Points2Surf
Learning Implicit Surfaces from Point Clouds
Supplementary Material

Philipp Erler[1] [0000−0002−2790−9279], Paul Guerrero[2] [0000−0002−7568−2849], Stefan Ohrhallinger[1,3] [0000−0002−2526−7700], Niloy J. Mitra[2,4] [0000−0002−2597−0914], and Michael Wimmer[1] [0000−0002−9370−2663]

[1] TU Wien, Favoritenstrasse 9-11 / E193-02, 1040 Vienna, AT
{perler,wimmer,ohrhallinger}@cg.tuwien.ac.at
[2] Adobe, 1 Old Street Yard, London EC1Y 8AF, UK
{guerrero,nimitra}@adobe.com
[3] VRVis, Donau-City-Straße 11, 1220 Vienna, AT
ohrhallinger@vrvis.at
[4] University College London, 66 Gower St, London WC1E 6BT, UK
n.mitra@cs.ucl.ac.uk

S.1 Overview

In this supplementary material, we provide more details on the training and evaluation setup for the DeepSDF baseline (Section S.2); timings of our method compared to our baselines (Section S.3); evaluations on a new dataset with more organic shapes (Thingi10k [1]), consisting of both quantitative (Section S.4) and qualitative (Section S.5) comparisons; and additional qualitative comparisons on the ABC dataset (Section S.6).

Our source code, pre-trained model, and dataset are available at: https://github.com/ErlerPhilipp/points2surf

S.2 DeepSDF Training and Evaluation Setup

We followed the method in the original paper with minor adaptations. For the training set, we take our query points and the corresponding Signed Distances (SD) as GT SDF samples. For the test sets, we take the point clouds from our dataset. For each point, we generate 2 SDF samples, one in positive and one in negative normal direction, with random offset. The normals are taken from the ground truth face closest to the sample. The corresponding SDs are the +-normal offset. We add 20% random samples from the unit cube with GT SD.

S.3 Timings

We compare the wall-clock times of Points2Surf vanilla to the baselines. Because AtlasNet is very fast, we show the mean of 3 runs. We take the times of reconstructing the 100 shapes of the ABC med-noise test set using 16 worker processes. With this, we can show a fair comparison that takes parallelization, loading times and different bottlenecks into account. We ran the timings on a consumer-grade PC with a Ryzen 7 3600X, 64 GB DDR4 RAM and a GTX 1070. The mean times per shape in seconds are: Points2Surf 712.1, DeepSDF 199.5, SPR 157.5, AtlasNet 0.1. Since DeepSDF and SPR need normals, we included 156.5 seconds for the normal estimation using PCPNet.

The heuristic and sign propagation allows us to reduce the number of inferred query points to 1.67% of a full grid in ABC var-noise. Assuming linear scaling, the inference time is reduced from almost 12 hours to 11.5 minutes. This speed-up comes at the cost of 19.2 seconds per shape. The heuristic is less efficient with noise. For the ABC no-noise, the number of query points is reduced to 0.77% and for ABC max-noise to 2.33%.

We trained a smaller version of e_{uniform} with only about 15% parameters. The mini version reduces the reconstruction time by roughly 68% compared to e_{vanilla} (from 11.5 to 3.7 minutes per mesh), at the cost of an increase in the reconstruction error of roughly 55% on ABC var-noise (Chamfer distance from 150.6 for e_{vanilla} to 234.3 for the small version of e_{uniform}).

S.4 Quantitative Comparison on Thingi10k

We show a quantitative comparison of reconstruction errors on the Thingi10k dataset in Table S1. The Chamfer distance between reconstructed and ground truth surfaces *averaged* over all shapes in a dataset is shown. Both the absolute value of the error multiplied by 100 (abs.), and the error relative to Point2Surf (rel.) are shown to facilitate the comparison. Our method consistently performs better than the baselines, due to its strong and generalizable prior. Note that Points2Surf was *not* retrained on the Thingi10k – this shows generalization results.

Table S1. Quantitative comparison of reconstruction errors on the Thingi10k dataset. Note that none of the methods was *not* retrained on the Thingi10k in order to test generalization to new data.

	DeepSDF		AtlasNet		SPR		Points2Surf	
	abs.	rel.	abs.	rel.	abs.	rel.	abs.	rel.
Thingi10k no-noise	9.16	6.48	5.29	3.74	1.78	1.26	**1.41**	**1.00**
Thingi10k med-noise	8.83	5.99	5.19	3.52	1.81	1.23	**1.47**	**1.00**
Thingi10k max-noise	12.28	4.68	4.90	1.87	3.23	1.23	**2.62**	**1.00**
Thingi10k sparse	9.56	4.54	5.64	2.68	2.35	1.12	**2.11**	**1.00**
Thingi10k dense	8.35	6.19	5.02	3.72	1.57	1.16	**1.35**	**1.00**
average	9.64	5.58	5.21	3.11	2.15	1.20	**1.79**	**1.00**

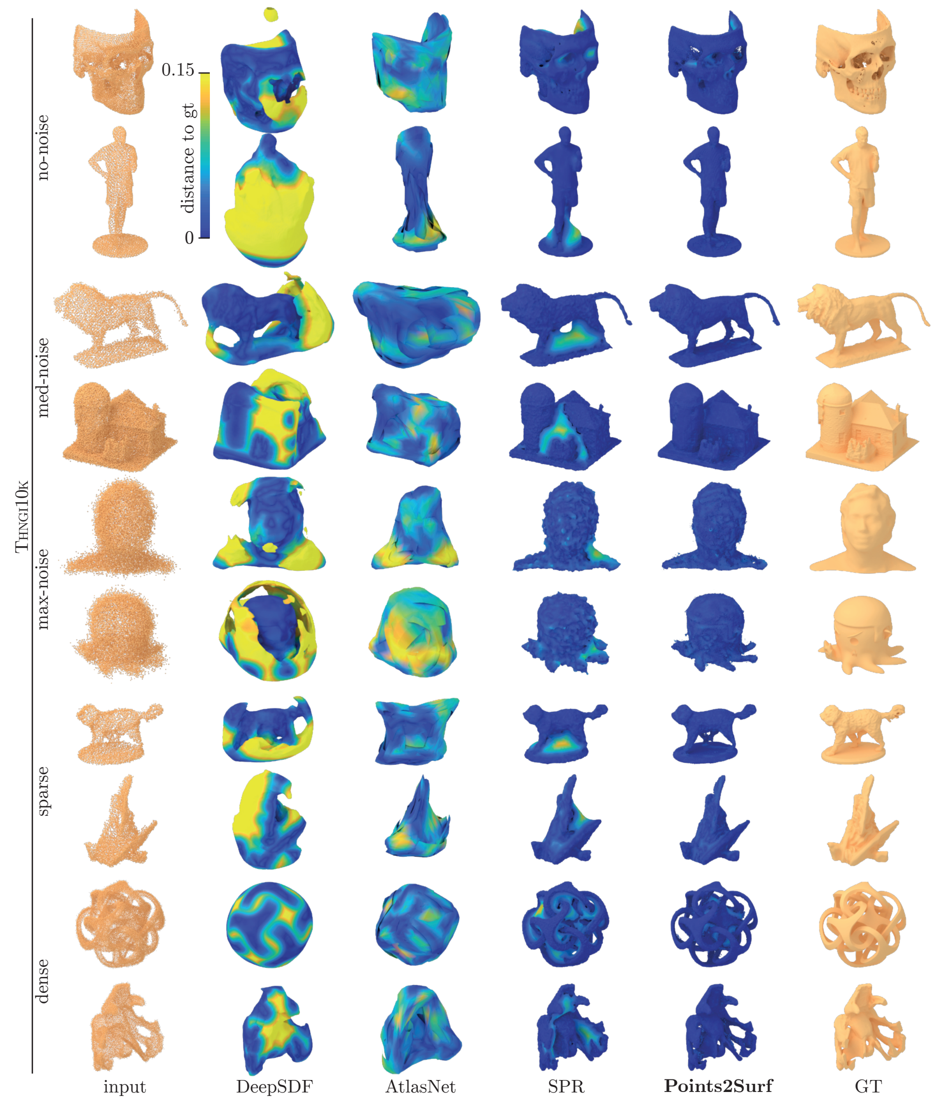

Fig. S1. Qualitative comparison of surface reconstructions on the THINGI10K dataset. We evaluate two examples from each dataset variant with each method. Colors show the distance of the reconstructed surface to the ground truth surface, shades of blue indicating low error.

S.5 Qualitative Results on Thingi10k

We evaluate POINTS2SURF on the THINGI10K dataset [1]. We take 100 meshes that are tagged with 'scan' or 'sculpture'. These objects are mostly animals, humans and faces, many of them realistic, some artistic. We create the same dataset variants as in the FAMOUS dataset: a version without noise (THINGI10K *no-noise*), a version with a medium noise strength $0.01L$ (Thingi10k *med-noise*), and a version with maximum amount of noise $0.05L$ (Thingi10k *max-noise*). Additionally we create sparser and denser point clouds by varying the number of scans: a variant with 5 scans instead of 10 (Thingi10k *sparse*), and a version with 30 scans (Thingi10k *dense*), both with a medium noise strength of $0.01L$. some qualitative results are shown in Figure S1.

S.6 Additional Qualitative Results on ABC

We provide additional qualitative comparisons on the ABC dataset in Figure S2. The evaluation setup is the same as for Figure 4 in the paper.

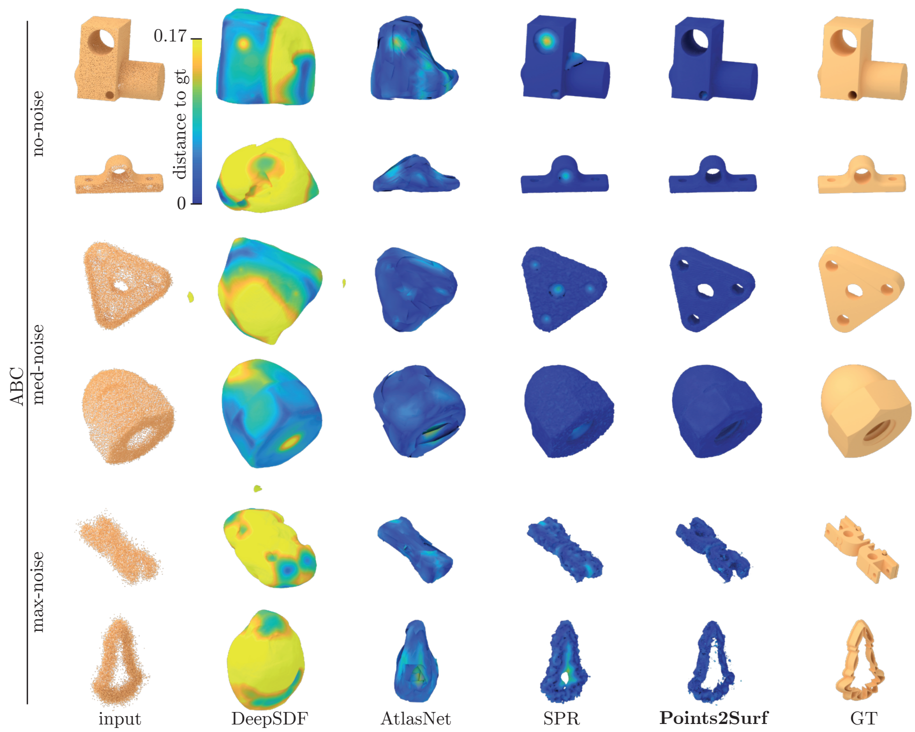

Fig. S2. Additional qualitative comparison of surface reconstructions on the ABC dataset. We evaluate two examples from each dataset variant with each method. Colors show the distance of the reconstructed surface to the ground truth surface.

References

1. Zhou, Q., Jacobson, A.: Thingi10k: A dataset of 10,000 3d-printing models. arXiv preprint arXiv:1605.04797 (2016)