CrossSDF: 3D Reconstruction of Thin Structures From Cross-Sections

Supplementary Material

A. Additional Results

A.1. CT Scan Results

We present qualitative reconstructions of real anatomical structures with two vessel structures from the IRCADb-01 dataset [36] and two from the Medical Decathlon dataset [4], using pixel-level human-annotated segmentations. We evaluate *CrossSDF* on hepatic vessels from slices spaced 5mm apart, and liver vessels at 1mm apart. We display the results across different methods in Fig. A1 which demonstrates our model's ability to handle clinically relevant challenges. As no ground truth 3D geometry is available, we created training and test splits by withholding $\approx 10\%$ of the slices and reported the 2D intersection-over-union (IoU) on the test set. To do so, we use integer division to compute a slice skipping frequency to withhold cross-sections. For example, for the hepatic vessel of patient 31, we remove every $61//10 = 6^{th}$ cross-section to compute 2D IoU. The quantitative results can be found in Tab. 3.

Similarly to the synthetic scenes present in the main paper, OReX [33] generally results in overly smooth geometry. In some cases, such as the thin and highly-branched contours of Patient 33's Hepatic Vessel, OReX fails to converge to a good reconstruction. Screened Poisson [19] handles these cases more faithfully but is prone to breakages from the skipped slices, which is evident both qualitatively and quantitatively. Finally, we note CT scans with sparse slice thicknesses are particularly difficult to reconstruct, highlighting our model's robustness in such scenarios [1].

A.2. Additional Synthetic Results

Additional Comparisons. We display further qualitative comparisons in Figs. A2 and A4. We also report quantitative comparisons of 3D Volume IoU in Tab. A1. In Fig. A3 we also show results from Bermano *et al*. [9], a non-neural method that can handle arbitrary cross-section orientations, and works by a barycentric blending of indicator functions in the cells of the planar arrangements of the cross-sections. As can be seen in the figure, their approach is susceptible to laddering artifacts, and fails to run on more complex scenes due to poor scalability.

No 2D SDF Labels. Although we supervise our model with 2D SDF labels, our symmetric difference loss prevents the neural field from learning each 2D SDF beyond its interior/exterior classification. For this reason, the optimization could be driven using indicator function labels (*e.g.,* '1' for exterior and '-1' for interior). In Fig. A5 we present the result of doing so on the Balloon Dog scene (denoted as "No 2D SDF Labels"). We find that this leaves the predictable laddering artifacts along each contour. We attribute this to producing a discontinuous loss surface at the classification boundary. Conversely, our use of an L2 loss on 2D distance labels provides a smooth guide to this boundary.

Standard Hash-grid. In addition to the baselines reported in the main paper, we conduct a further ablation by removing all of the key components of our *CrossSDF* approach; *i.e.,* the adaptive encoding, Fourier features, and symmetric difference loss. We denote this as the 'Standard Hash-grid' baseline. The quantitative results for the Elephant scene are presented in Tab. A2. Removing these components leads to a degradation in performance, with the full model achieving the lowest CD and HD values.

A.3. Supplementary Video

In the video, available on our website, we provide comparison of a reconstruction from our approach compared to the baselines. We also visualize the optimization process on a different scene.

B. Implementation Details

We implemented our solution in PyTorch and used the accelerated tiny-cuda-nn [28] implementation for our hash-encoding module. All scenes were trained using a single 24GB NVIDIA RTX 4090. We train our network for a total of 500 epochs. The running time is commensurate with the complexity of the scene, taking as little as 20 minutes in the simplest cases (*e.g.,* Elephant in Fig. A4) to 5 hours in the most complex scenes (*e.g.,* Alveolis in Fig. A2). Our method does not have to create a (cubic-complexity) arrangement of cross-sectional planes, unlike [9, 46], and thus we scale better with the input complexity, only depending on the number of samples. Our network hyper-parameters are not tuned for each dataset. We use the same weights for losses and hash encoding resolution for all the datasets displayed. The mesh is extracted using marching cubes at 512^3 at the end of training (similarly for competing methods).

B.1. Model Architecture

In Tab. A3 we provide our hyper-parameters and network specifications. We use geometric initialization [43] to start the model as a sphere at the beginning of training. The neural networks are trained using the ADAM optimizer. The learning rate is initialized to 5×10^{-4}, and reduced by a factor of 0.9 every 10 epochs.

		Thin Structures											
	Method	Alveolis (100)		Cerebral (75)		Coronaries (75)		Coro (75)		Heart (100)		Pulmonary (75)	
		Align.	Non-Align.	Align.	Non-Align.	Align.	Non-Align.	Align.	Non-Align.	Align.	Non-Align.	Align.	Non-Align.
IoU ↑	Neural-IMLS [40]	0.050	0.048	0.44	0.045	0.029	0.040	0.71	0.74	0.033	0.021	0.056	0.045
	Screened Poisson [19]	0.84	0.72	0.85	0.79	0.78	0.68	0.85	0.89	0.28	**0.71**	0.89	0.86
	OReX [33]	0.37	0.12	0.64	0.33	0.51	0.63	0.96	**0.97**	0.32	0.42	0.59	0.74
	CrossSDF (Ours)	**0.87**	**0.76**	**0.94**	**0.90**	**0.88**	**0.78**	**0.99**	**0.97**	**0.77**	0.70	**0.96**	**0.94**

Table A1. Volume IoU comparison across thin structures.

Method	Elephant	
	CD (↓)	HD (↓)
CrossSDF (full model)	**0.58**	**7.1**
Standard Hash-grid	1.51	15.0

Table A2. Ablation of *CrossSDF* with all the components removed on the Elephant scene using aligned planes.

Hyper-parameter	**Value**
SDF Network Parameters	
M_{SDF} hidden layers	1
M_{SDF} hidden layer width	256
M_{hash} hidden layers	1
M_{hash} hidden layer width	128
M_{RFF} hidden layers	1
M_{RFF} hidden layer width	128
Constant for RFF scaling α	0.1
Softplus Activation β	100
Hash-Grid and Fourier Features	
Hash-grid levels	16
Minimum hash-grid resolution	2^5
Maximum hash-grid resolution	2^{10}
Hash-grid feature dimension	4
Hash dictionary size	2^{22}
Gaussian distribution variance for RFF	1.0
Sampling and Regularization	
Batch Size	2^{17}
Minimum Surface β	100
Threshold Samples Per Contour	50
λ_{eik} Eikonal reg. weight	1×10^{-3}
λ_{min} minimum surface reg. weight	5×10^{-2}
Weight decay weight	2×10^{-3}

Table A3. Hyperparameters and network specifications.

B.2. Data Pre-Processing

Contouring. For our synthetic dataset, we extract contours (as polylines) by taking the mesh and cutting plane parameters, and use this to compute a set of contours that results from cutting the mesh with the plane (*e.g.,* each vertex has two connecting edges). For medical CT scans, human-annotated segmentations come in pixelized binary mask format, with slices along one axis. For each mask we generate a contour by running marching squares at 512 resolution.

Sampling. For the planar sampling Ω_{pl}, we employ a combination of uniform sampling, on-contour sampling, and fixed-radius sampling, as described below:

- **On-Contour Sampling:** (Ω_{on}): We uniformly sample 25 points along each edge of the contour.
- **Fixed-Radius Sampling:** For each on-contour sample $\mathbf{x} \in \Omega_{\text{on}}$, we take two additional samples, each located a fixed distance ϵ from the edge in the perpendicular direction. One sample is positioned outward, perpendicular to the contour, while the other is positioned inward.
- **Uniform Sampling:** In each slicing plane, we perform uniform sampling, generating 10,000 samples per plane.
- **Adaptive Contour Sampling:** We sample a bounding box around each contour until at least 50 interior samples are gathered. This ensures thin structures with low cross-sectional area are captured.

Following a similar approach to OReX, we compute labels for the newly generated samples at predetermined intervals of 0, 50, 100, 200, and 300 epochs as part of a pre-processing step. During each re-sampling phase, the fixed radius ϵ is gradually reduced over time with values $\epsilon = 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}, 2^{-8}$. The label pre-processing step is efficient, taking less than three minutes for all tested scenes.

B.3. Baselines

For the point cloud reconstruction baseline methods, we use the on-contour samples Ω_{on} as a dense point cloud along each contour. For POCO [11], we use the model pre-trained on the ABC 10K dataset [21], which produced the best results relative to the alternative pre-trained models available.

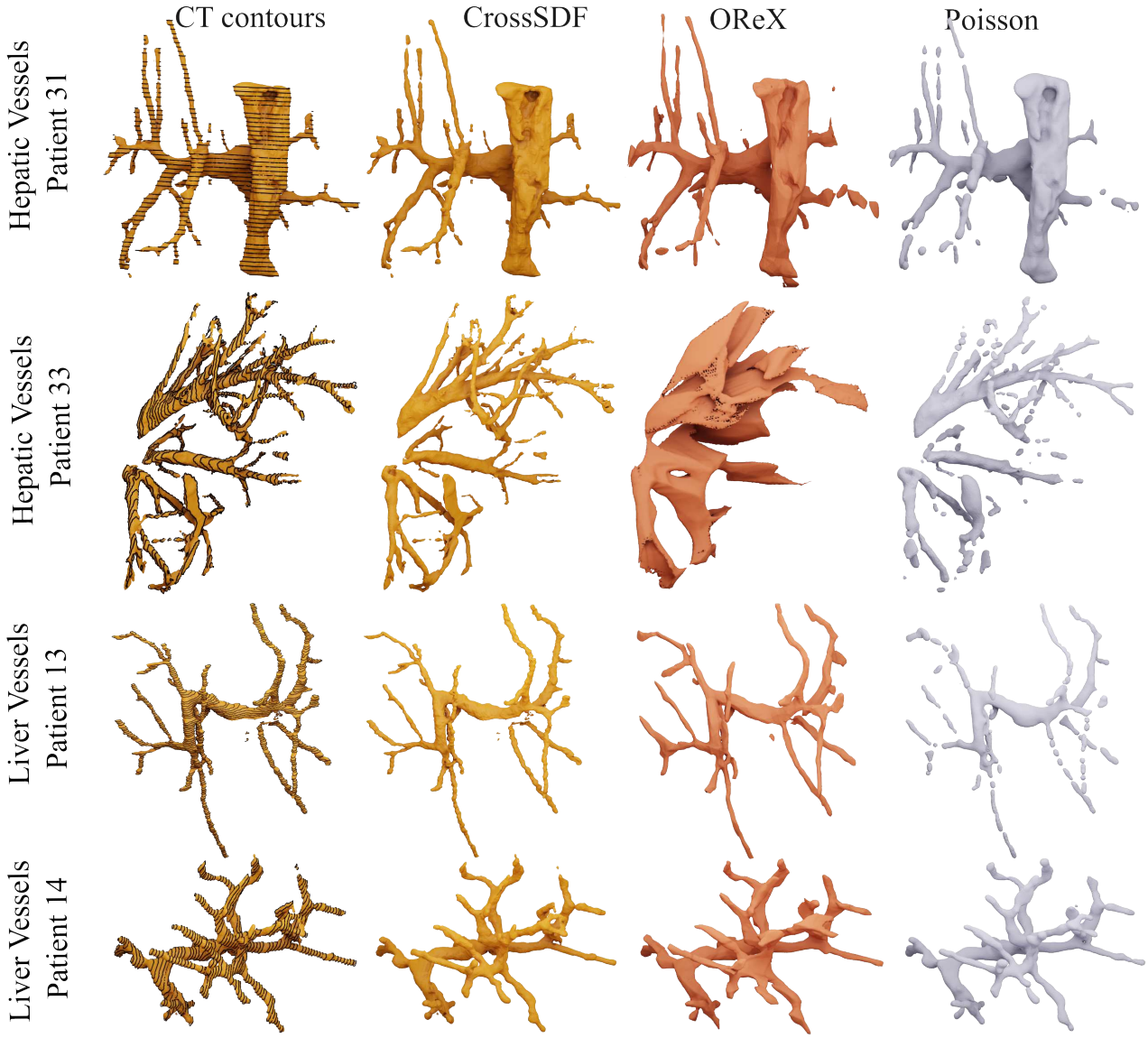

Figure A1. Additional qualitative results on real CT vessel structures. We compare our *CrossSDF* approach with existing methods. Note the 'CT contours' are overlayed on our result to improve viewing clarity.

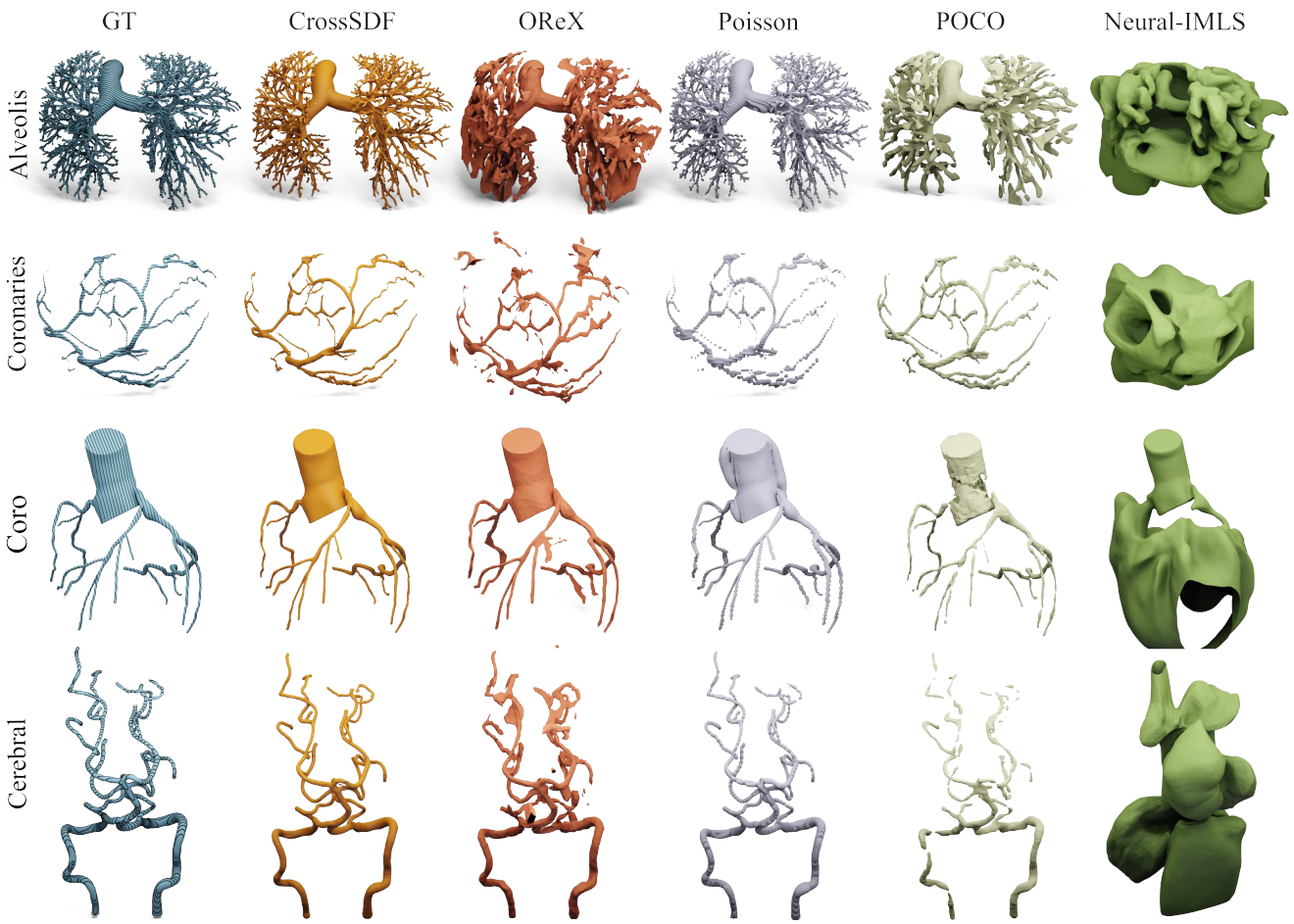

Figure A2. Additional qualitative results of thin synthetic data in the aligned setting. We compare our model with existing methods.

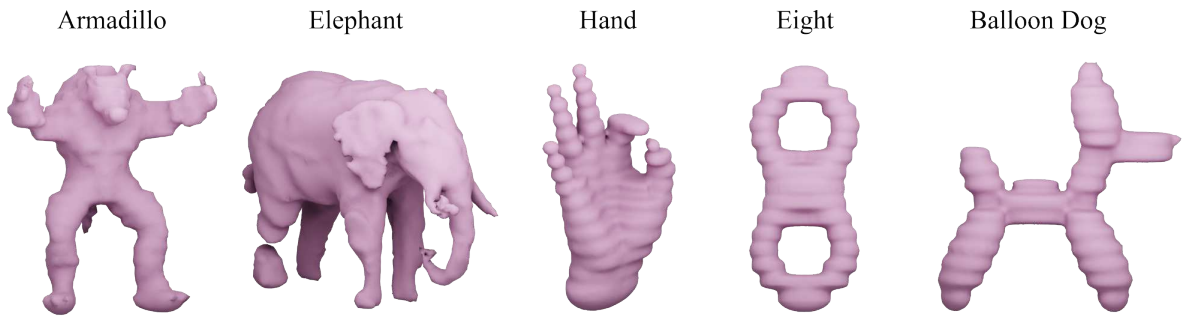

Figure A3. Additional qualitative results from the method outlined in Bermano *et al*. [9] using our synthetic dataset. Note the extreme laddering artifacts.

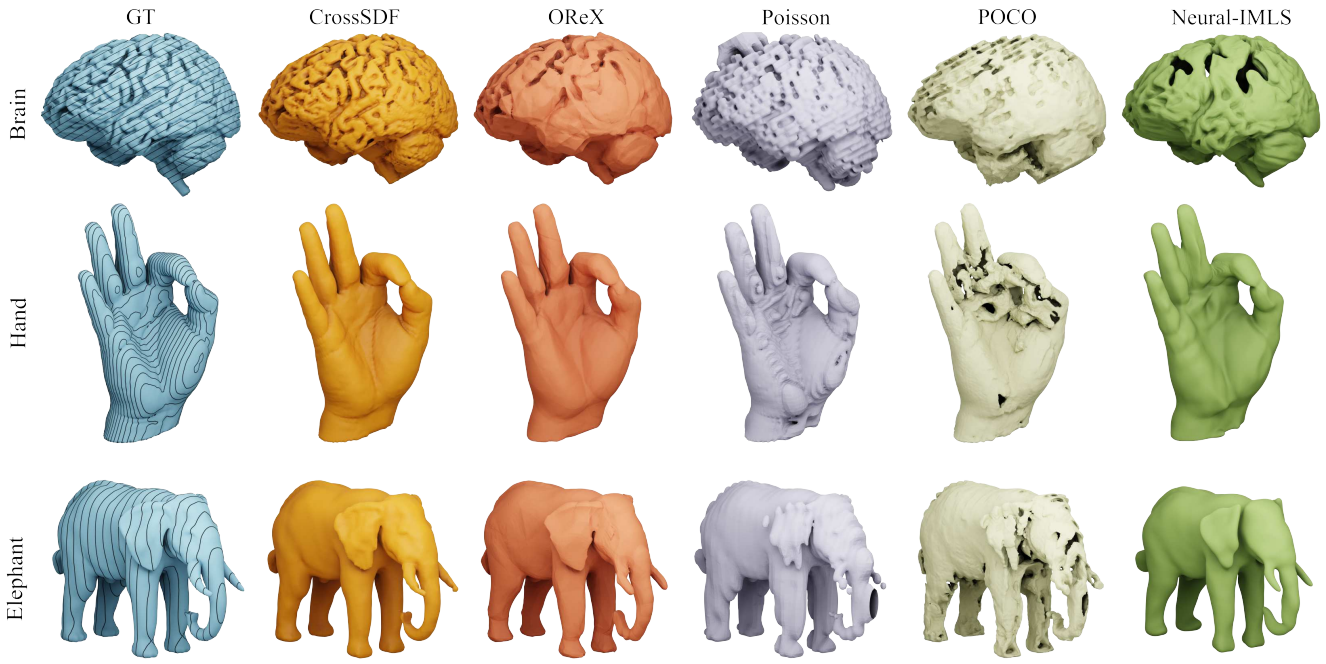

Figure A4. Additional qualitative results of different methods on the thick synthetic dataset, for aligned slices. We compare our model with existing methods.

Figure A5. Additional qualitative ablation results on the Balloon Dog scene.