1-Lipschitz Neural Distance Fields

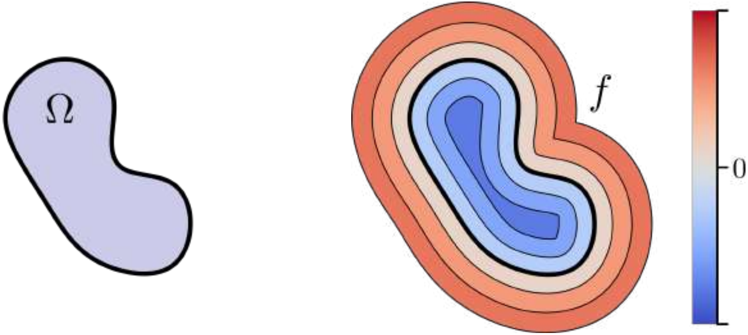
Guillaume Coiffier[1], Louis Béthune[2]

1 Université Catholique de Louvain
2 Apple

Implicit Representation of Geometry

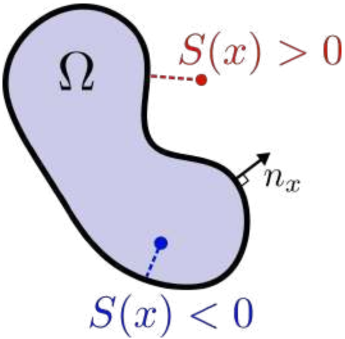
Represent a compact object $\Omega \subset \mathbb{R}^n$ as a level set of a continuous function:

$$\Omega = \{x \in \mathbb{R}^n \mid f(x) \leqslant 0\}$$



Signed Distance Function

$$S(x) = \begin{cases} -d(x, \partial\Omega) & \text{if } x \in \Omega \\ d(x, \partial\Omega) & \text{otherwise} \end{cases}$$

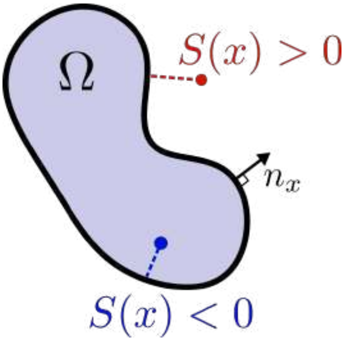


Signed Distance Function

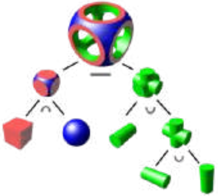
$$S(x) = \begin{cases} -d(x, \partial\Omega) & \text{if } x \in \Omega \\ d(x, \partial\Omega) & \text{otherwise} \end{cases}$$

Eikonal equation

$$\begin{cases} ||\nabla S(x)|| & = 1, \quad \forall x \in \mathbb{R}^n \\ S(x) & = 0, \quad \forall x \in \partial\Omega \\ \nabla S(x) & = n_x, \quad \forall x \in \partial\Omega \end{cases}$$

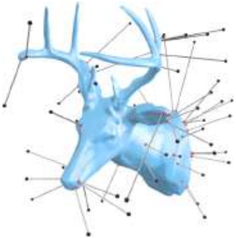


Applications



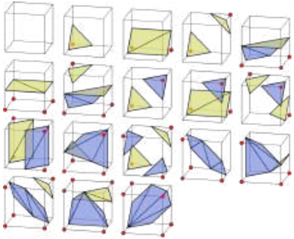
Constructive Solid Geometry

[Ricci (1973)]



Closest Point Query

[Sharp and Jacobson (2022)]

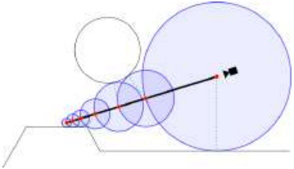


Marching Cubes

[Lorensen and Cline (1987)]

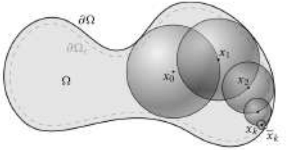
Rendering

Snail shader by Inigo Quilez



Empty Sphere Query

[Hart (1995)]



Monte-Carlo Simulation

[Sawhney and Crane (2020)]

Signed Distance Function

Simple shapes:
Direct approach

```
float sdHexagram( in vec2 p, in float r )
{
    const vec4 k = vec4(-0.5,0.8660254038,0.5773502692,1.7320508076);
    p = abs(p);
    p -= 2.0*min(dot(k.xy,p),0.0)*k.xy;
    p -= 2.0*min(dot(k.yx,p),0.0)*k.yx;
    p -= vec2(clamp(p.x,r*k.z,r*k.w),r);
    return length(p)*sign(p.y);
}
```

https://iquilezles.org/articles/distfunctions2d/

Complex shapes:
Neural Distance Fields

[Gropp et al. (2020)]

Approximate a solution of the Eikonal equation

Neural Distance Fields

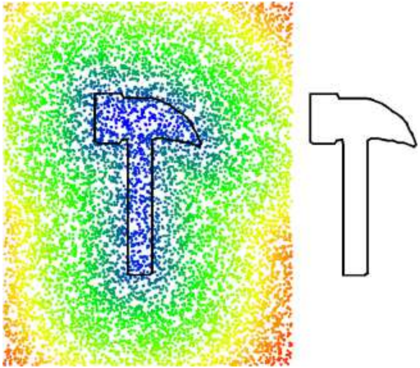
Neural Distance Fields: State of the Art Approach

- Define a neural network f_θ with parameters $\theta \in \mathbb{R}^K$
- Optimize over θ to minimize some loss function

$$\mathcal{L}_{fit} = \mathbb{E}_{x \in \mathbb{R}^n} \left[(f_\theta(x) - S(x))^2 \right]$$

$$\mathcal{L}_{normal} = \mathbb{E}_{x \in \partial\Omega} \left[(\nabla f_\theta(x) - n_x)^2 \right]$$

$$\mathcal{L}_{eikonal} = \mathbb{E}_{x \in \mathbb{R}^n} \left[(||\nabla f_\theta(x)|| - 1)^2 \right]$$



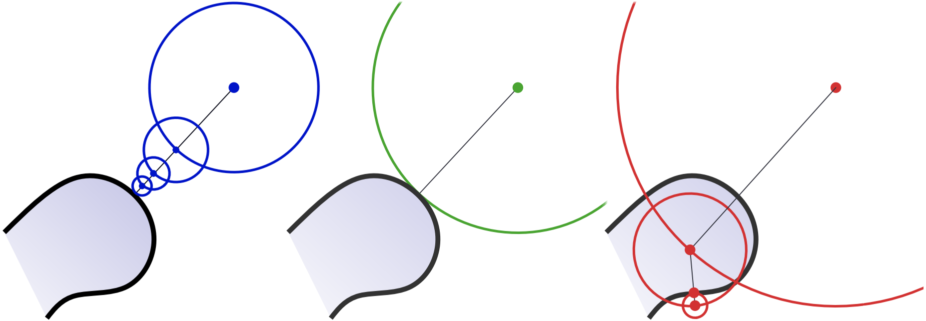
Dataset $\{(x_i, S(x_i))\}$

Neural Distance Fields: Limitations

f_θ does not *exactly* satisfy the Eikonal equation

Geometric queries are guaranteed only if $||\nabla f_\theta|| \leqslant 1$

$\hookrightarrow$ f_θ should be **1-Lipschitz**.



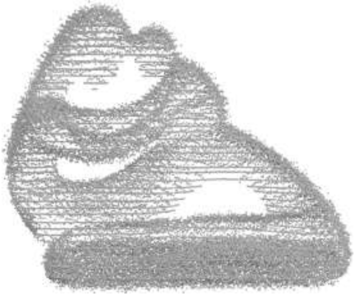
$|f| < |S|$ S $|f| > |S|$

Neural Distance Fields: Limitations

Problem: ground truth distances may not be available in practice.

Triangle soup

Incomplete point cloud



Noisy point cloud

1-Lipschitz Neural Distance Fields

1-Lipschitz Neural Networks

Consider a (feed-forward) neural network $f_\theta = f^{(L)} \circ ... \circ f^{(1)}$ with L layers.
Specifying its architecture defines a functional space $\mathcal{F} = \left\{ f_\theta \mid \theta \in \mathbb{R}^K \right\}$

[1] *"A Unified Algebraic Perspective on Lipschitz Neural Networks"*, Araujo et al., ICLR (2023)

1-Lipschitz Neural Networks

Consider a (feed-forward) neural network $f_\theta = f^{(L)} \circ ... \circ f^{(1)}$ with L layers.
Specifying its architecture defines a functional space $\mathcal{F} = \left\{ f_\theta \mid \theta \in \mathbb{R}^K \right\}$

Idea: Define $\mathcal{F}$ as a subset of all 1-Lipschitz functions.

[1] *"A Unified Algebraic Perspective on Lipschitz Neural Networks"*, Araujo et al., ICLR (2023)

1-Lipschitz Neural Networks

Consider a (feed-forward) neural network $f_\theta = f^{(L)} \circ \ldots \circ f^{(1)}$ with L layers.
Specifying its architecture defines a functional space $\mathcal{F} = \left\{ f_\theta \mid \theta \in \mathbb{R}^K \right\}$

Idea: Define $\mathcal{F}$ as a subset of all 1-Lipschitz functions.

1-Lipschitz Layer[1]

$$x \mapsto x - 2WD^{-1}\sigma(W^T x + b)$$

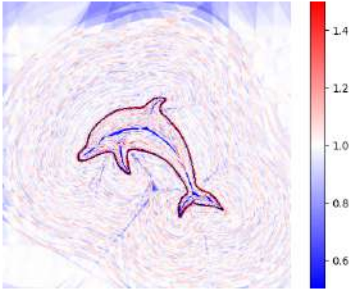
where:

$$\sigma(x) = \text{ReLU}(x) = \max(x, 0)$$
$$D = \text{diag}\left(\sum_j |(W^T W)_{ij} \exp(q_j - q_i)|\right)$$

Parameters to optimize: W, b, q

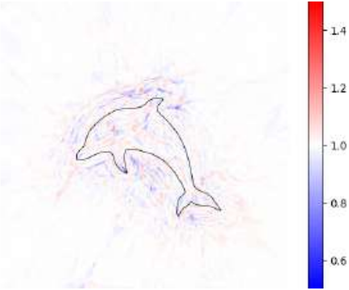
[1] *"A Unified Algebraic Perspective on Lipschitz Neural Networks"*, Araujo et al., ICLR (2023)

Plot of Gradient Norm



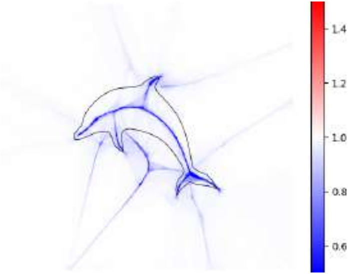
$\min \mathcal{L}_{fit}$

$\max ||\nabla f_\theta|| =$ **13.8**



$\min \mathcal{L}_{fit} + 0.1\mathcal{L}_{eikonal}$

$\max ||\nabla f_\theta|| =$ **1.4**



$\min \mathcal{L}_{fit}$ with
Lipschitz architecture

$\max ||\nabla f_\theta|| =$ **0.998**

Learning a Distance Field without Ground Truth

Learning a SDF from occupancy labels with Lipschitz networks

Suppose we can determine $y : \mathbb{R}^n \to \{-1, 1\}$ defined as:

$$y(x) = \begin{cases} -1 \text{ if } x \in \Omega \\ 1 \text{ if } x \notin \Omega \end{cases}$$

[2] *"Achieving Robustness in Classification Using Optimal Transport with Hinge Regularization"*, Serrurier et al., CVPR (2021)

Learning a SDF from occupancy labels with Lipschitz networks

Suppose we can determine $y : \mathbb{R}^n \to \{-1, 1\}$ defined as:

$$y(x) = \begin{cases} -1 \text{ if } x \in \Omega \\ 1 \text{ if } x \notin \Omega \end{cases}$$

hinge-Kantorovitch-Rubinstein loss[2]

$$\mathcal{L}_{hKR} = \mathbb{E}_x\left[-y f_\theta(x)\right] + \lambda \mathbb{E}_x\left[\max(0, m - y f_\theta(x))\right], \quad m, \lambda > 0$$

[2] *"Achieving Robustness in Classification Using Optimal Transport with Hinge Regularization"*, Serrurier et al., CVPR (2021)

Learning a SDF from occupancy labels with Lipschitz networks

Suppose we can determine $y : \mathbb{R}^n \to \{-1, 1\}$ defined as:

$$y(x) = \begin{cases} -1 \text{ if } x \in \Omega \\ 1 \text{ if } x \notin \Omega \end{cases}$$

hinge-Kantorovitch-Rubinstein loss[2]

$$\mathcal{L}_{hKR} = \mathbb{E}_x\left[-y f_\theta(x)\right] + \lambda \mathbb{E}_x\left[\max(0, m - y f_\theta(x))\right], \quad m, \lambda > 0$$

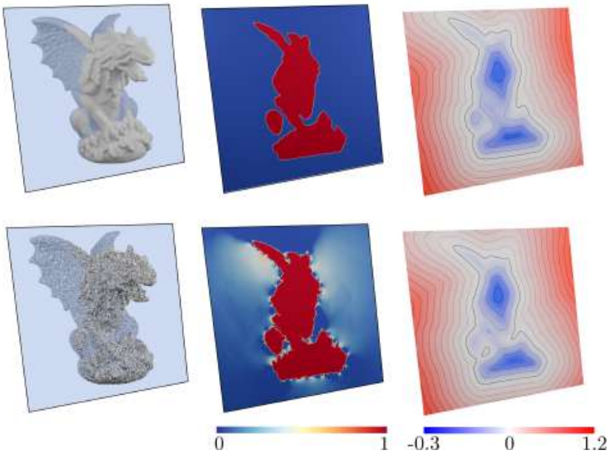
$f^* = \text{argmin}_\theta \mathcal{L}_{hKR}$ over 1-Lipschitz functions is a SDF up to margin parameter m

[2] *"Achieving Robustness in Classification Using Optimal Transport with Hinge Regularization"*, Serrurier et al., CVPR (2021)

Generalized Winding Number for Inside/Outside Partitioning

$$w_{\partial\Omega}(x) = \frac{1}{4\pi}\int_{\partial\Omega} d\Theta(x)$$

$w \in \{0, 1\}$ for manifolds
$w \in [0, 1]$ for point clouds[3]



[3] *"Fast Winding Numbers for Soups and Clouds"*, Barill et al., ACM Transactions on Graphics (2018)

Overview of the method

1. **Input:** Point cloud or triangle soup representing $\partial\Omega$
2. Sample points x_i uniformly in a loose box around Ω
3. Use the winding number to assign $y_i \in \{-1, 1\}$ to x_i
4. Minimize $\mathcal{L}_{hKR}$ over (x_i, y_i) for some 1-Lipschitz architecture

Input

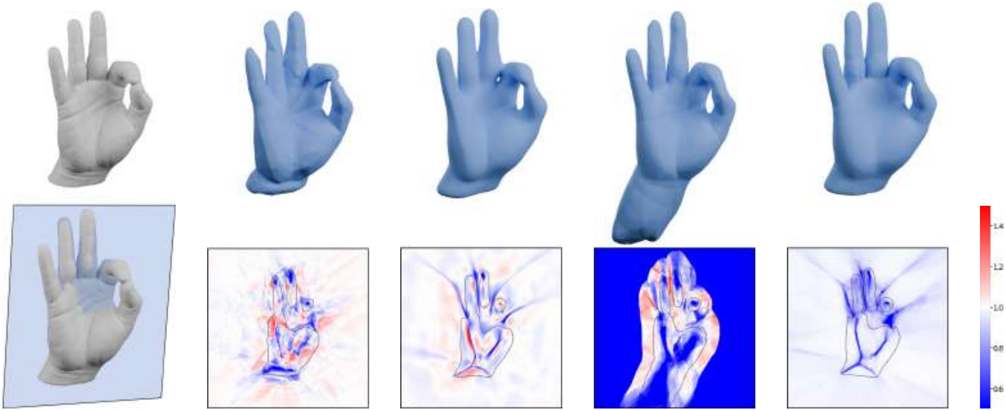
Samples

0 level set

SDF

Results

Surface Reconstruction and Gradient Correctness



Ground Truth   MLP[1]   SIREN[2]   SALD[3]   Ours

[1] *"On the Effectiveness of Weight-Encoded Neural Implicit 3D Shapes"*, Davies et al., (2021)
[2] *"Implicit Neural Representations with Periodic Activation Functions"*, Sitzmann et al., NeuRIPS (2020)
[3] *"SALD: Sign Agnostic Learning with Derivatives"*, Atzmon and Lipman, ICLR (2020)

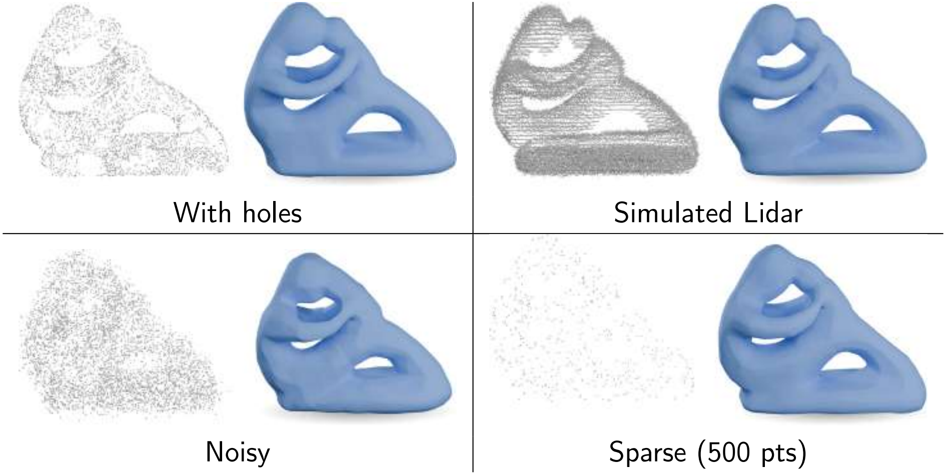
Direct Geometrical Queries

Ray Marching

Level Set Sampling

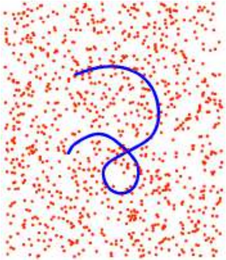
Skeleton Sampling

Resistance to corrupted inputs

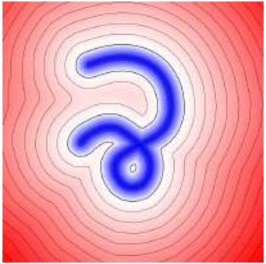


Unsigned Distance Function for Curves and Open Surfaces

Take $y = -1$ on $\partial\Omega$ and $y = 1$ on $\mathbb{R}^n \backslash \partial\Omega$. Loss margin m acts as a thickness parameter.



Input point cloud

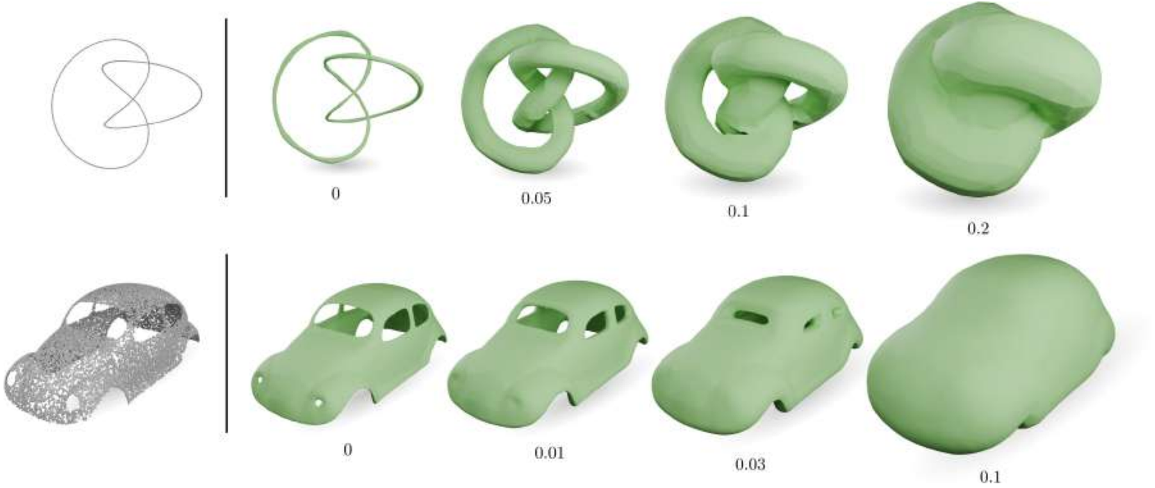


$m = 10^{-1}$

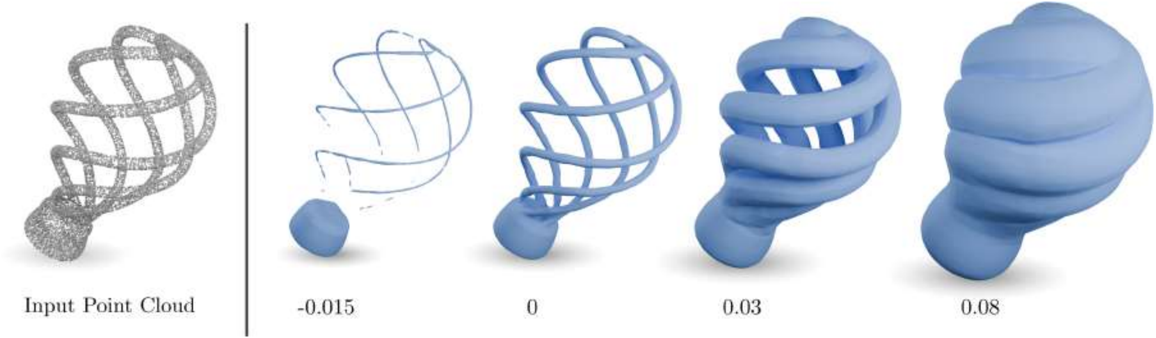
$m = 10^{-2}$

$m = 10^{-3}$

Unsigned Distance Function for Curves and Open Surfaces



Thank you for your attention!

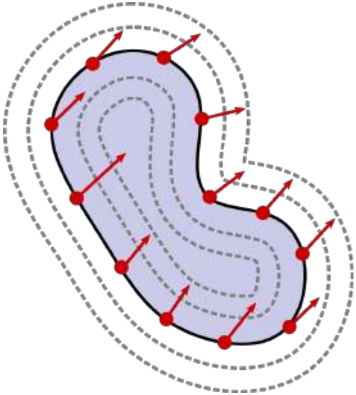


https://github.com/GCoiffier/1-Lipschitz-Neural-Distance-Fields

Limitations and Future works

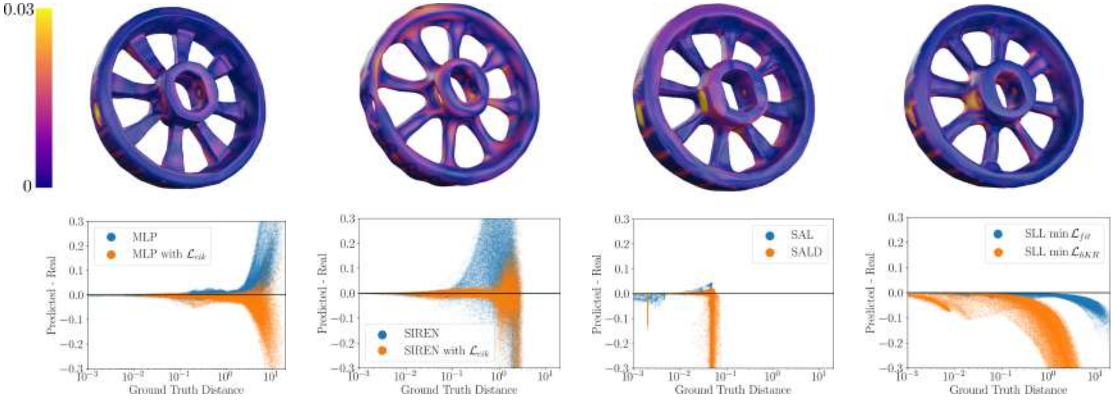
Capture of small details

Capture of sharp edges



Update and deformations

Our Methods Always Underestimates the True Distance



MLP[1] SIREN[2] SALD[3] Ours

---

[1] *"On the Effectiveness of Weight-Encoded Neural Implicit 3D Shapes"*, Davies et al., (2021)
[2] *"Implicit Neural Representations with Periodic Activation Functions"*, Sitzmann et al., NeuRIPS (2020)
[3] *"SALD: Sign Agnostic Learning with Derivatives"*, Atzmon and Lipman, ICLR (2020)

Ricci, A. (1973). A constructive geometry for computer graphics. *The Computer Journal, 16*(2), 157–160. https://doi.org/10.1093/comjnl/16.2.157

Sharp, N., & Jacobson, A. (2022). Spelunking the deep: Guaranteed queries on general neural implicit surfaces via range analysis. *ACM Transactions on Graphics*, *41*(4), 1–16. https://doi.org/10.1145/3528223.3530155

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, *21*(4), 163–169. https://doi.org/10.1145/37402.37422

Hart, J. (1995). Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces. *The Visual Computer*, *12*. https://doi.org/10.1007/s003710050084

Sawhney, R., & Crane, K. (2020). Monte Carlo geometry processing: A grid-free approach to PDE-based methods on volumetric domains. *ACM Transactions on Graphics*, *39*(4), 123:123:1–123:123:18. https://doi.org/10.1145/3386569.3392374

Gropp, A., Yariv, L., Haim, N., Atzmon, M., & Lipman, Y. (2020). Implicit Geometric Regularization for Learning Shapes.

Araujo, A., Havens, A., Delattre, B., Allauzen, A., & Hu, B. (2023). A Unified Algebraic Perspective on Lipschitz Neural Networks.

Serrurier, M., Mamalet, F., Gonzalez-Sanz, A., Boissin, T., Loubes, J.-M., & Del Barrio, E. (2021). Achieving robustness in classification using optimal transport with hinge regularization. *CVPR*, 505–514. https://doi.org/10.1109/CVPR46437.2021.00057

Barill, G., Dickson, N. G., Schmidt, R., Levin, D. I. W., & Jacobson, A. (2018). Fast winding numbers for soups and clouds. *ACM Transactions on Graphics*, *37*(4), 1–12. https://doi.org/10.1145/3197517.3201337

Davies, T., Nowrouzezahrai, D., & Jacobson, A. (2021). On the Effectiveness of Weight-Encoded Neural Implicit 3D Shapes. https://doi.org/10.48550/arXiv.2009.09808

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., & Wetzstein, G. (2020). Implicit Neural Representations with Periodic Activation Functions. *NeuRIPS*, *33*, 7462–7473.

Atzmon, M., & Lipman, Y. (2020). Sald: Sign agnostic learning with derivatives. *ICLR*.