Unsupervised Learning of Signed Distance Functions from 3D Point-Clouds

Tran Ba Quang 1252013 Supervised by Fayolle Pierre-Alain

Abstract

The purpose of this work is to propose a new method for constructing the signed distance function to a given 3D shape defined by an oriented point-cloud. Our approach to this problem consists in approximating the signed distance function by solving a p-Poisson problem via a deep neural network. The parameters of the neural network are obtained by minimizing a loss function involving the p-Poisson problem to be solved. The approximation of the signed distance function to a given 3D shape is then available in the form of a deep neural network that can be queried at any point in space. A trade-off between smoothness and accuracy of the approximated distance function is controlled by the parameter p.

1 Introduction

Implicit neural representations have proven to be effective for learning 3D shapes (see Section 2 for references). These implicit functions depict the surface as their zero-level set, and can be learned via Multi-Layer Perceptrons (MLPs). Being continuous functions, implicit neural representations are independent of resolutions, making them a compact method for storing the shape of 3D objects.

Given the benefits of implicit functions in learning 3D shapes, recent works have put a great amount of effort in approximating them by deep neural networks, in particular a lot of interest has been focused on learning approximations of the signed distance function to a given shape. Previous works did this by solving the eikonal equation via a deep neural network [12, 22], however, both of them can suffer from undesired artifacts [16, 23]. Additionally, it is sometimes necessary to trade approximation quality to smoothness. Indeed, the distance function has undefined derivatives at the cut-locus, the set of points with at least two different shortest paths to the surface.

In this paper, we want to construct the signed distance function from 3D point-clouds (a set of 3D points sampled on the surface of an object). Although point-cloud data are now easier to achieve due to the accessibility of 3D scanners or techniques such as photogrammetry, such data-sets are difficult to work with as they lack in connectivity and structure. Our approach is to feed them in to a MLP and find the signed distance functions by solving a p-Poisson problem. Solutions are searched as deep neural networks, whose parameters are trained by minimizing a loss function involving the p-Poisson problem of interest. To avoid artifacts, we also include additional terms in the definition of the loss function, such as a term to prevent zero-level sets at off-surface points.

2 Related Work

Surface reconstruction Many approaches for reconstructing surfaces from point-cloud data were proposed. Carr et al. reconstruct surfaces by fitting Radial Basis Functions (RBF) to the input point-cloud, it involves solving a dense linear system of equations [6]. Ohtake et al. reconstruct the surface by fitting quadric shape functions locally then blend the local shape functions into a global approximation with a partition of unity [19]. Kazhdan et al. approximate the shape indicator function by solving the Poisson problem $\Delta u = \mathrm{div}(\boldsymbol{n})$, where the vector field $\boldsymbol{n}$ is obtained by extrapolating the sampled surface normals [13]. This approach, however, tends to over-smooth the surfaces. To tackle this problem, they add positional constraints to the previous algorithm [14].

Implicit neural representation. Approximating the implicit function via a neural network is one of the popular approaches over the last few years for surface reconstruction. Some methods approximate the signed distance function by using some of its unique properties, to be specific, solving the eikonal problem [12,22], solving variational problems [10, 18], or applying both eikonal term and second order derivatives to the training set [5] Other works construct the signed distance functions based on ground truth distances [1, 2, 9, 11, 20, 21, 25]. Yifan et al. propose a method for improving the fidelity of the reconstructed surface by sampling iso-points [26]. Yariv et al. introduce an approach to reconstructing the surface from images [24]. Chibane et al. approximate the unsigned distance function based on the ground truth, which allow them to reconstruct both closed and open surfaces [7]. Lipman propose a new loss term for learning the implicit neural representation, inspired by the theory of phase transitions of fluids [16]. Wang et al. improve the details of previous work [12] by incorporating a B-Spline function as an encoding function [23]. There is also some research focused on reconstructing the surfaces only [4].

Incorporating derivatives into training neural networks. Derivatives have been proven to be useful in training neural networks in general [8]. For the problem of surface reconstruction in particular, many approaches use derivatives to solve the unique properties of the signed distance function as mentioned above [5, 10, 12, 18, 22]. The approach proposed in [2] show significant improvement in recovering details when adding a derivative term to the loss function proposed in [1].

3 Background

3.1 Signed distance function

The signed distance function $f(\boldsymbol{x})$ of a set Ω in a metric space is a function that returns the distance between a point $\boldsymbol{x}$ and $\partial\Omega$, the boundary of Ω. The sign of $f(\boldsymbol{x})$ is determined by whether $\boldsymbol{x}$ is inside or outside Ω. Without loss of generality, we define $f(\boldsymbol{x})$ to be negative inside and positive outside the domain Ω

$$f(\boldsymbol{x}) = \begin{cases} -d(\boldsymbol{x}, \partial\Omega) & \text{if } \boldsymbol{x} \in \Omega \\ d(\boldsymbol{x}, \partial\Omega) & \text{if } \boldsymbol{x} \in \Omega^C \end{cases} \tag{1}$$

One property of the signed distance function is that it satisfies the eikonal equation

$$\|\nabla f(\boldsymbol{x})\| = 1 \tag{2}$$

where $\|\cdot\| = \|\cdot\|_2$ is the euclidean 2-norm, and $\nabla f(\boldsymbol{x})$ denotes the gradient of the function $f(\boldsymbol{x})$

$$\nabla f(\boldsymbol{x}) = \left[\frac{\partial f}{\partial x}\ \frac{\partial f}{\partial y}\ \frac{\partial f}{\partial z}\right]^T. \tag{3}$$

One can obtain an approximation of the signed distance function by solving the p-Poisson problem [3]

$$\Delta_p f(\boldsymbol{x}) = 1 \tag{4}$$

where Δ_p is the p-Laplacian

$$\Delta_p f(\boldsymbol{x}) = \nabla \cdot (\|\nabla f(\boldsymbol{x})\|^{p-2} \nabla f(\boldsymbol{x})) \tag{5}$$

and $\nabla \cdot F$ denotes the divergence of a vector field F

$$\nabla \cdot F = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial F}{\partial z}. \tag{6}$$

One can show that as $p \to \infty$, the solution to (4) converges to the exact distance function.

3.2 Deep neural network

A deep neural networks with inputs $\boldsymbol{x}$ and parameters θ, denoted as $u(\boldsymbol{x}, \theta)$, can be trained to represent different functions, in our case, it will be used to approximate the signed distance function to a given 3D shape. In this paper, we use a feed-forward fully connected deep neural network defined by

$$\begin{aligned} x_L &= W_L \sigma(\boldsymbol{x}_{L-1}) + b_L \\ \boldsymbol{x}_{L-1} &= W_{L-1} \sigma(\boldsymbol{x}_{L-2}) + b_{L-1} \\ &\dots \\ \boldsymbol{x}_2 &= W_2 \sigma(\boldsymbol{x}_2) + b_2 \\ \boldsymbol{x}_1 &= W_1 \sigma(\boldsymbol{x}) + b_1 \end{aligned}$$

where L is the number of layers of the neural network, W_i and b_i correspond to the parameters θ, $\sigma(\cdot)$ is a non-linear activation and $u(\boldsymbol{x}, \theta) = x_L$.

Feed-forward deep neural networks are trained by the backpropagation algorithm. Each training sample $\boldsymbol{x}$ is fed into the deep neural networks to get the output x_L, the output error is then calculated and backpropagated to compute the error of each layer. Finally, the parameters W_i and b_i are adjusted with regard to the error of each layer.

4 Approach

4.1 Surface reconstruction

Our approach consists in representing the signed distance function (or its approximation) of a given 3D shape via a deep neural network.

Given an input 3D point-cloud $\mathcal{X} = \{\boldsymbol{x_i}\}$ where $\boldsymbol{x_i} = (x_i, y_i, z_i)$ are points in space, sampled on the surface S of an object, we want to approximate the signed distance function to the surface S by optimizing the parameters θ of a deep neural network $u(\boldsymbol{x}, \theta)$.

Firstly, in order to enforce that the function u vanishes on $\mathcal{X}$, we introduce the geometric loss

$$\ell_g(\theta) = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x_i} \in \mathcal{X}} |u(\boldsymbol{x_i}, \theta)| \tag{7}$$

Secondly, we want $u(\cdot, \theta)$ to approximate the signed distance function to S. In order to achieve this, we want u to be solution to the the p-Poisson problem $\Delta_p u = 1$. With $\boldsymbol{x}$ being random points in space, we enforce the p-Poisson constraint by introducing the following loss term

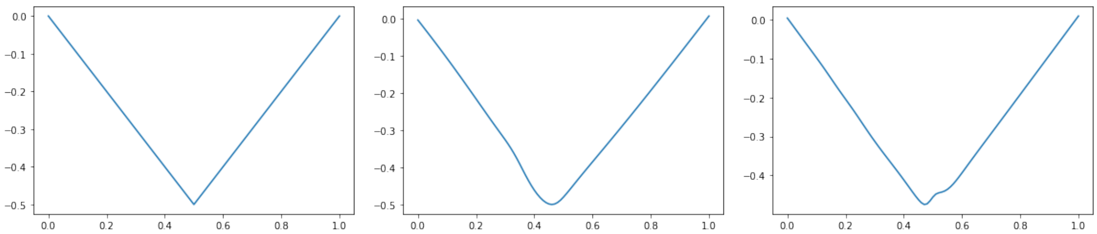
$$\ell_p(\theta) = \mathbb{E}_{\boldsymbol{x}}(\Delta_p u(\boldsymbol{x}, \theta) - 1)^2 \tag{8}$$

Finally, we need a constraint to prevent undesired artifacts, e.g., extra zero level-sets away from the surface. This is done by adding the loss term [22]

$$\ell_c = \mathbb{E}_{\boldsymbol{x} \notin \mathcal{X}}(\psi(u(\boldsymbol{x}, \theta))) \tag{9}$$

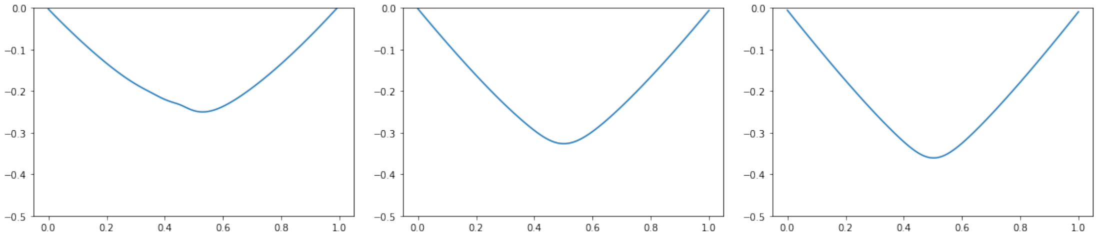
where

$$\psi(u(\boldsymbol{x}, \theta)) = exp(-\alpha\, |u(\boldsymbol{x}, \theta)|), \alpha \gg 1 \tag{10}$$



GT IGR SIREN

Figure 1: Signed distance function approximated with [1] and [22] on a 1D data set.



$p = 4$ $p = 6$ $p = 8$

Figure 2: Signed distance function approximated by solving the p-Poisson problem on a 1D data set.

By weighting each term, our final loss function becomes

$$\ell(\theta) = w_g \ell_g(\theta) + w_p \ell_p(\theta) + w_c \ell_c(\theta). \tag{11}$$

In our experiments, we used w_g, w_p, w_c to be 10, 1, and 0.5 respectively.

4.2 Implementation

Neural network architecture. We use the same neural network architecture as described in [20]: a feed-forward fully connected network with 8 hidden layers of hidden size 512, and a skip connection from the input to the fourth layer. Each layer is followed by the softplus activation function

$$\text{softplus}(x) = \frac{1}{\beta} \ln(1 + \exp(\beta x)). \tag{12}$$

The weights θ of the neural network are initialized with the same method as proposed in [1], in which $\theta = \theta_0$ such that the deep neural network $u(\boldsymbol{x}, \theta_0)$ is an approximation of the signed distance function to a unit sphere.

Training. The weights θ of the neural network are optimized by Stochastic Gradient Descent. We use the Adam optimizer [15] with constant learning rate of 0.0001. We draw 16384 sample points from the input point-cloud at each iteration. For evaluating (8), we sample 256 points in 2D and 8192 points in 3D from an uniform distribution at each iteration.

5 Experiments and Results

5.1 Results in 1D

We tried first to approximate the signed distance function to the boundary of the segment $[0, 1]$ in 1D. The corresponding "surface" point-cloud is the set $\mathcal{X} = \{0, 1\}$. We made some comparisons between the results from our approach and the results from IGR [12] and SIREN [22], as well as between the results obtained for different values of p.

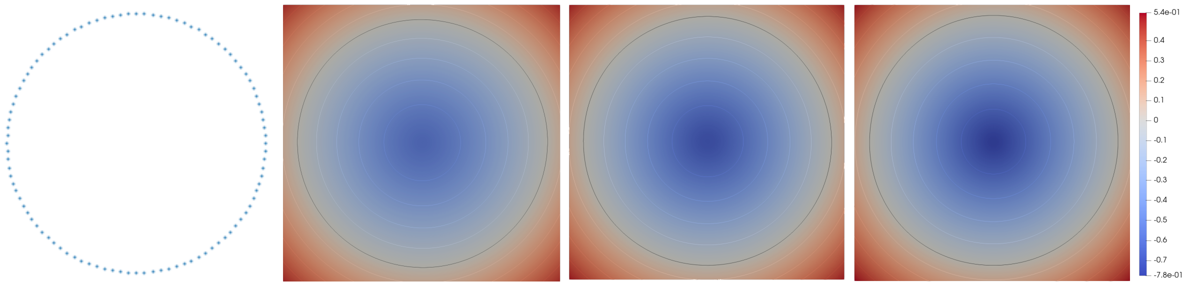
Both IGR and SIREN attempts to learn the signed distance functions by solving the eikonal equation via a deep neural network. The approximated signed distance functions are shown in Fig. 1.

Compared to the previous methods, we can produce smooth approximations of the signed distance function, while at the same time, avoid poor solutions. Figure 2 also shows that the trade-off between smoothness and accuracy can be adjusted by changing p.

5.2 Results in 2D and 3D

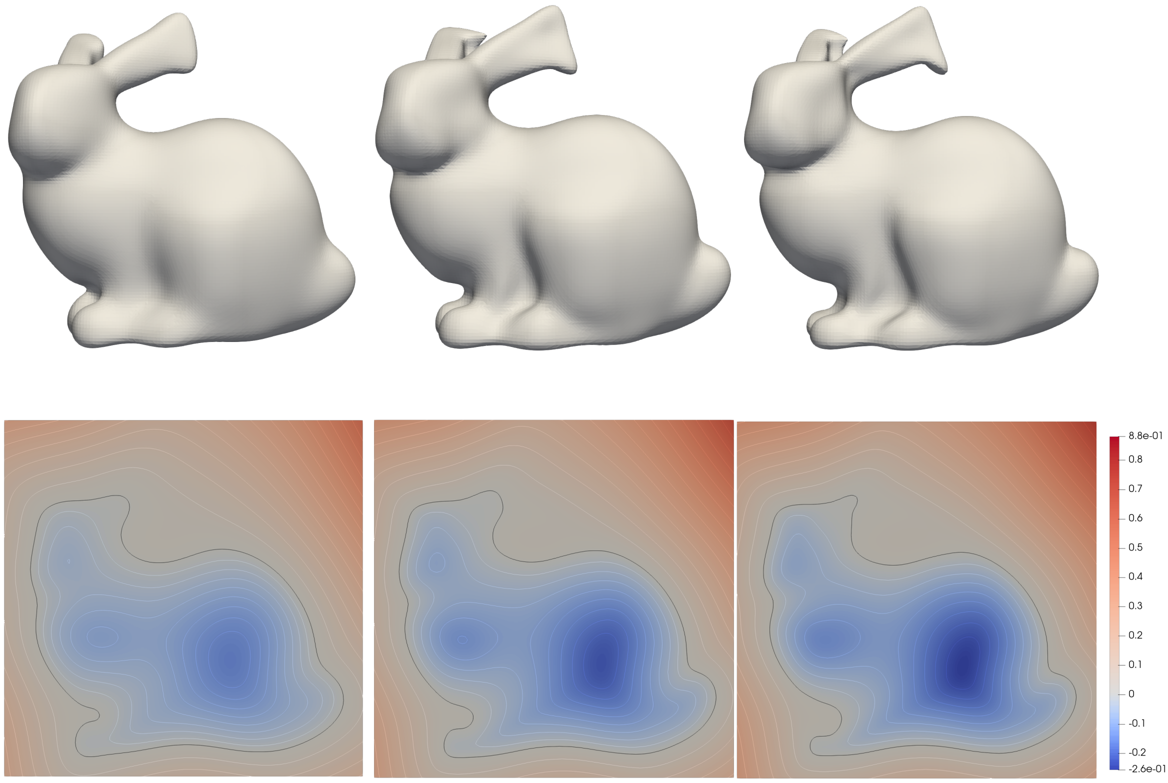
We trained our model on 2D and 3D point-clouds for 2500 epochs. The results are shown in Fig. 3 for the 2D example and in Fig. 5 for the 3D example.

For the 2D experiments, we used an input point-cloud with 100 points sampled on a unit circle. Our method produced a precise boundary and the approximate distance gets closer and closer to the ground truth as the p



Input point-cloud $p = 4$ $p = 6$ $p = 8$

Figure 3: Signed distance function to a circle approximated by solving the p-Poisson problem, the boundaries are shown in black.



$p = 4$ $p = 6$ $p = 8$

Figure 5: Signed distance function approximated by solving the p-Poisson problem on the Stanford Bunny point-cloud. Upper: reconstructed surface. Lower: a slice of each result (zero-level sets are shown in black)

value is increased (from 4 to 8 in this experiment).
For the 3D experiments, we used the Stanford Bunny point-cloud made of 35947 points. By increasing p, not only the distance but the recovered details, as well, get more accurate.

The surface (curve in 2D) is obtained from the zero level-set of the neural network $u(\mathbf{x})$ by the Marching Cubes algorithm [17]. We used a grid resolution of 128^2 for the 2D example and 128^3 for the 3D example.

6 Conclusion and discussion

We have presented a method for approximating the signed distance function to a shape given an input point-cloud. Our method is based on the fact that the solution to the p-Poisson problem (4) is a smooth approximation of the signed distance function. We search for a solution to (4) as a deep neural network. Its parameters are obtained by minimizing the loss function (11). Through experiments, we show that this approach is capable of solving the p-Poisson equation effectively, and thus delivering an approximation of the signed distance function to the input shape.

Drawbacks. Nevertheless, we observed some drawbacks in our approach. First, the results are dependent on the weights assigned to each loss term. Setting them properly required a lot of experiments.
The parameters θ of the deep neural networks have to be trained for each individual shape. The training requires a lot of iterations. On the other hand, once a model has been trained, its corresponding signed distance can be queried efficiently.

Possible directions for future works. There are several directions that could be considered to improve the current work. Firstly, one can use a simple normalization to improve the accuracy of the distance approximation near the boundary [3]. Secondly, we have to experiment with the effectiveness of our method on noisy data sets. Possibly additional terms should be added in the loss function (11) to deal with noise in the input data.

References

[1] M. Atzmon and Y. Lipman, "Sal: Sign agnostic learning of shapes from raw data," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 1, 2, 3

[2] M. Atzmon and Y. Lipman, "Sald: Sign agnostic learning with derivatives," 2020. 1, 2

[3] A. G. Belyaev and P.-A. Fayolle, "On variational and pde-based distance function approximations," Computer Graphics Forum, vol. 34, pp. 104–118, Wiley Online Library, 2015. 2, 5

[4] Y. Ben-Shabat and S. Gould, "Deepfit: 3d surface fitting via neural network weighted least squares," arXiv preprint arXiv:2003.10826, 2020. 1

[5] Y. Ben-Shabat, C. H. Koneputugodage, and S. Gould, "Digs : Divergence guided shape implicit neural representation for unoriented point clouds," 2021. 1, 2

[6] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 67–76, 2001. 1

[7] J. Chibane, A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," 2020. 1

[8] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Świrszcz, and R. Pascanu, "Sobolev training for neural networks," 2017. 2

[9] T. Davies, D. Nowrouzezahrai, and A. Jacobson, "On the effectiveness of weight-encoded neural implicit 3d shapes," 2021. 1

[10] W. E and B. Yu, "The deep ritz method: A deep learning-based numerical algorithm for solving variational problems," 2017. 1, 2

[11] P. Erler, P. Guerrero, S. Ohrhallinger, M. Wimmer, and N. Mitra, "Points2surf: Learning implicit surfaces from point clouds," A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds., Computer Vision – ECCV 2020, vol. 12350 of Lecture Notes in Computer Science, pp. 108–124, Springer International Publishing, Oct. 2020, ISBN 978-3-030-58558-7. 1

[12] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," Proceedings of Machine Learning and Systems 2020, pp. 3569–3579, 2020. 1, 2, 3

[13] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," Proceedings of the fourth Eurographics symposium on Geometry processing, vol. 7, 2006. 1

[14] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," ACM Transactions on Graphics (ToG), vol. 32, no. 3, pp. 1–13, 2013. 1

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. 3

[16] Y. Lipman, "Phase transitions, distance functions, and implicit neural representations," 2021. 1

[17] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," ACM siggraph computer graphics, vol. 21, no. 4, pp. 163–169, 1987. 4

[18] J. Müller and M. Zeinhofer, "Deep ritz revisited," 2020. 1, 2

[19] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," ACM Trans. Graph., vol. 22, no. 3, pp. 463—-470, Jul. 2003. 1

[20] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," 2019. 1, 3

[21] V. Sitzmann, E. R. Chan, R. Tucker, N. Snavely, and G. Wetzstein, "Metasdf: Meta-learning signed distance functions," 2020. 1

[22] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," arXiv, 2020. 1, 2, 3

[23] P.-S. Wang, Y. Liu, Y.-Q. Yang, and X. Tong, "Spline positional encoding for learning 3d implicit signed distance fields," 2021. 1

[24] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, R. Basri, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," 2020. 1

[25] W. Yifan, L. Rahmann, and O. Sorkine-Hornung, "Geometry-consistent neural shape representation with implicit displacement fields," 2021. 1

[26] W. Yifan, S. Wu, C. Oztireli, and O. Sorkine-Hornung, "Iso-points: Optimizing neural implicit surfaces with hybrid representations," 2021. 1