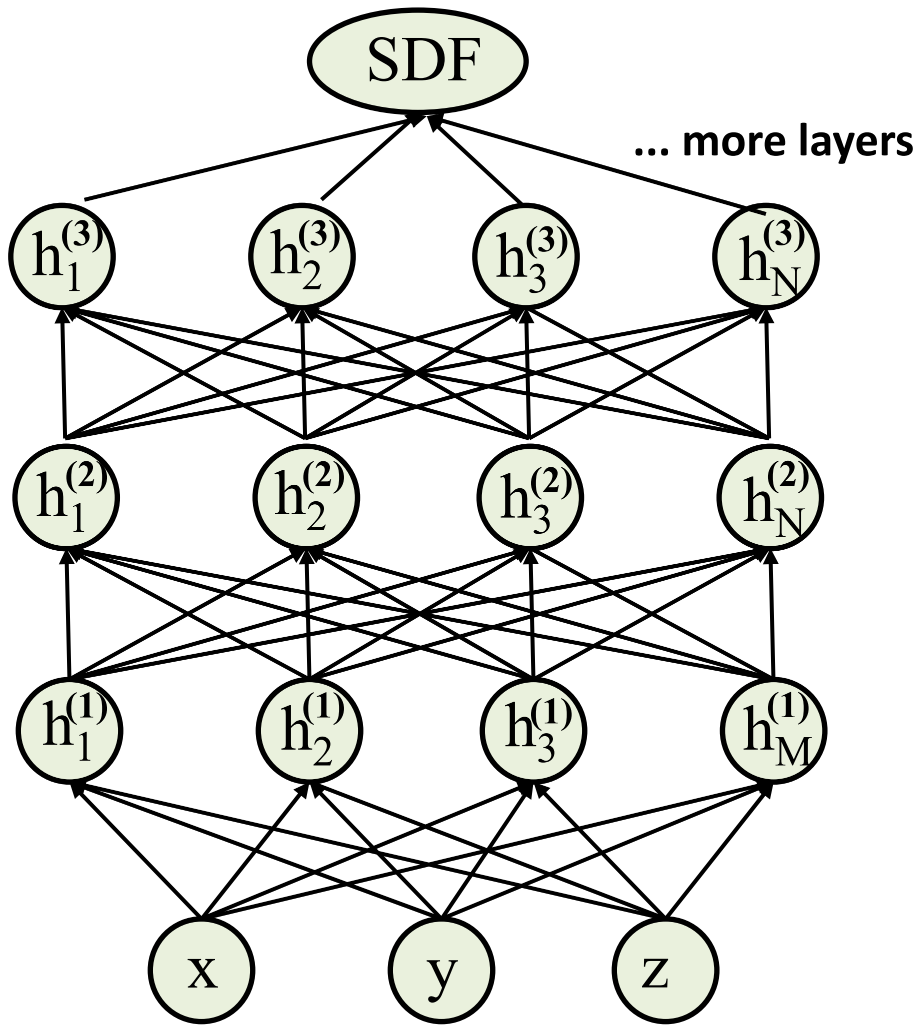
Otherwise e.g. an offset point that originated from one finger might intersect or enter inside another finger. If this happens, we get the reconstruction on the left!



“Reconstruction and Representation of 3D Objects with Radial Basis Functions”, SIGGRAPH 2001

More recent developments...

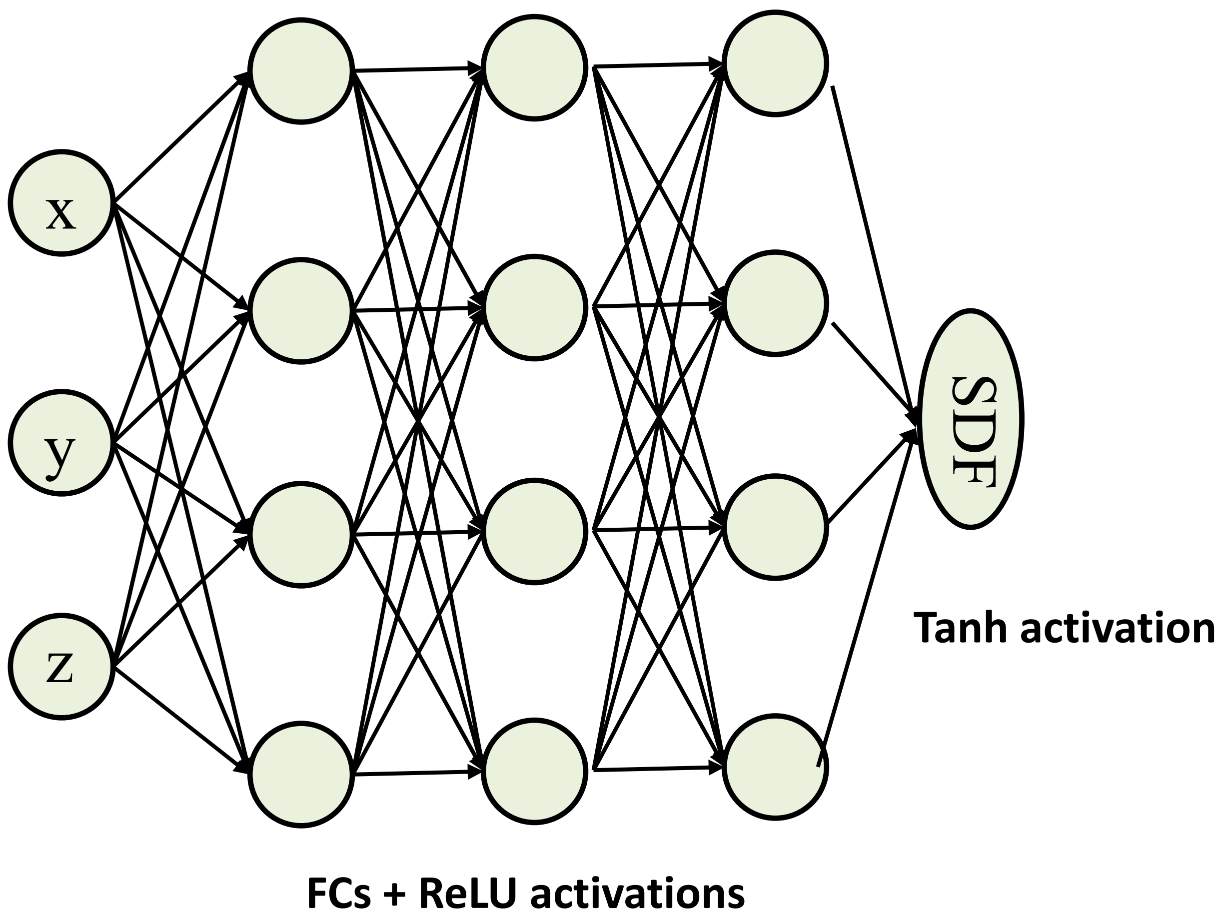
... Deep SDF



DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 19

The most recent development...

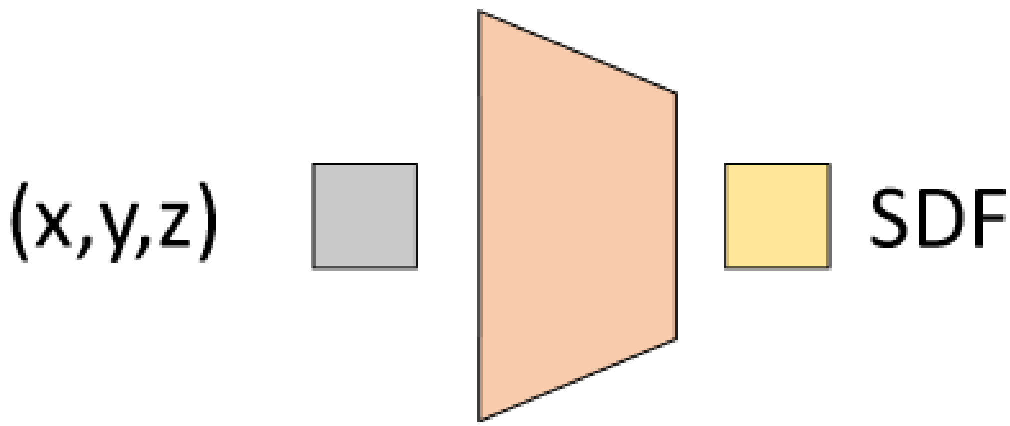
... Deep SDF



DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 19

The most recent development...

... Deep SDF



Single Shape DeepSDF

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 19

Loss function

Given a sample point p and its ground-truth SDF value s:

$$L(\ f(\mathbf{p}),\ s_p\ ) = \left|\ f(\mathbf{p}) - \ s_p\ \right|$$

i.e., L1 loss, summed over all sample points
(sample from a Gaussian centered at each input point)

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 19

Loss function

Given a sample point p and its ground-truth SDF value s:

$$L(\ f(\mathbf{p}),\ s_p\ ) = \left|\ f(\mathbf{p}) - \ s_p\ \right|$$

i.e., L1 loss, summed over all sample points
(sample from a Gaussian centered at each input point)

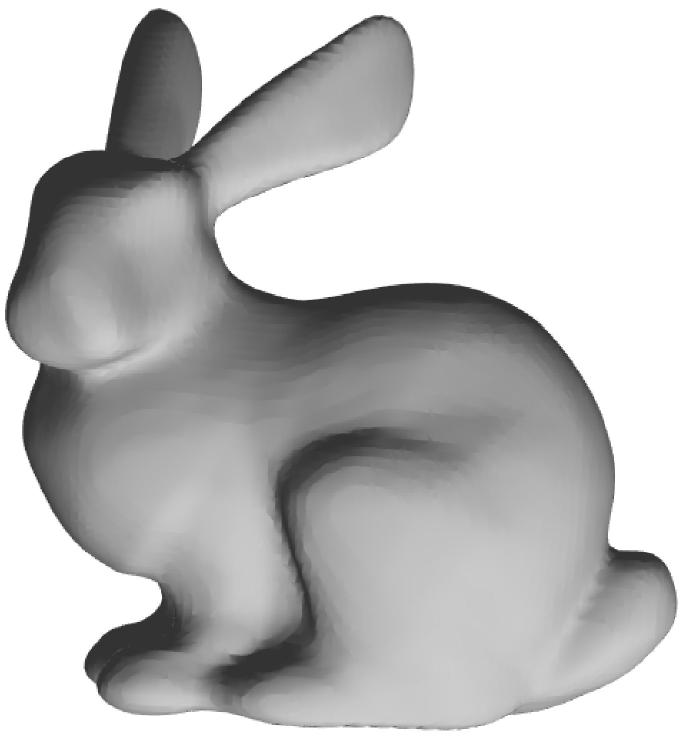
Even better, use **clamped L1 loss** (focus on predicting correct SDFs near the surface)

$$L(\ f(\mathbf{p}),\ s\ ) = \left|\ clamp(\ f(\mathbf{p}),\ \delta\ ) - \ clamp(s_p,\ \delta\ )\right|$$

$$clamp(\ x,\ \delta\ ) = \min(\ \delta,\ \max(-\delta, x))$$

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 19

DeepSDF vs RBF



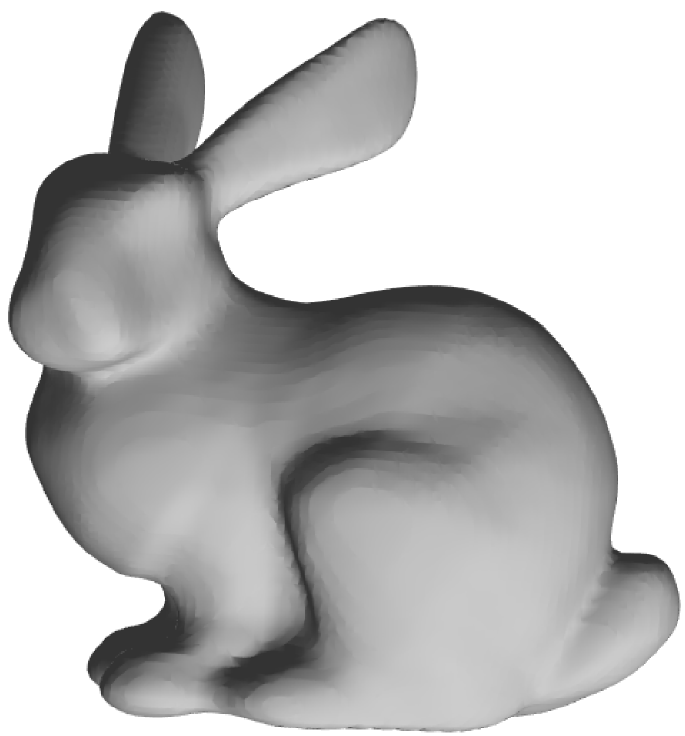
Single-Shape DeepSDF

RBF
Tri-harmonic basis functions r^3

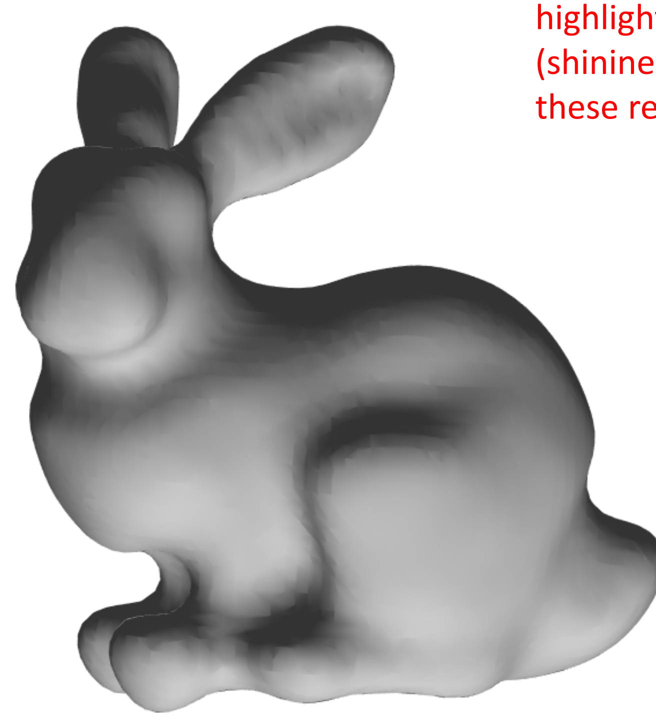
Note: I disabled specular highlights (shininess for these renderings)

DeepSDF vs MLS

Note: I disabled specular highlights (shininess for these renderings)



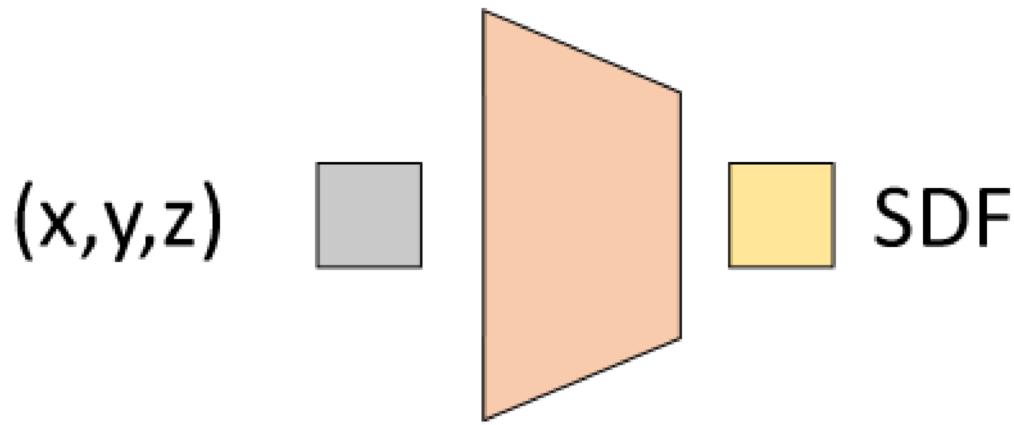
Single-Shape DeepSDF



MLS

Loss function

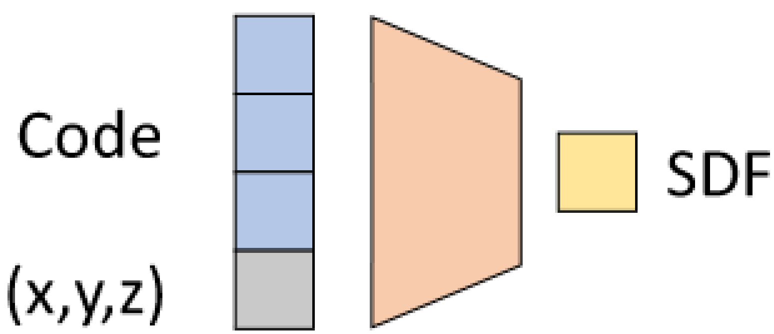
Here we trained on samples from point cloud, then evaluated the RBF or DeepSDF on grid points to reconstruct the same input shape



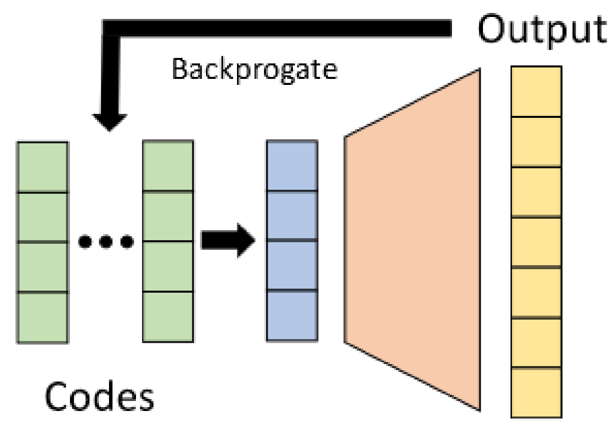
Single Shape DeepSDF

Loss function

The main power of DeepSDF (and similar techniques) is in other generative tasks e.g., encode a RGB image to a code, then decode it to SDF of **a shape not seen during training**, or train it on many shapes to **generate new shapes** etc (more about generative models next time)



(b) Coded Shape DeepSDF



(b) Auto-decoder

So we have a predicted SDF at each **grid point**... what to do next?

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

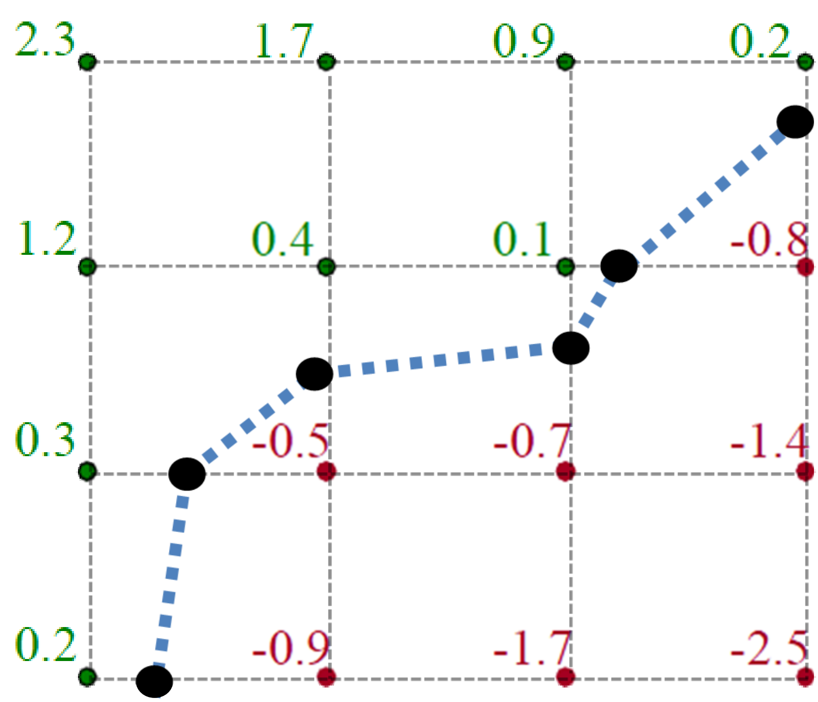
Isosurface extraction

Let’s start with 2D case

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

Isosurface extraction

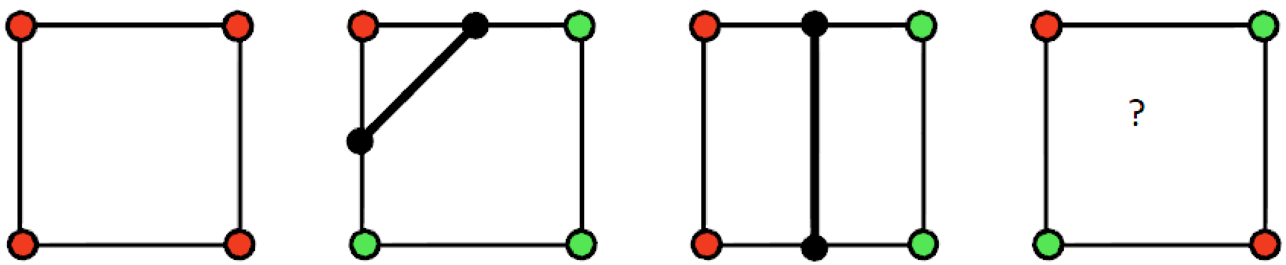
Let's start with 2D case



Marching cubes – 2D

$2^4 = 16$ possible combinations of positive/negative (green/red) values on the vertices of the square

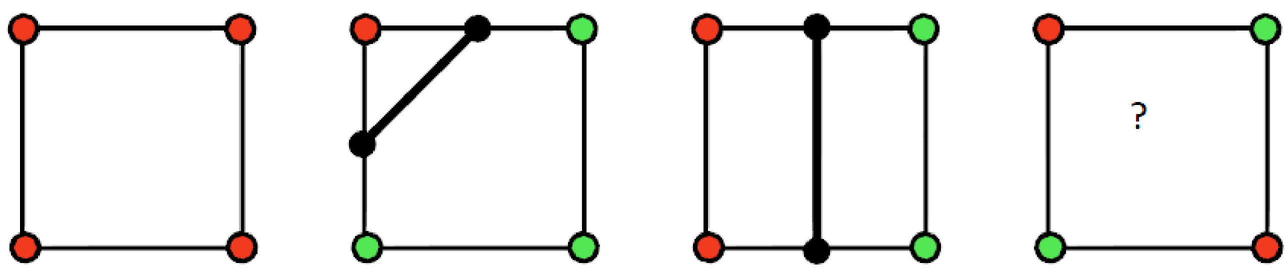
Due to symmetry, 4 unique configurations:



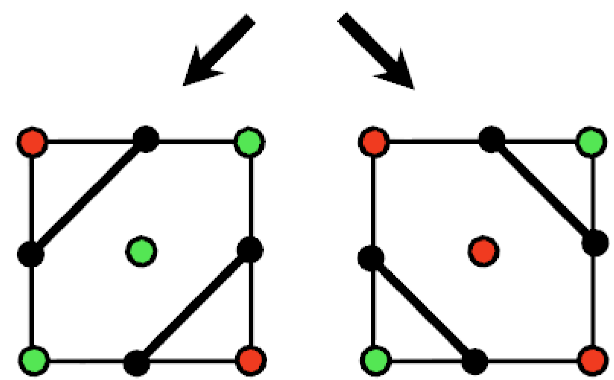
Marching cubes – 2D

$2^4 = 16$ possible combinations of positive/negative (green/red) values on the vertices of the square

Due to symmetry, 4 unique configurations:



Linear interpolation to find edges and normals,
but we might have ambiguity:
Check value at the center point.



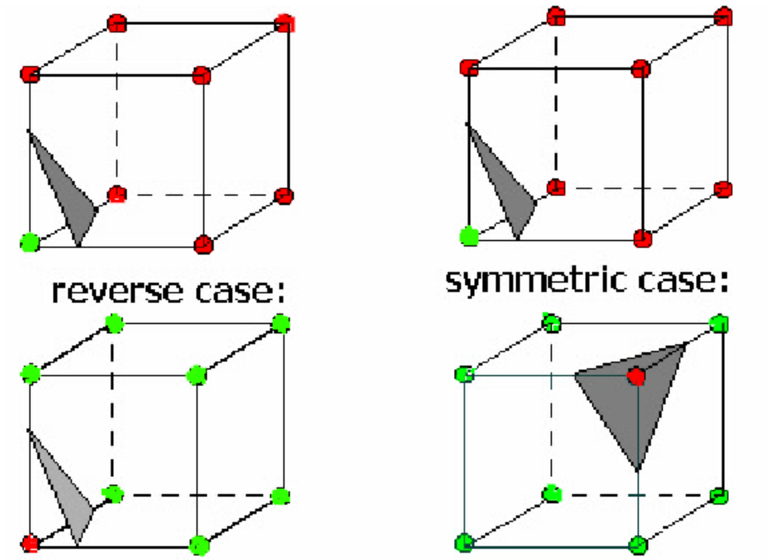
Marching cubes - 3D

Classify grid vertices as inside/outside

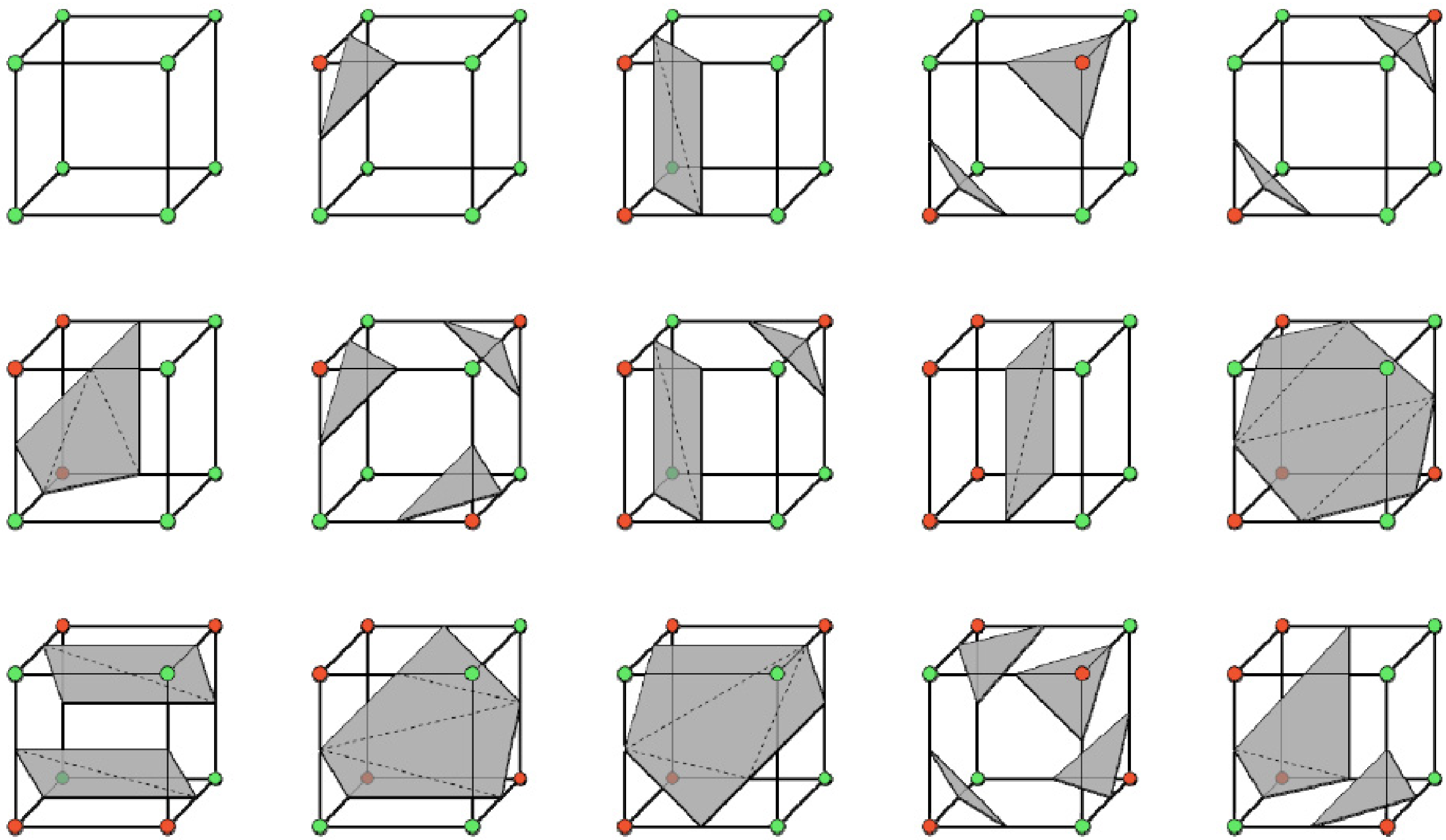
2^8 = 256 cases in 3D

Look-up table for each case - each entry defines the edge configuration to be created

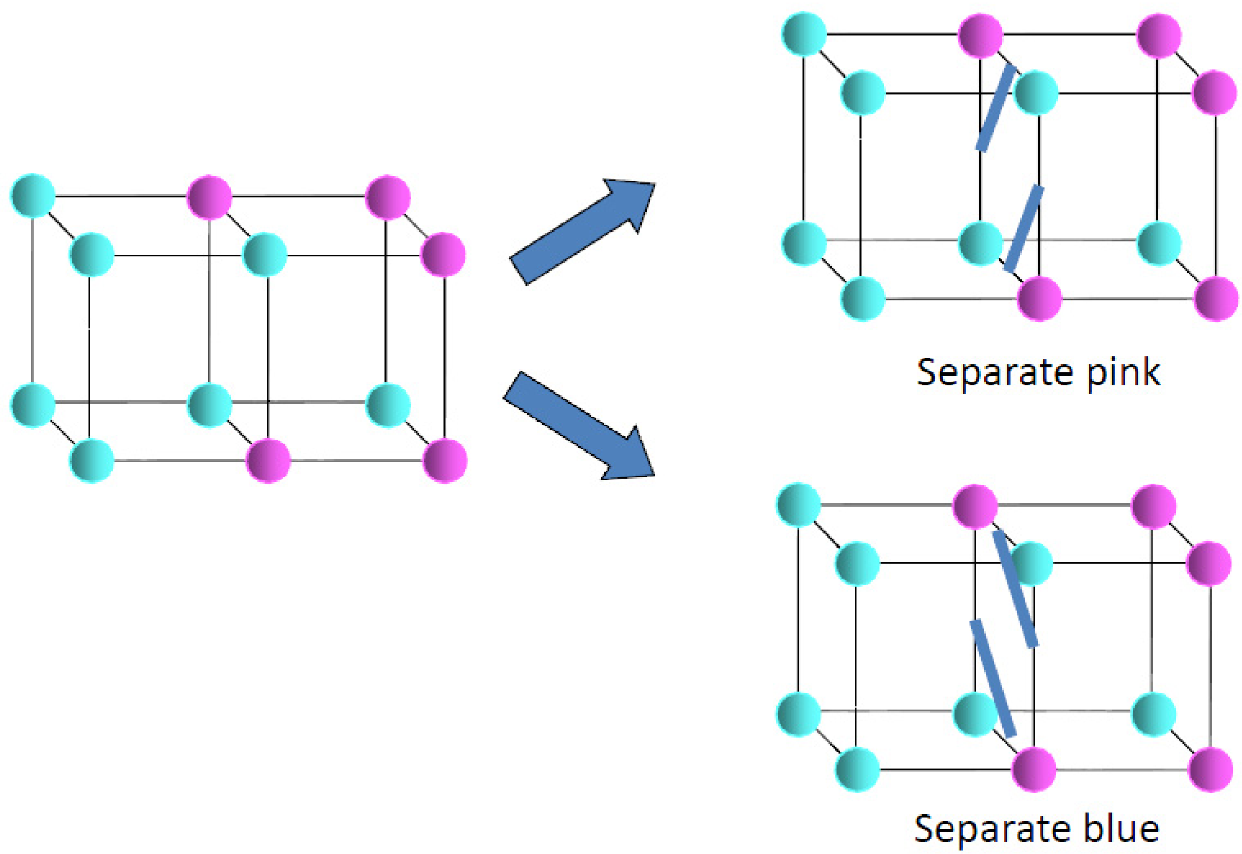
Due to symmetries (see below), 15 unique configurations



Marching cubes - 3D

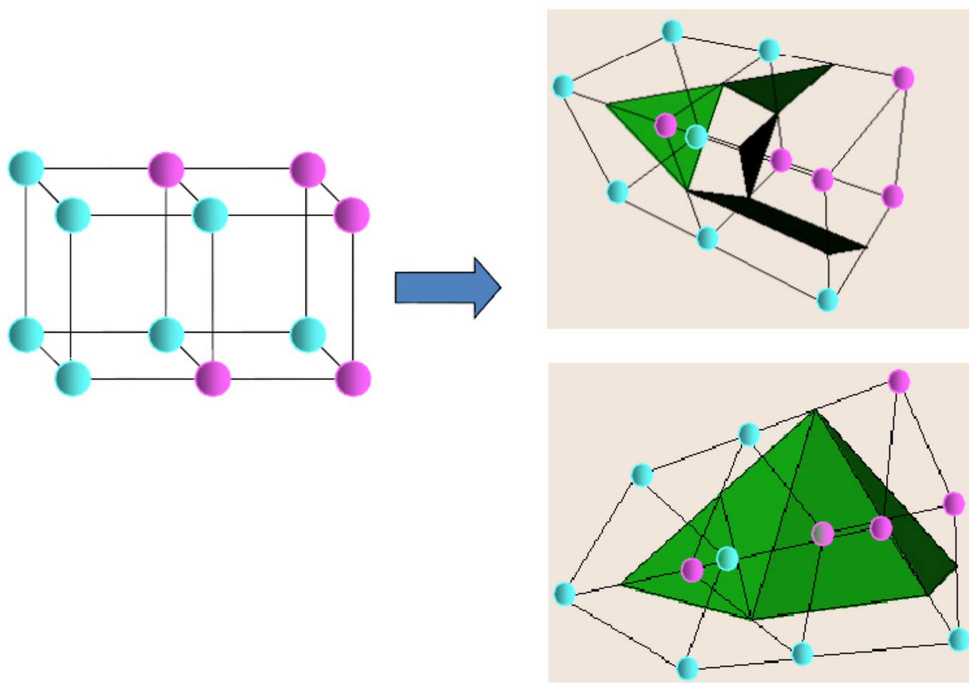


Ambiguity also in 3D



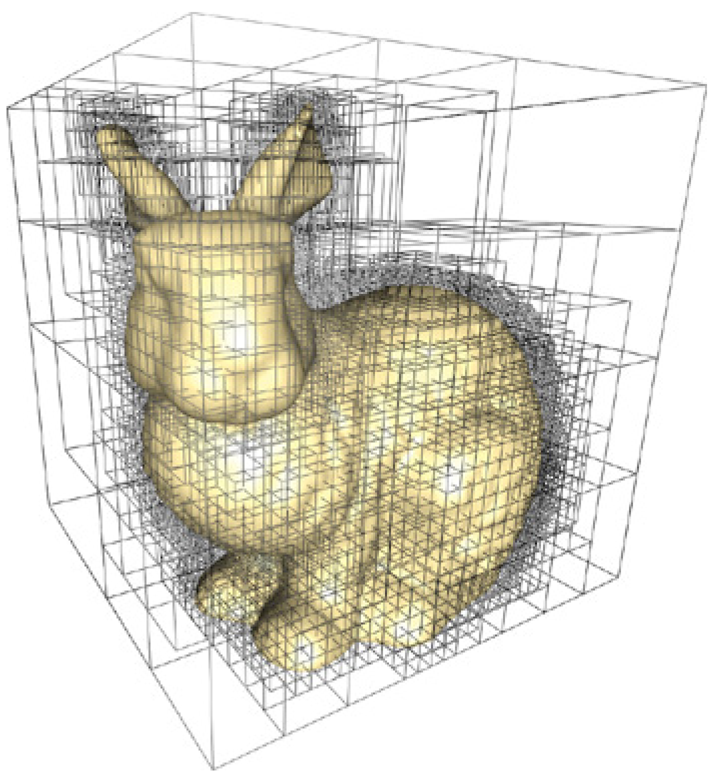
Resolve ambiguity

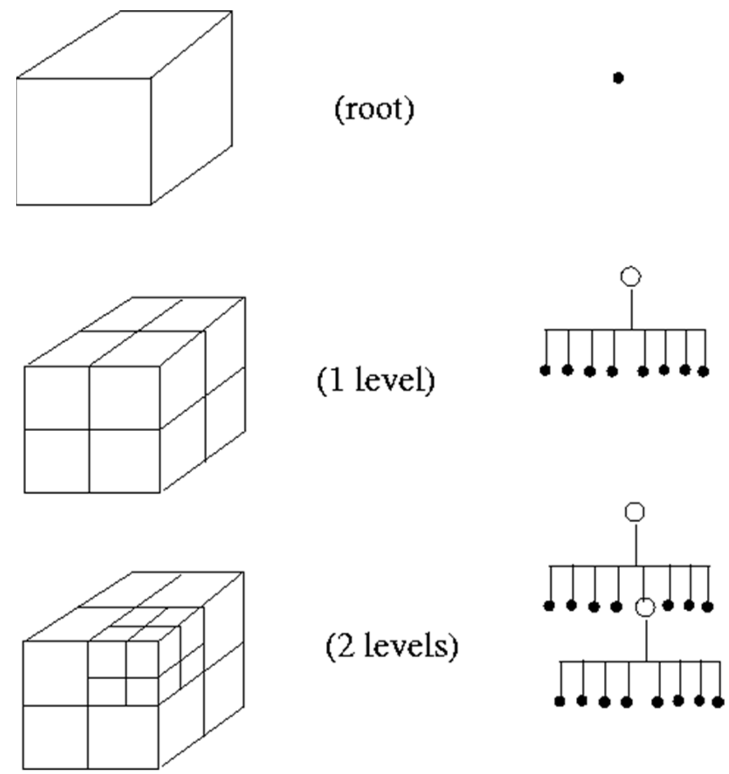
Look at neighboring cube, and select configuration that avoids holes



Extensions

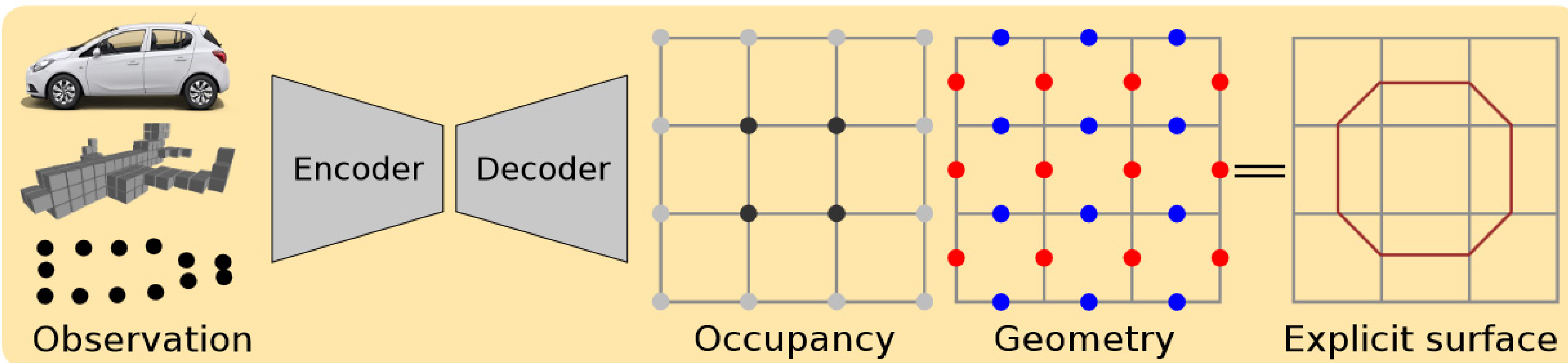
Use adaptive grid (octree data structure)





Deep Marching Cubes

Differentiable marching cube network layer by predicting occupancy at each grid point and vertex locations along each edge



Deep Marching Cubes: Learning Explicit Surface Representations, CVPR 18

Assignment 4 is out!

Implement four implicit surface reconstruction techniques:

(a) SDF based on nearest point's tangent plane
(b) Moving Least Squares SDF
(c) RBF interpolation network SDF
(d) DeepSDF (single-shape variant)

... start early!

Marking Scheme (**574**)

- **5% Assignment 1** (warm-up): Shape Classification
- **10% Assignment 2:** Multi-View Convolutional Networks
- **5% Mini-Assignment:** Choose a paper for presentation
- **22% Assignment 3:** Point-based Networks
- **23% Assignment 4:** Implicit Surface Reconstruction
- **15% Reaction Reports**
- **20% Paper presentation**

Marking Scheme (**674**)

- **70% same assignments & reaction reports ...**
  Divide their total 80 points with 1.142 = 70%
  (note: mini-assignment is the project proposal)
- **30% Project + Project Presentation**