Supplementary Material

VisCo Grids: Surface Reconstruction with Viscosity and Coarea Grids

A Computing $\nabla f(w_I)$

We will use the notation set in Section 3. For the losses in Eq. 3 in the main paper (the normal term) and Eq. 11 we require computing $\nabla f(w_I)$, where w_I is the center of the voxel of interest. For simplicity we will consider the voxel $[0, h]^3$. The 8 trilinear basis functions for this voxel are

$$\varphi_{abc}(x, y, z) = \frac{1}{h^3} \begin{Bmatrix} x & \text{if } a = 0 \\ h - x & \text{if } a = 1 \end{Bmatrix} \cdot \begin{Bmatrix} y & \text{if } b = 0 \\ h - y & \text{if } b = 1 \end{Bmatrix} \cdot \begin{Bmatrix} z & \text{if } c = 0 \\ h - z & \text{if } c = 1 \end{Bmatrix} \tag{1}$$

where the corners of the voxel are indexed by $a, b, c \in \{0, 1\}$. Given function values at these corner nodes, f_{abc}, the trilinear interpolant of f inside the voxel is

$$f_{abc}(x, y, z) = \sum_{a,b,c \in \{0,1\}} f_{abc} \varphi_{abc}(x, y, z). \tag{2}$$

Taking the gradient of this interpolant we get

$$\nabla f_{abc}(x, y, z) = \sum_{a,b,c \in \{0,1\}} f_{abc} \nabla \varphi_{abc}(x, y, z) \tag{3}$$

and for the center voxel point, $(x, y, z) = \frac{1}{2}(h, h, h)$, we have

$$\nabla \varphi_{abc} \left(\frac{h}{2}, \frac{h}{2}, \frac{h}{2} \right) = \frac{1}{4h} \left[(-1)^a, (-1)^b, (-1)^c \right]. \tag{4}$$

Similarly we can compute the gradient at an arbitrary point (x, y, z) inside a voxel.

B Implementation Details

For all experiments we follow a coarse-to-fine approach. We start optimizing at a $64 \times 64 \times 64$ grid resolution, then scale up to $128 \times 128 \times 128$ and finish at $256 \times 256 \times 256$. At each scale up we initialize the higher resolution grid values, f_I, by using trilinear interpolation within the voxels of the coarser grid. After each up-sampling, we prune grid voxels by removing those with an SDF value higher/lower than threshold $\pm t$, where $t \in \{0.4, 0.9\}$ enabling faster training and lower memory consumption; this is especially useful at the highest resolution grid. For 30 and 21 minutes running time budgets we prune with $t = 0.9$, and for 15 and 8 minutes with $t = 0.4$. For 30 minutes budget we perform 5 epochs at 64 resolution, 5 epochs at 128 and 3 at 256. For the rest of running time budgets we perform 2 epochs at each resolution, except for the 8 minutes budget, where at 256 resolution we perform only 1 epoch. Each epoch consists of 12800 iterations. At each training iteration the batch is composed by sampling random 10% of the *active* voxels (those which are left after pruning).

For all the final experiments we set $\lambda_p = 0.1$, $\lambda_n = 10^{-5}$, $\lambda_v = 1e-4$, $\lambda_c = 1e-6$, $\epsilon = 1e-2$. As for the optimizer, we use Adam [1] with a constant learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. All models are trained with a single NVIDIA Quadro GP-100 GPU.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

		Anchor	Daratech	DC	Gargoyle	Lord Quas	Mean
30 min	d_C	**0.21**	**0.26**	**0.15**	**0.17**	**0.12**	**0.19**
	d_H	3.00	4.06	2.22	4.40	1.06	2.95
	$d_{\overrightarrow{C}}$	0.15	0.14	0.09	0.11	0.07	0.11
	$d_{\overrightarrow{H}}$	1.07	1.76	2.76	0.95	0.64	1.44
21 min	d_C	0.27	0.27	0.16	**0.17**	**0.12**	0.20
	d_H	5.60	4.06	2.13	4.33	0.99	3.42
	$d_{\overrightarrow{C}}$	0.14	0.14	0.09	0.11	0.07	0.12
	$d_{\overrightarrow{H}}$	1.17	1.77	2.77	0.93	0.64	1.45
15 min	d_C	0.27	**0.26**	**0.15**	**0.17**	**0.12**	0.20
	d_H	5.68	4.20	2.23	4.45	1.10	3.53
	$d_{\overrightarrow{C}}$	0.14	0.14	0.09	0.11	0.07	0.11
	$d_{\overrightarrow{H}}$	1.10	1.77	2.80	1.04	0.66	1.47
8 min	d_C	0.28	**0.26**	**0.15**	**0.17**	0.13	0.20
	d_H	5.69	4.15	2.23	4.45	1.14	3.53
	$d_{\overrightarrow{C}}$	0.15	0.13	0.09	0.11	0.07	0.11
	$d_{\overrightarrow{H}}$	1.15	1.78	2.78	0.98	0.68	1.48

Table I: Quantitative results of VisCo for different training time budgets on the surface reconstruction benchmark [2]. Reducing the running time from 30 mins to 8 mins only marginally reduces reconstruction metrics.

We did a grid search for all the hyper-parameters in the range of 10^{-6} to 10^{-1} with multiplicative steps of 10^{-1}. We observed minimal performance difference. For all benchmark datasets we use the exact same hyper-parameters. More specifically, for the two proposed new losses – Viscosity and Coarea – we observe no performance change in the ranges $[5e-3, 5e-2]$ and $[5e-7, 5e-6]$, respectively, which allows for consistency across scenes with fixed hyper-parameters (see Fig. 7 and 8).

We will publish the source code which reproduces the experimental results upon the paper acceptance.

C Training time

In this section, we present qualitative and quantitative results of VisCo for different running time budgets. We experiment with the running time versus reconstruction quality trade-offs and show that short time training produces comparable reconstruction quality to longer time training. In Tab. 1 we show quantitative results and in Fig. 1 qualitative results. Note that reducing the running time from 30 mins to 8 mins only marginally reduces reconstruction metrics, while qualitatively produces indistinguishable reconstruction results. The different running times versions were created mostly by reducing the number of epochs per resolution from 5 down to 2 (see more details in Sec. B). We strongly believe that further significant speedups are possible with a more efficient implementation.

Below we report average time and memory footprint required for a single training iteration on NVIDIA Quadro GP100 GPU. Because of the pruning applied to the grid, we need to learn only a sparse set of the grid values (we call them *active*).

- 64^3 resolution (57% of the grid values are active): 2.3 msec, 975MB VRAM
- 128^3 resolution (31% of the grid values are active): 8.6 msec, 1070MB VRAM
- 256^3 resolution (30% of the grid values are active): 25.8 msec, 1650MB VRAM

For neural networks (INRs) every point evaluation requires forward and backward in a network involving all network's parameters in general, while for a grid we only require nearby grid function values (learnable parameters). Typical iteration times for NN (taken from DiGS) are:

- 66.5K params: 5.2-12.0 msec
- 2.1M params: 17.5 msec

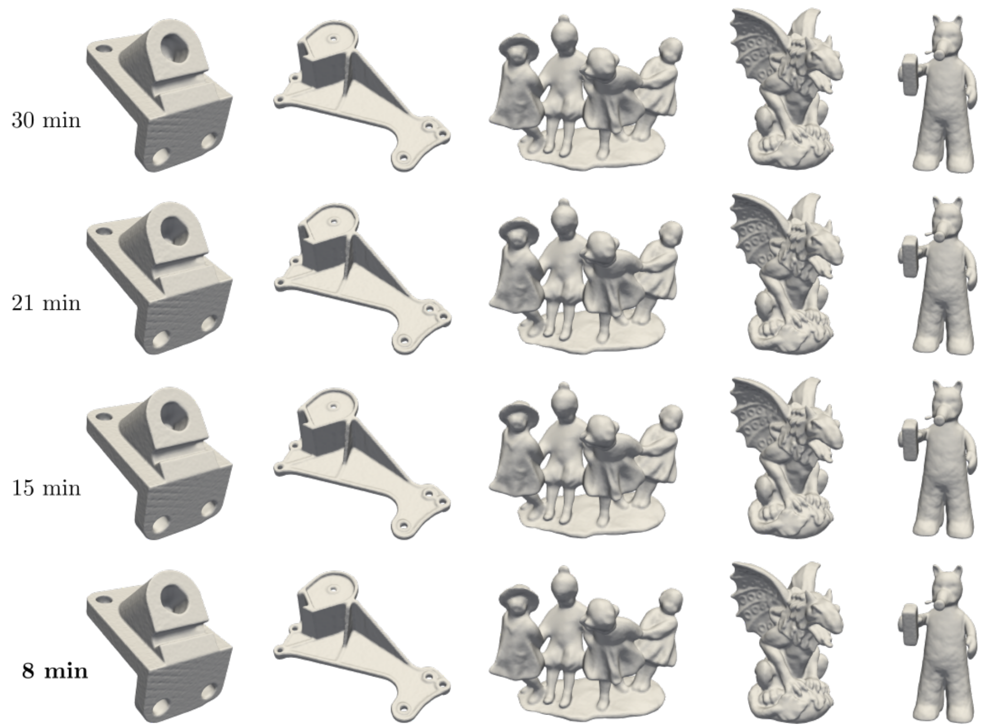

Figure I: Qualitative results of VisCo for different training time budgets on the surface reconstruction benchmark [2]. Note that reduction of the training time does not result in inferior reconstruction. The models trained for 30 mins and 8 mins produce indistinguishable reconstruction results.

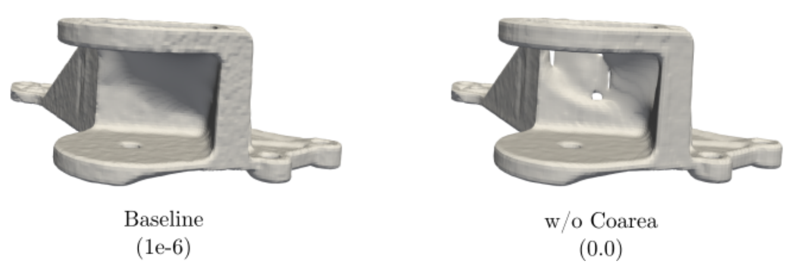

Figure II: Visual comparison for Daratech between all loses vs. w/o Coarea. Reconstructed meshes from Tab. 3. Note small holes when removing the Coarea loss.

D Daratech Coarea Effect

In this section we further study why in Tab. 3, Daratesh seem to have better reconstruction w/o Coarea loss. Visual inspection reveals higher qualitative result for the mesh reconstructed with the Coarea loss although it has a higher quantiative error. We observe small holes when removing the Coarea loss, see Fig. II.

References

[1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[2] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019.