MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

METAVERSE
CENTER

CV 804: 3D RENDERING & GEOMETRY PROCESSING

WEEK 12 / EXERCISE 6: Implicit Surface Reconstruction

HAO LI

04/15/24

Implicit Surface Reconstruction

- Estimate point cloud normals (precomputed)
- Estimate signed distance function (SDF)
- Evaluate the distances on a uniform grid
- Extract mesh via marching cubes

Exercise 6

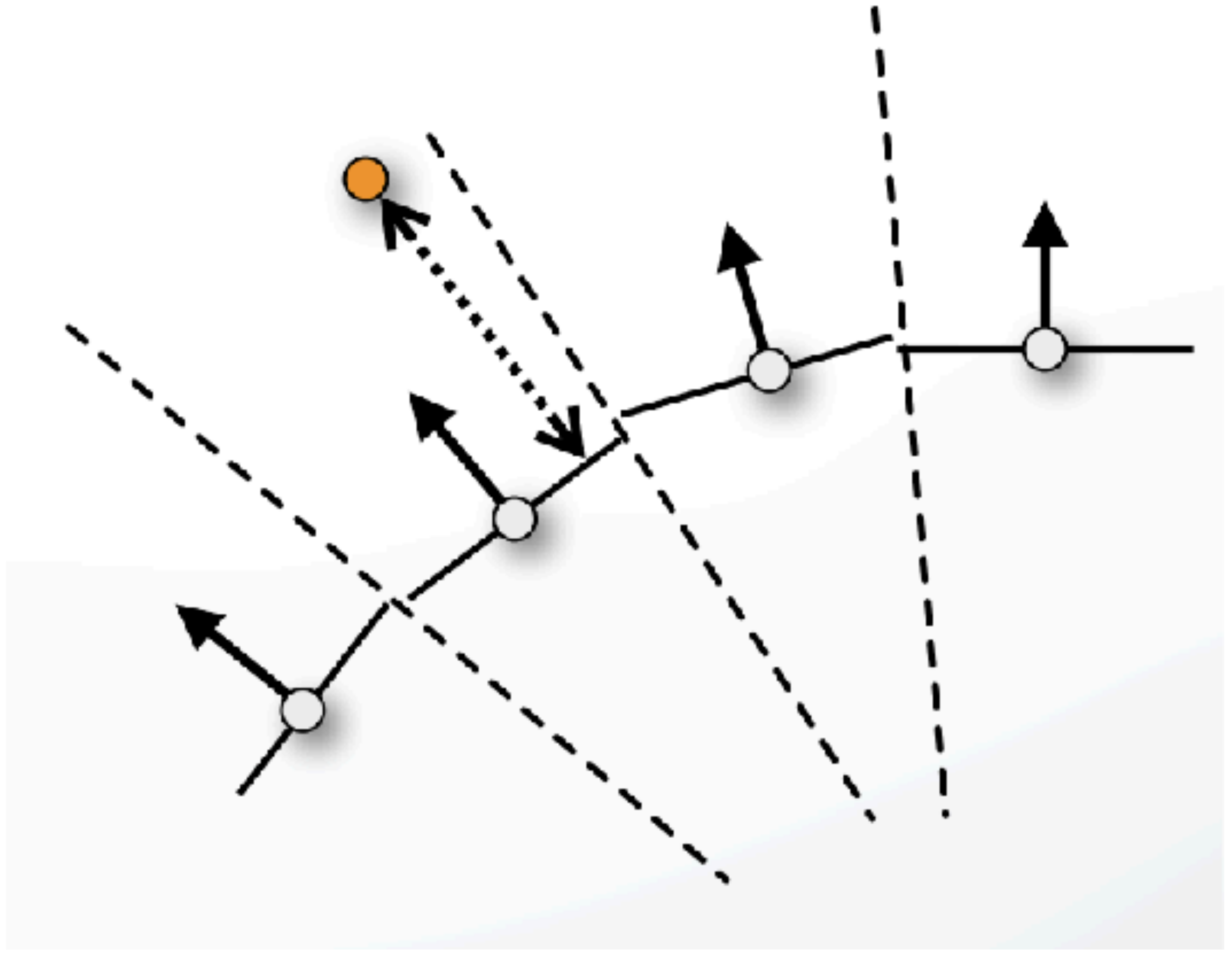
- You will be given 3 sets of point clouds
- Implement two popular methods to estimate signed distance function for implicit surface reconstruction:
  - Hoppe distance from tangent planes [Hoppe 92]
  - Triharmonic Radial Basis Functions (RBFs)

Libraries

- OpenGL, OpenMesh
- Generic Matrix Methods (gmm)
  - used for solving linear equation
- IsoEx: marching cubes

Hoppe Distance

- Distance from tangent plane
  - point & normal forms a local tangent plane
  - use distance from closest point's tangent plane

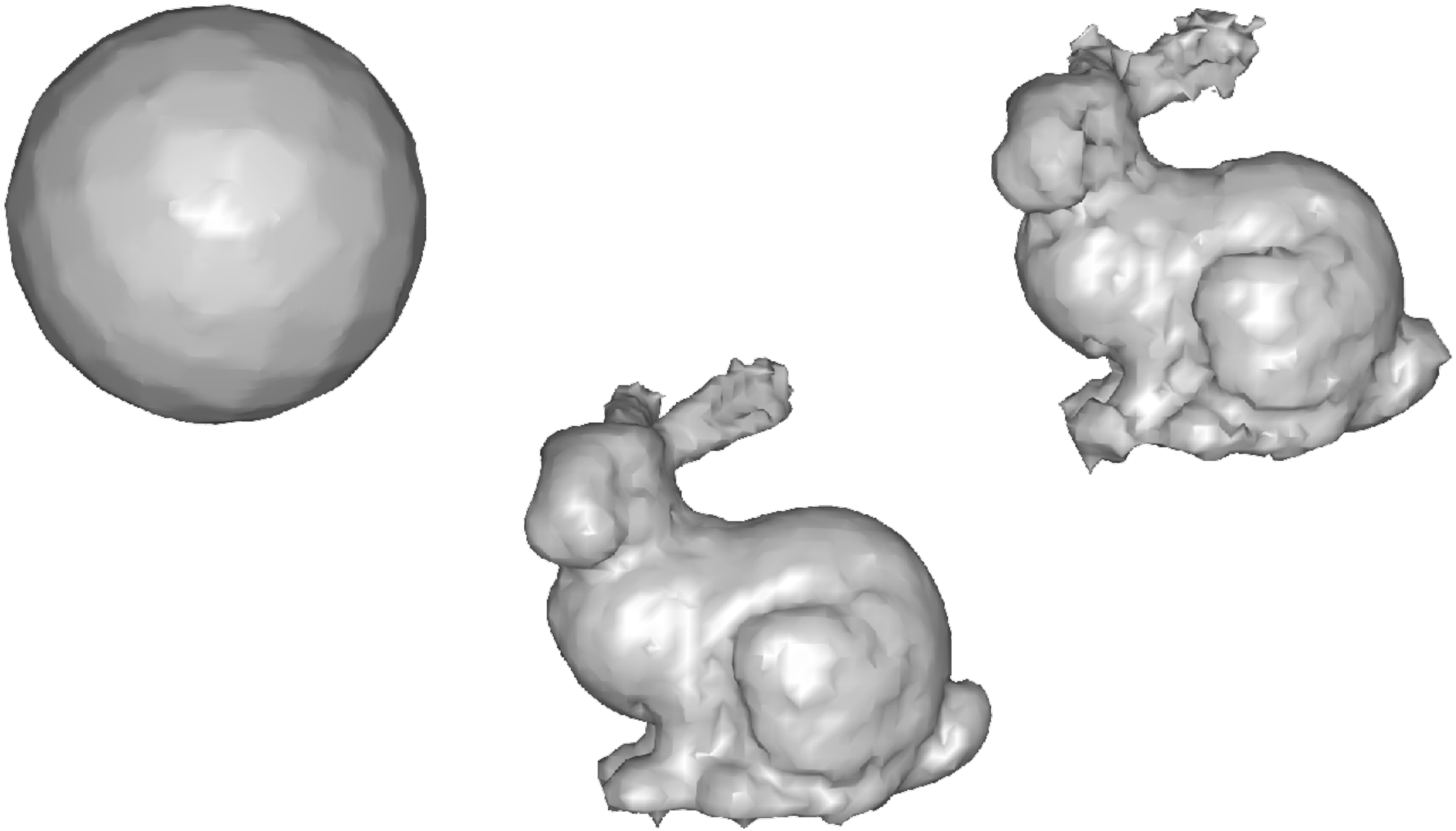


Hoppe Distance

- Distance from tangent plane
  - point & normal forms a local tangent plane
  - use distance from closest point’s tangent plane
- ImplicitHoppe::operator() in ImplicitHoppe.hh
- Surface reconstruction from unorganized points.

  [Hoppe et al. ’92]

Results



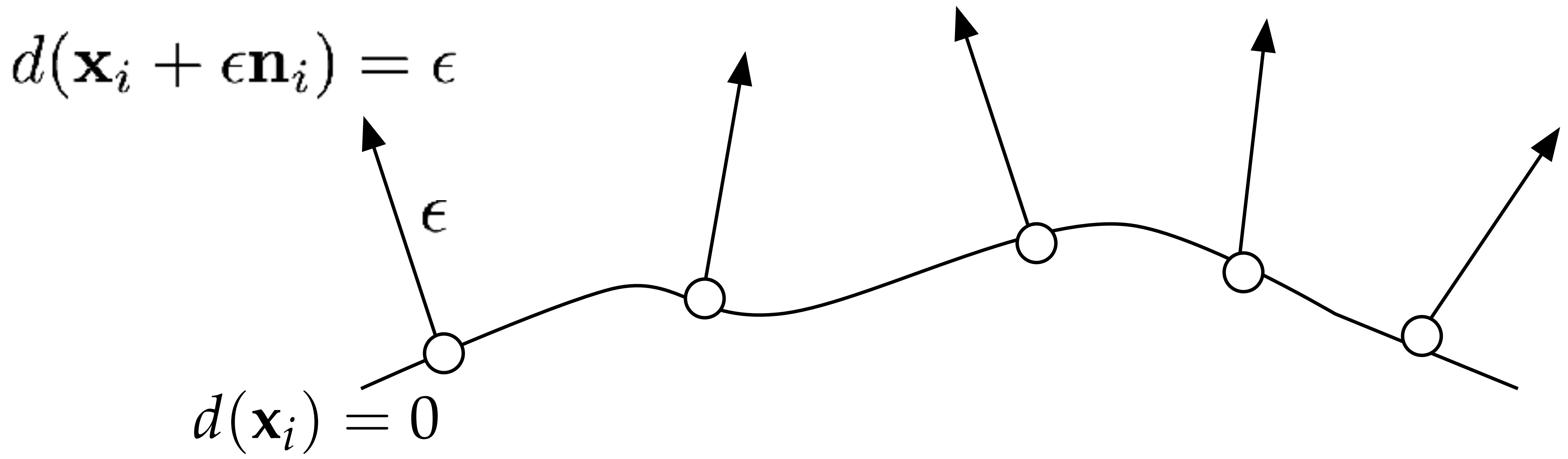
RBF

- Radial Basis Functions (RBFs)
  - Sum of shifted, weighted kernel functions

$$d(\mathbf{x}) = \sum_i w_i \phi(\|\mathbf{x} - \mathbf{c}_i\|)$$

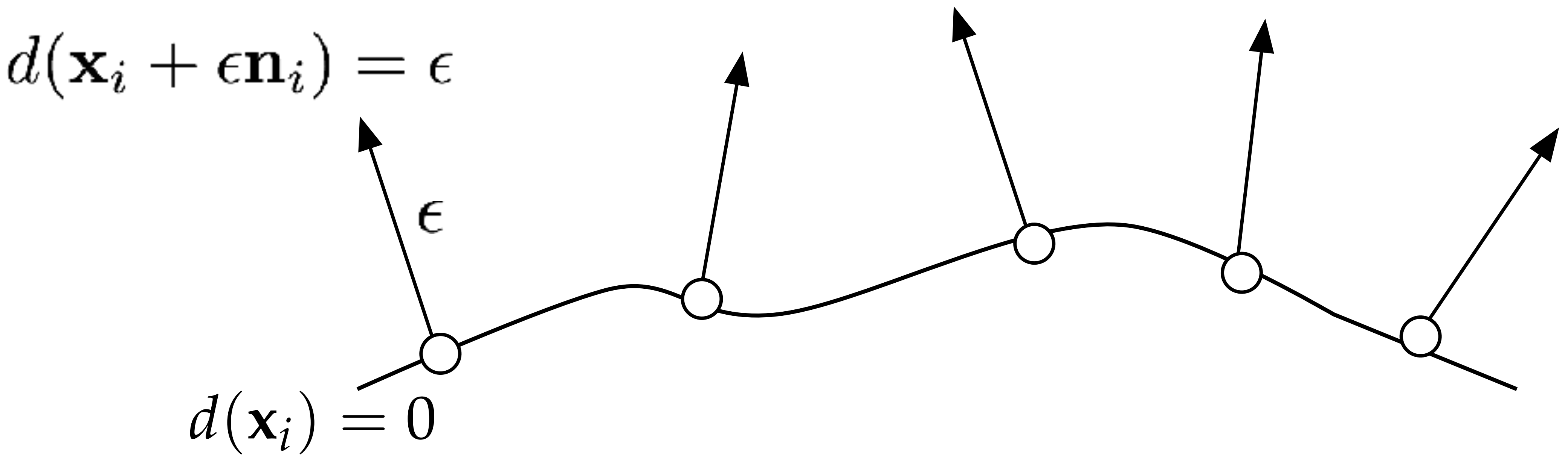
- Triharmonic RBFs: $\phi(x) = x^3$
- Solve for the weights using on- and off-surface constraints and gmm library
- ImplicitRBF::ImplicitRBF() in ImplicitRBF.cc

On- and Off-Surface Constraints



$$\begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \dots & \phi(\|\mathbf{x}_1 - (\mathbf{x}_n + \epsilon \mathbf{n}_n)\|) \\ \vdots & \ddots & \vdots \\ \phi(\|(\mathbf{x}_n + \epsilon \mathbf{n}_n) - \mathbf{x}_1\|) & \dots & \phi(\|(\mathbf{x}_n + \epsilon \mathbf{n}_n) - (\mathbf{x}_n + \epsilon \mathbf{n}_n)\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{2n} \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_{2n} \end{bmatrix}$$

On- and Off-Surface Constraints



How to choose the epsilon?

- Find the bounding box of object:

  min_p = [min_x, min_y, min_z]

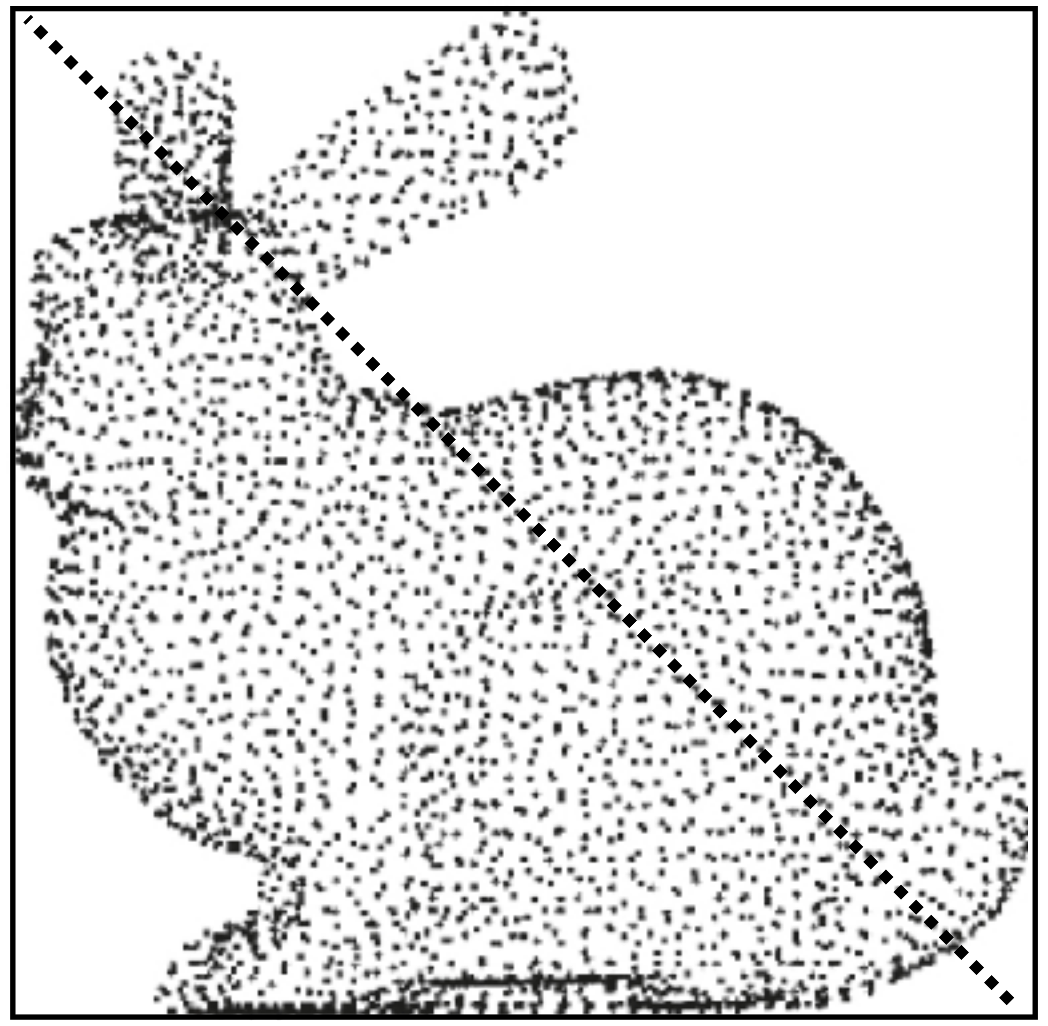
  max_p = [max_x, max_y, max_z]

- Calculate the distance:

  d_box = || max_p - min_p ||

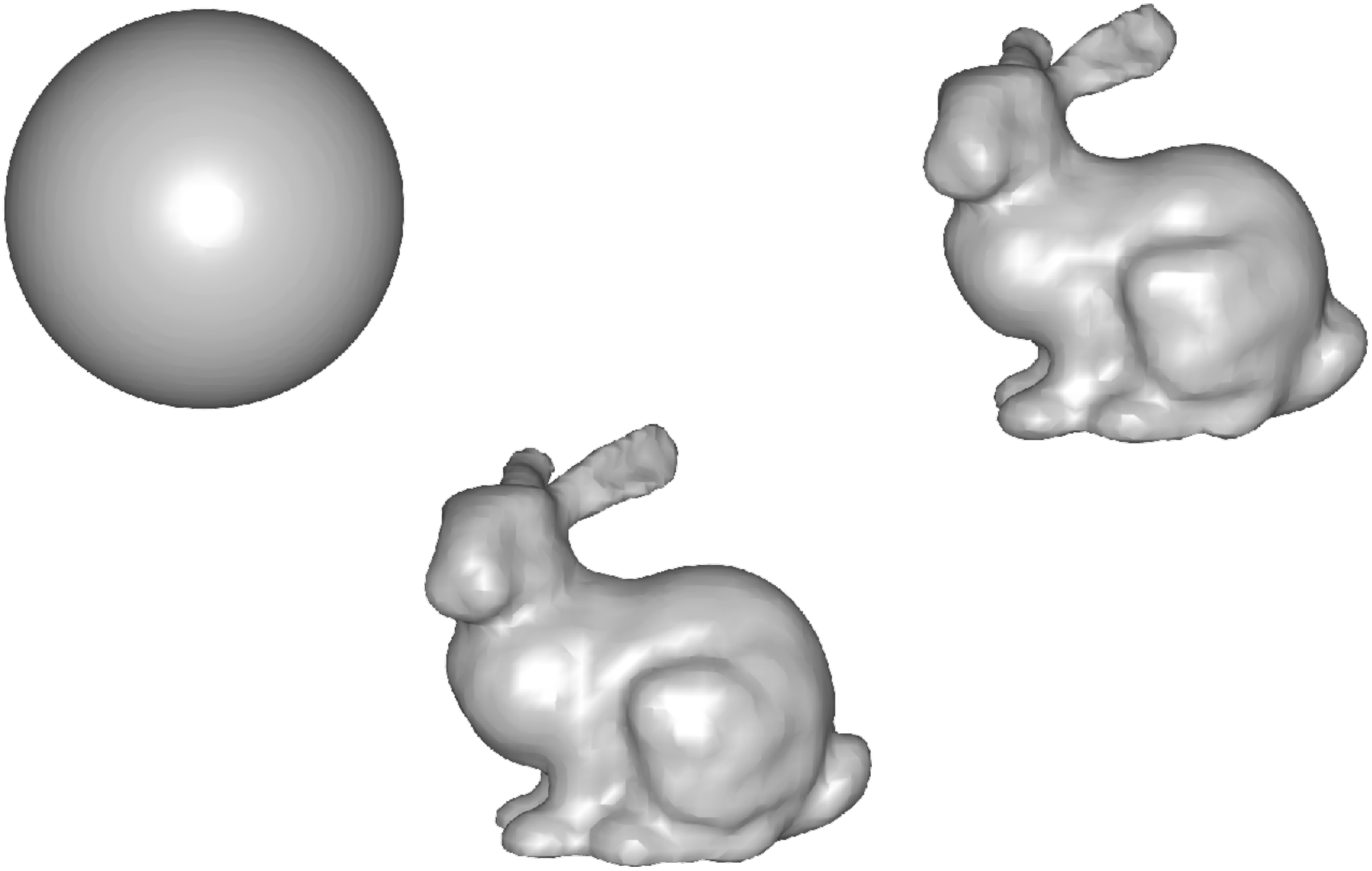
- Calculate epsilon:

  eps = scale * d_box



- The value of scale should be small, you can try different values of scale yourself to see its effect.
- Tip: scale=0.01 should work well

Results



Submission

- Deadline: **Tuesday 23rd April, 11:59 pm**
- Follow submission instructions on Ex.6 webpage:
  - Upload a .zip compressed file named “Exercise6-YourName.zip” to Moodle
  - Include your code with comments
  - Include a readme file:
    - Describe how you solved each problem
    - Describe problems you encountered
  - Include JPEG frames or a video

Contact

INSTRUCTOR
- Hao Li, hao.li@mbzuai.ac.ae
- Office: Building 1B, 1st floor (please schedule first)

TEACHING ASSISTANTS
- Phong Tran, the.tran@mbzuai.ac.ae
- Long-Nhat Ho, long.ho@mbzuai.ac.ae

OFFICE HOURS
* Office Hours: TBD, will be posted soon
* Emails (include “CV804” in title)

QUESTIONS?

www.hao-li.com

THANKS!

MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

hao.li@mbzuai.ac.ae