Iso-Points: Optimizing Neural Implicit Surfaces with Hybrid Representations Supplemental Material

Wang Yifan[1] Shihao Wu[1] Cengiz Öztireli[2] Olga Sorkine-Hornung[1]

[1]ETH Zurich [2]University of Cambridge

We provide more technical details and experimental results for our iso-points representation. Upon acception we will also release our code and data.

1. Details for MVS

1.1. Cause of inner structures.

State-of-the-art implicit differentiable renderers, *e.g.* IDR [2] and DVR [1], are typically supervised with two losses requiring three types of samples. The first loss, appearance loss, aims at optimizing the surface texture and the appearance-dependent geometry. Naturally, supervising this loss requires *on-surface* samples, which are points whose SDF prediction is 0. The second loss, silhouette loss, is responsible for optimizing the appearance-independent geometry. Depending on sign of the predicted SDF value, a point is classified to be inside (SDF < 0) or outside (SDF > 0) the shape. This classification must match the 2D silhouette of the shape in the input images. For this purpose, two types of samples are drawn: *out-surface* samples and *in-surface* samples. The out-surface samples are drawn on camera rays outside the object silhouette, which are penalized if their SDF prediction is negative; vice versa for in-surface samples. This supervision is inherently noisy, due to depth-ambiguity for drawing the in-surface points. Indeed, as illustrated in Figure 6 of the main paper, for a camera ray passing through a pixel inside the 2D silhouette, although all the points along this ray are classified as inside the silhouette mask (and thus penalized for positive SDF predictions), their true label actually depends on the depth. While this issue is mitigated by multi-view supervision, it leads to incorrect inner structures as shown in Figure 8 of the main paper.

1.2. Sampling with iso-points.

We sample the three types of samples with the help of iso-points. The utility of iso-points increases sampling speed thanks to advantageous computation complexity compared to ray-tracing, but also improves sample accuracy for the in-surface samples.

In particular, given a set of uniform iso-points extracted by methods described in Section 3.1 of the main paper, we first filter the occluded points using the point rasterizer Differential Surface Splatter (DSS) [3]. From the remaining points we subsample or upsample to obtain a subset of N points and perturb with white Gaussian noise ($\sigma = 0.05$). These points are then projected onto the current iso-surface using Newton-method using Eq.1 in the main paper. The resulting projected points constitute to the on-surface samples for the appearance loss.

For the sampling of out-and-in-surface points, out goal is to determine a tighter depth range using iso-points. We do so by approximating the first and last intersections of a camera ray and the iso-surface. To this end, for each camera position, we use DSS to divide the iso-points to visible and occluded subset, denoted as $\mathcal{P}_f$ and $\mathcal{P}_b$. Then, same as DVR and IDR, we randomly sample camera rays by shooting from the camera center through uniformly sampled pixels on the input images. Denoting the camera center as $\mathbf{c}$ and the normalized camera ray as $\mathbf{r}$, we can compute the distance to the camera ray from an arbitrary 3D point $\mathbf{p}$ as

$$r(\mathbf{p}) = \|\mathbf{p} - \mathbf{c}\|^2 - ((\mathbf{p} - \mathbf{c})^T \mathbf{r}_0)^2. \tag{1}$$

Now, we can approximate the first and last intersections between a camera ray and the iso-surface, denoted as $\mathbf{u}$ and $\mathbf{v}$ respectively, with

$$\mathbf{u} = \arg\min_{\mathbf{p} \in \mathcal{P}_f} r(\mathbf{p}), \quad \mathbf{v} = \arg\min_{\mathbf{p} \in \mathcal{P}_b} r(\mathbf{p}). \tag{2}$$

As a result, the tight sample range is

$$\left[(\mathbf{u} - \mathbf{c})^T \mathbf{r}_0, (\mathbf{v} - \mathbf{c})^T \mathbf{r}_0 \right]. \tag{3}$$

Given this tighter depth range, we can proceed to sample the out-and-in-surface points following the practice proposed in IDR. Namely, for each camera ray, we sample and evaluate the SDF value of $T = 32$ equidistant points on the segment between $\mathbf{u}$ and $\mathbf{v}$, and pick the one with the smallest SDF value. If the camera ray shoots through the inside of the object's 2D silhouette, the picked point is an in-surface sample, otherwise we obtain an out-surface sample.

In practice, we only need to apply this sampling strategy to in-surface samples. For out-surface points, we compute

the intersections between the camera ray and the unit-box centered around the object, and randomly sample a point between the two intersections.

2. Ablation study for iso-points regularizations

In this section, we conduct an ablation study in support of our iso-points regularization experiments (Section 4.2 of the main paper) to inspect the effect of each individual regularization technique.

Recall our optimization objective:

$$\mathcal{L} = \gamma_{\text{onSDF}}(\mathcal{L}_{\text{onSDF}} + \mathcal{L}_{\text{isoSDF}}) + \gamma_{\text{normal}}(\mathcal{L}_{\text{normal}} + \mathcal{L}_{\text{isoNormal}}) + \gamma_{\text{offSDF}}\mathcal{L}_{\text{offSDF}} + \gamma_{\text{eikonal}}\mathcal{L}_{\text{eikonal}}, \quad (4)$$

where the loss terms with on-surface points are weighted using iso-points to reduce the impact of noisy data, *i.e.*

$$\mathcal{L}_{\text{onSDF}} = \frac{1}{|\mathcal{Q}_s|}\sum_{\mathbf{q}_s \in \mathcal{Q}_s} v(\mathbf{q}_s)|f(\mathbf{q}_s)| \quad (5)$$

$$\mathcal{L}_{\text{normal}} = \frac{1}{|\mathcal{Q}_s|}\sum_{\mathbf{q}_s \in \mathcal{Q}_s} v(\mathbf{q}_s)|1 - \cos(J_f^{\mathsf{T}}(\mathbf{q}_s), \mathbf{n}_s)|. \quad (6)$$

In the ablation study, we drop the outlier weights $v(\mathbf{q}_s)$, $\mathcal{L}_{\text{isoSDF}}$ and $\mathcal{L}_{\text{isoNormal}}$ from the optimization objective in turn. Additionally, we disable the periodic update to demonstrate the necessity of having dynamic iso-points.

As shown in Figure 1, when optimizing without the outlier weights $v(\mathbf{q}_s)$, bulges appear close to the surface, as the neural implicit function overfits to severe noise in the input data, similar to the baseline without any regulation. When $\mathcal{L}_{\text{isoNormal}}$ is removed, the resulting surface is considerably less smooth. Similar but less prominent effect can be observed when $\mathcal{L}_{\text{isoSDF}}$ is removed. When optimizing with static iso-points extracted from the early training stage, the result becomes over-smoothed and contains artifact, as the premature iso-points does not capture the fine-scale structure sufficiently. Finally, combining all regularizations achieves a significantly better balance between sharpness and smoothness.

We also evaluate the reconstructions quantitatively using the DTU evaluation protocol as in the main paper, as shown in Table 1. The computed score, L1-Chamfer, aligns with the visual quality shown in Figure 1, especially sensitive to low-frequency errors (see *no regularizations* vs *periodic update*). It's worth noting that high-frequency errors seems to be underestimated by L1-Chamfer in all the cases (*e.g.* *without* $\mathcal{L}_{\text{isoNormal}}$ vs *without* $\mathcal{L}_{\text{isoSDF}}$).

3. Performance Analysis

In this section, we report more a detailed performance analysis of our implementation.

We benchmark the individual steps during iso-points extraction, namely projection, resampling, and upsampling. For clarity, we summarize the pipeline introduced in Section 3 of the main paper: First, a maximum of 10 Newton iterations (Eq.1 in the main paper) are performed in the projection step with the early termination threshold ϵ for set to $5 \cdot 10^{-5}$. In the resampling step, we run 5 iterations of repulsion (Eq.2 in the main paper). The resulting point set is re-projected to the iso-surface via the Newton method as before, and points that do not satisfy the termination condition ($|f(\mathbf{p})| >= \epsilon$) are removed. This filtered point set is upsampled to the original density, before a final projection step. All aforementioned steps are implemented on GPU.

	Chamfer Distance
no regularizations	0.69
full regularization	0.56
without $v(\mathbf{q}_s)$	0.62
without $\mathcal{L}_{\text{isoSDF}}$	0.59
without $\mathcal{L}_{\text{isoNormal}}$	0.58
without periodic update	1.02

Table 1: Ablation study for iso-points regularizations used in Section 4.2 of the main paper.

The results shown in Table 2 is conducted on a NVIDIA GTX 1080 Ti using $22,000$ iso-points and a 3-layer SIREN model with 256 channels in the hidden layers. As a reference, we also report the runtime of a forward and backward pass with $22,000$ points on the same model.

	time (s)	
total	0.080	
projection	0.038	47.32%
re-sampling	0.012	14.50%
upsampling	0.029	37.49%
reference	0.009	

Table 2: Benchmarking the individual steps of iso-points extraction.

When optimizing implicit surfaces from image inputs, sampling with iso-points instead of ray-tracing can contribute to faster training. As mentioned in Section 4.1 of the main paper, since only the visible iso-points are included in the loss function, we can limit the computation of on-surface samples by filtering the iso-points by visibility, prior to the projection step. The visibility detection can be efficiently implemented using a point rasterizer. Note that since the iso-surface only changes locally and gradually during training, we do not need to extract the iso-points (including projection, resampling, and upsampling) in every training iteration.

In Table 3, we report the runtime of one training iteration (including sampling, forward and backward passes) with iso-points sampling and ray-tracing sampling. The networks used in this experiment are identical to IDR [2]. Both sampling methods yield approximately 2048 training points.

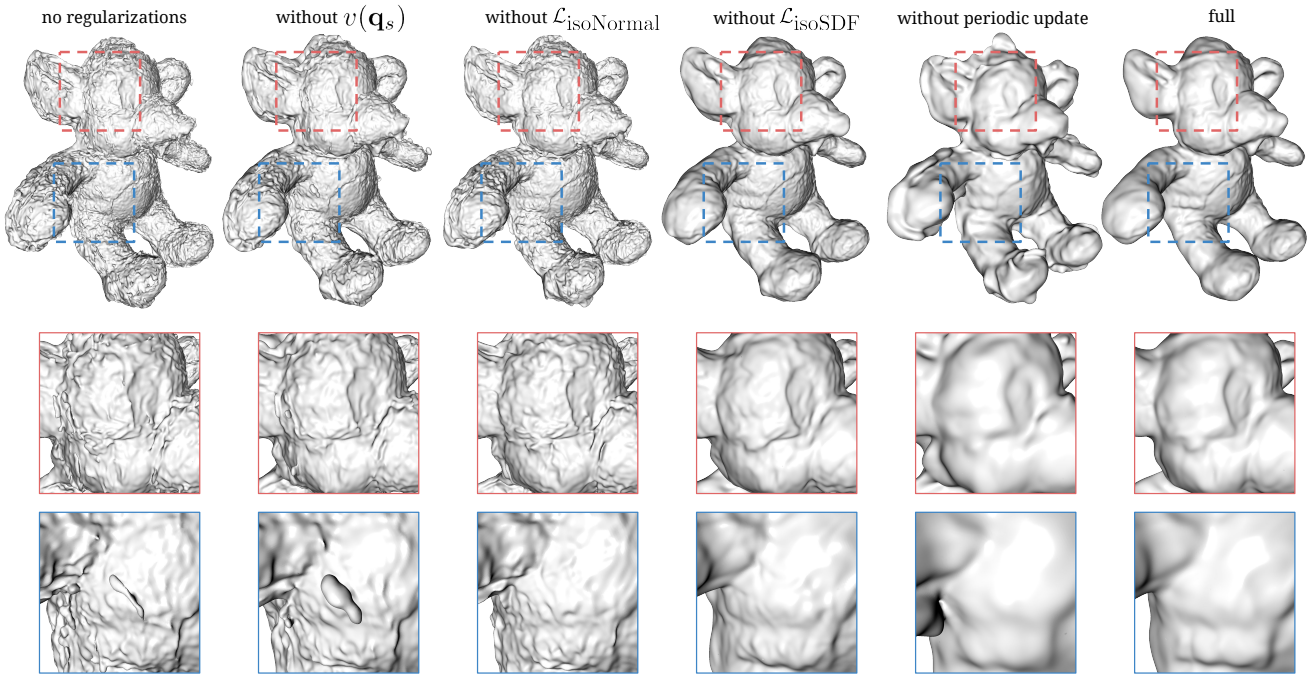

Figure 1: Ablation study of iso-points regularizations.

	ray-tracing	iso-points
time (s)	0.5676	0.3481

Table 3: Runtime comparison of two different sampling strategies in multi-view reconstruction

References

[1] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 3504–3515, 2020. 1

[2] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Proc. IEEE Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2

[3] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 38(6):1–14, 2019. 1