Supplementary Material for Unsupervised Inference of Signed Distance Functions from Single Sparse Point Clouds without Learning Priors

Chao Chen[1] Yu-Shen Liu[1] * Zhizhong Han[2]
[1]School of Software, BNRist, Tsinghua University, Beijing, China
[2]Department of Computer Science, Wayne State University, Detroit, USA
chenchao19@mails.tsinghua.edu.cn liuyushen@tsinghua.edu.cn h312h@wayne.edu

1. Network and Training

The MLP in surface paraeterization is formed by 5 fully connected layers of size 128, 256, 512, 256, and 3. We have ReLU on the first two layers, leaky ReLU on the third and fourth layers, and tanh on the final output layer. In SDF inference, MLP1 is formed by 10 fully connected layers. The first 9 layers all have a dimension of 512, and the last layer has a dimension of 128. We leverage the ReLU after each layer. Both MLP2 and MLP3 are formed by 1 fully connected layer of size 1, and do not leverage any activation function.

For each shape, we train our network in $40,000$ iterations on a NVIDIA GTX 1080Ti GPU using ADAM optimizer with a batch size of $5,000$ and an initial learning rate of 0.0001.

2. Sampling

Sampling 3D Queries We leverage a method introduced by NeuralPull [4] to sample queries around each point on the point cloud. We use Gaussian distribution with each point as its center and set the standard deviation as the distance to the 51th nearest neighbor in the point cloud. We sample 5000 queries around point clouds in each iteration.

Sampling 2D Points for Surface Parameterization We use uniform distribution in a range of $[0, 1]$ to sample 2D points. In each iteration, we sample $2,000$ points to generate a 3D point cloud to supervise the surface parameterization, and smaple $5,000$ points to generate a coarse surface estimation.

3. More Comparisons

We report more comparisons with the latest methods SIREN [5], IGR [3] under KITTI [2]. We provide IGR and SIREN the normals as input. We report visual comparison in Fig. 1. With sparse points, IGR cannot reconstruct reasonable surfaces, or even no surfaces. Compared to SIREN, our reconstructed surfaces are much smoother and more complete.

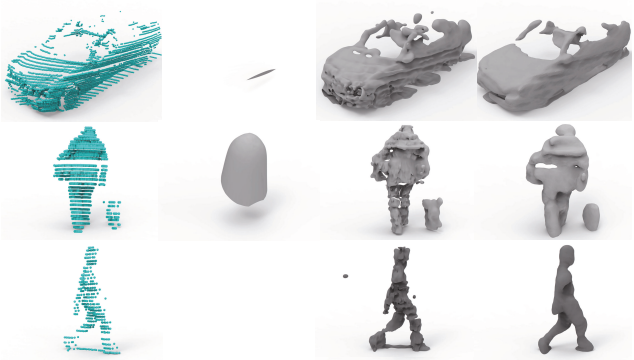

Figure 1. Visual comparisons with IGR and SIREN under KITTI.

4. More Visualization

We show our reconstructed surfaces from a large scale scan of a road from KITTI [1] in Fig. 2. We separate the point cloud into different sections, and use our method to reconstruct a mesh from each section. Our method can handle sparse point clouds and reconstruct smooth and complete surfaces.

We also use TSNE [6] to visualize the feature space learned for conducting TPS interpolation in Fig. 3. We can see that the features we learned to conduct TPS interpolation are compact, where features of queries (small dots) are closely surrounding features of surface points (big dots). This makes it easier to use surface points in basis functions for approximating SDFs based on TPS interpolation.

We visualize our optimization including learned level sets, the feature space to perform TPS interpolation and more results with real scans in our video. Please watch our

*The corresponding author is Yu-Shen Liu. This work was supported by National Key R&D Program of China (2022YFC3800600), the National Natural Science Foundation of China (62272263, 62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data.

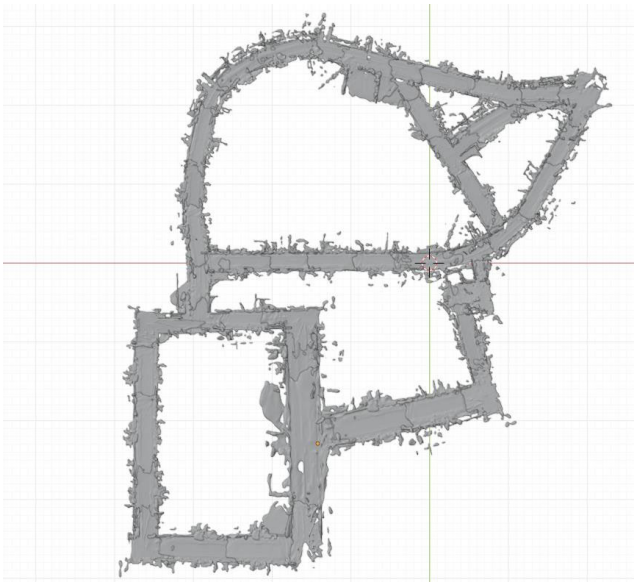

Figure 2. Visualization of our reconstruction from a large scale scan of a road KITTI. Please watch our video for more details.

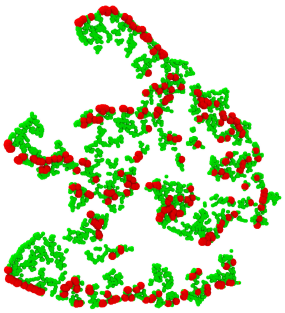

Figure 3. We use TSNE to visualize the feature space learned for conducting TPS interpolation. Large dots indicate features of surface points, small dots indicate features of 3D queries. Please watch our video for the visualization of feature spaces learned in different iterations.

video for more details.

method	parameter	time/min
NeedleDrop	5968897	26.83
ShapeGF	2668707	34.22
NeuralPull	**2169601**	**17.74**
OnSurf	7247723	31.60
Ours	2501553	20.24

Table 1. Comparison of training time and number of parameters.

5. Complexity

We report the comparison with others in terms of training time and number of parameters in Tab. 1. Our method has fewer parameters and shorter training time than other methods for learning SDFs from sparse points. While we have one additional branch for surface parameterization, which makes us have more parameters and a little longer training time than NeuralPull [4].

6. Code

Our code is available at https://github.com/chenchao15/NeuralTPS.

References

[1] Alexandre Boulch, Pierre-Alain Langlois, Gilles Puy, and Renaud Marlet. Needrop: Self-supervised shape representation from sparse point clouds using needle dropping. In *International Conference on 3D Vision*, 2021. 1

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition*, 2012. 1

[3] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv*, 2002.10099, 2020. 1

[4] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning*, 2021. 1, 2

[5] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, 2020. 1

[6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 1