OReX: Object Reconstruction from Planar Cross-sections Using Neural Fields
(Supplementary material)

Haim Sawdayee
The Blavatnik School of Computer Science
Tel Aviv University
haimsawdayee@mail.tau.ac.il

Amir Vaxman
School of Informatics
The University of Edinburgh
avaxman@inf.ed.ac.uk

Amit H. Bermano
The Blavatnik School of Computer Science
Tel Aviv University
amberman@tauex.tau.ac.il

We supplement and expand on details from our main document.

1. Implementation details

1.1. Sampling and extraction

We train our network for a total of 650 epochs. At the $0, 50, 100, 200, 300, 450$ epoch marks, we add new points to the dataset, where we always train on the last three such sets. The points are sampled from these distributions:

1. We compute the convex hull of the object. We then scale it up by 5% and sample 16384 points on it uniformly by selecting faces with probability proportional to area and sampling a point inside it using triangle point picking method.

2. We sample every plane uniformly with 2048 points, within a bounding box that is aligned to the principal axes of the contours.

3. For epoch 0, we initially sample $n = 2$ points uniformly on every edge, and off-surface points in distance $\epsilon = 2^{-4}$. In subsequent epoch marks we add $n = 2, 3, 3, 4, 5$ and $\epsilon = 2^{-5}, 2^{-5}, 2^{-6}, 2^{-6}, 2^{-7}$.

The mesh is extracted by dual-contouring on a grid sampled by inference of $f(x)$ (both values and gradients), where we use grid resolution 300^3.

1.2. Architecture

Our base architecture is a 7-hidden layer MLP of layer width 64. Unlike a typical Neural Field, our MLP is residual, that is it computes an offset from the input, rather than the absolute values (as seen in Fig. 2 of the main paper). Input coordinates are modulated using sin and cos Positional Encoding, with 5 learnable frequencies concatenated to the original input.

Refinement iterations: We run our prediction 10 times through our network in a recurrent manner, both during training and test time. Except for the OReXNet output, the MLP also produces a small hidden code of size 32 that is passed along the iterations to the network. The first iteration is fed with a learned constant C.

1.3. Training hyperparameters

For all experiments in the paper, we use the hinge loss with $\lambda = 10^{-4}$ and set the hinge point to be at $\alpha = 100$. We trained with initial learning rate $lr = 10^{-2}$ and reduced it by a factor of $\gamma = 0.9$ every 10 epochs.

1.4. Cross-section placement

The placement and the density of the cross-section planes is a determining factor for resulting quality—in fact, this is the entire challenge, as regions that are not sampled can only be extrapolated, as in Fig 2. To show the robustness of our method, our experiments show three patterns of selected cross-sections: randomly-oriented slices (Elephant, Oil lamp), regularly-spaced axis-aligned planes (Balloondog, Figure eight, Hand ok, Rose), and medically-generated slices (Skull, Hart, Vertebrate, Abdomen).

2. Additional experiments

We demonstrate the advantages of using neural fields interpolators, rather than traditional smooth choices (like the mean-value interpolant by [1]). In Fig 1 and Fig 2 the neural field demonstrates self-similarity that correctly interpolates regions that are not sampled.

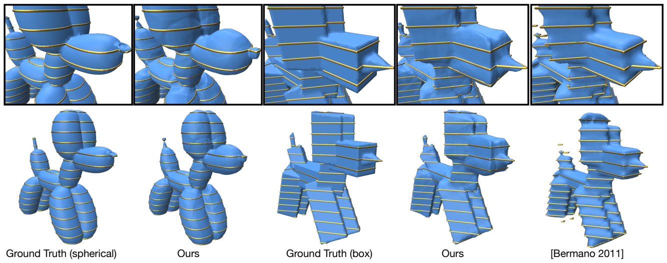

Figure 1. Self-similarities and how OReX leverages them. When the slices sample either a more round shape (left, ’round dog’) or a cube-like sharp shape (middle ’box dog’), the interpolation of the nose is consistent with the rest of the style, thanks to the natural self-similarities of the neural field. Methods that use a smoothing prior (e.g., [1], right) fail to capture the sharpness of the model.

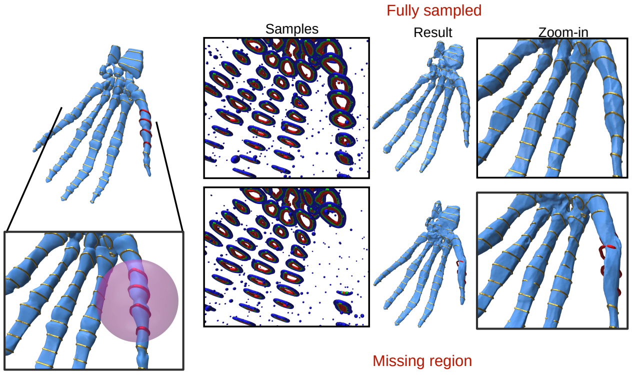

Figure 2. OReX naturally completes regions with missing samples, even as we omit them from the training process (red contours) Top row: the complete reconstruction. Bottom row: samples in the highlighted region are omitted. Everything else is trained as usual. Left to right: Grout Truth mesh, samples used in training, ORex reconstruction, with a zoomed-in view.

3. Quantitative comparisons

In Table 1, we expand our quantitative comparisons (Table 1 in the main text). We compare our method to three others: [1] is a method developed specifically for cross-section reconstruction with similar settings. We attempted to run the code for newer methods [3, 7], but could not produce results for or inputs.

In addition, we compare to two point-cloud-based methods [2,6], using our setting and with additional input information in the form of point normals. We measure performance using both global geometry measures (Hausdorf distance, Champfer Distance, and IoU of the reconstructed volumes), as well as metrics on the input cross-sections themselves (IoU in 2D). In all measures we demonstrate state-of-the-art results. In Fig 5 we visualized closest-point distances of our method.

4. Qualitative comparisons

Below are further visual comparisons with a recent point cloud method (fig. 3) and a concurrent reconstruction method based on slices (fig. 4). Our method produces watertight meshes that smoothly interpolate the input.

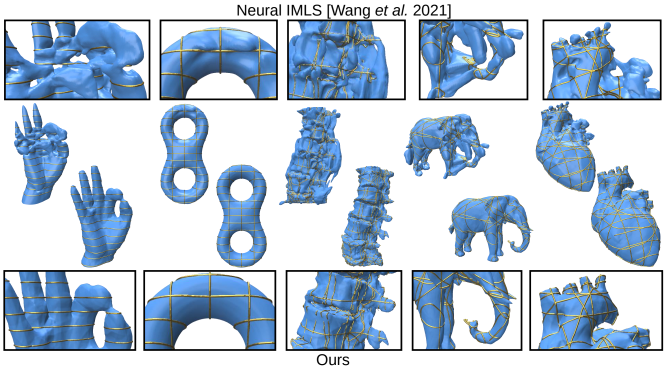

Figure 3. Comparison of OReX (bottom) to a recent point-cloud-based method [5] (top). Point-cloud methods typically expect a dense input covering the whole volume more or less uniformly; hence artifacts appear when feeding cross-sectional input.

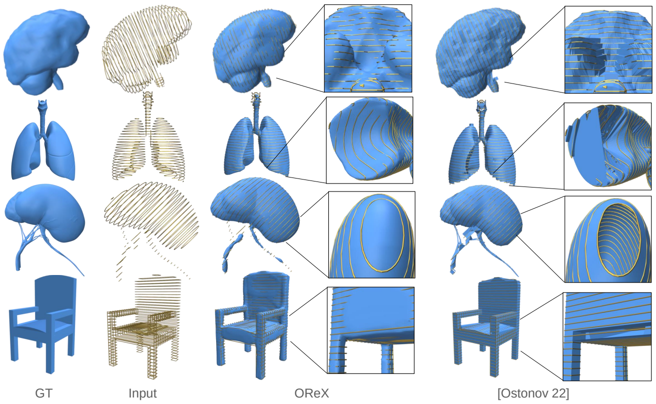

Figure 4. Comparison of OReX to a concurrent work by Ostonov [4]. Left to right: ground truth, input slices, our method, and Ostonov 22.

5. Stats

In Table 2, we report statistics on the inputs we used. Note the complexity of the shape as well as runtimes.

	Hausdorff distance					CD					IoU in 3*d*					IoU in 2*d*				
Input	OReX	Bermano et. al	point2mesh plane normals	point2mesh GT normals	Neural-IMLS	OReX	Bermano et. al	point2mesh plane normals	point2mesh GT normals	Neural-IMLS	OReX	Bermano et. al	point2mesh plane normals	point2mesh GT normals	Neural-IMLS	OReX	Bermano et. al	point2mesh plane normals	point2mesh GT normals	Neural-IMLS
Eight 15	**0.018**	0.065	0.219	0.046	0.237	**0.002**	0.020	0.029	0.005	0.035	**0.984**	0.865	0.842	0.961	0.680	**0.988**	0.984	0.795	0.980	**
Eight 20	**0.006**	0.033	0.045	0.014	0.052	**0.002**	0.016	0.013	0.004	0.004	**0.987**	0.893	0.915	0.974	0.971	**0.986**	0.976	0.961	0.971	0.975
Elephant	**0.056**	0.081	0.100	0.086	0.312	**0.006**	0.015	0.018	0.010	0.031	**0.966**	0.908	0.885	0.935	0.803	**0.975**	0.969	0.850	0.908	**
Balloon dog	**0.049**	0.194	0.078	0.086	0.264	**0.006**	0.021	0.015	0.010	0.044	**0.957**	0.868	0.897	0.928	0.659	**0.988**	0.977	0.926	0.956	**
hand ok	**0.063**	0.177	0.195	0.135	0.207	**0.008**	0.024	0.015	0.013	0.040	**0.955**	0.860	0.921	0.931	0.707	**0.987**	0.968	0.908	0.882	0.765
Armadillo	**0.050**	0.121	0.057	0.059	0.827	**0.009**	0.017	0.016	0.011	0.212	**0.939**	0.891	0.891	0.921	*	**0.964**	0.776	0.850	0.868	*

Table 1. Quantitative comparisons. We measure performance using the global metrics Hausdorff distance, Champfer Distance, and Intersection over Union (IoU) of the inner volume compared to the GT shape. In addition, we measure fidelity performance by reporting IoU of the "inside" regions on the input cross-sections. We compare our result to a dedicated cross-section-based reconstruction method [1], to two flavors of a point-cloud reconstruction method [2], and another recent point-cloud based method [6].

Input	#slices	#edges	#samples (last set)	Training time (h)	Meshing time (h)
Armadillo	26	11039	290412	2.90	0.22
Balloon dog	15	4579	138684	1.51	0.22
Eight (S)	15	2028	87664	1.06	0.22
Eight (M)	20	3178	120904	1.40	0.21
Elephant	24	10429	274116	2.75	0.23
Hand OK	15	4082	128744	1.42	0.22
Oil lamp	34	17387	433756	4.13	0.22
Abdomen	42	53448	1067530	10.62	0.23
Heart	25	4534	158264	1.77	0.23
Horse	29	4101	157796	1.83	0.23
Twisted rose	15	9000	227104	2.26	0.27
Skull	16	8821	225572	2.31	0.33
Vertebrae	36	14844	386847	4.22	0.24

Table 2. Summary of our model zoo statistics. We report the number of planar cross-sections (#slices), contour complexity (i.e., the total number of edges of the slices) (#edges), the number of points sampled on the last set, closest to the contour (#samples), training time (in hours), and time to extract the mesh after training (meshing time, in hours)

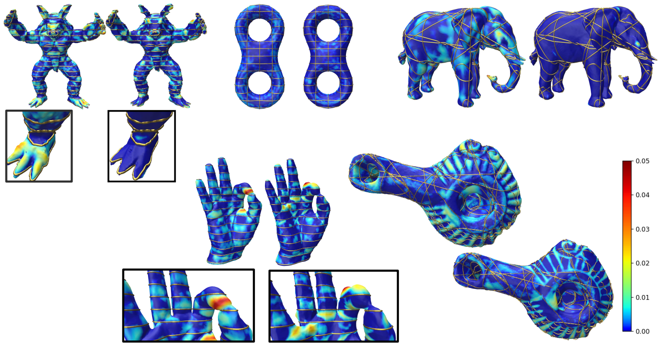

Figure 5. Closest-point distances from the ground truth (left meshes) to our reconstruction, and vice versa (right meshes). Scale is relative to the bounding-box diagonal.

References

[1] Amit Bermano, Amir Vaxman, and Craig Gotsman. Online reconstruction of 3d objects from arbitrary cross-sections. *ACM Transactions on Graphics (TOG)*, 30(5):1–11, 2011. 1, 2, 3

[2] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *arXiv preprint arXiv:2005.11084*, 2020. 2, 3

[3] Zhiyang Huang, Ming Zou, Nathan Carr, and Tao Ju. Topology-controlled reconstruction of multi-labelled domains from cross-sections. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 2

[4] Azimkhon Ostonov. Cut-and-approximate: 3d shape reconstruction from planar cross-sections with deep reinforcement learning. *arXiv preprint arXiv:2210.12509*, 2022. 2

[5] Zixiong Wang, Pengfei Wang, Qiujie Dong, Junjie Gao, Shuang-Min Chen, Shiqing Xin, and Changhe Tu. Neural-imls: Learning implicit moving least-squares for surface reconstruction from unoriented point clouds. *CoRR*, abs/2109.04398, 2021. 2

[6] Zixiong Wang, Pengfei Wang, Pengshuai Wang, Qiujie Dong, Junjie Gao, Shuangmin Chen, Shiqing Xin, Changhe Tu, and Wenping Wang. Neural-imls: Self-supervised implicit moving least-squares network for surface reconstruction, 2022. 2, 3

[7] Ming Zou, Michelle Holloway, Nathan Carr, and Tao Ju. Topology-constrained surface reconstruction from cross-sections. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015. 2