Iso-Points: Optimizing Neural Implicit Surfaces with Hybrid Representations

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

[1]ETH Zurich   [2]University of Cambridge

Abstract

Neural implicit functions have emerged as a powerful representation for surfaces in 3D. Such a function can encode a high quality surface with intricate details into the parameters of a deep neural network. However, optimizing for the parameters for accurate and robust reconstructions remains a challenge, especially when the input data is noisy or incomplete. In this work, we develop a hybrid neural surface representation that allows us to impose geometry-aware sampling and regularization, which significantly improves the fidelity of reconstructions. We propose to use iso-points as an explicit representation for a neural implicit function. These points are computed and updated on-the-fly during training to capture important geometric features and impose geometric constraints on the optimization. We demonstrate that our method can be adopted to improve state-of-the-art techniques for reconstructing neural implicit surfaces from multi-view images or point clouds. Quantitative and qualitative evaluations show that, compared with existing sampling and optimization methods, our approach allows faster convergence, better generalization, and accurate recovery of details and topology.

1. Introduction

Reconstructing surfaces from real-world observations is a long-standing task in computer vision. Recently, representing 3D geometry as a neural implicit function has received much attention [40, 36, 7]. Compared with other 3D shape representations, such as point clouds [29, 20, 54], polygons [24, 31, 39, 32], and voxels [8, 47, 35, 22], it provides a versatile representation with infinite resolution and unrestricted topology.

Fitting a neural implicit surface to input observations is an optimization problem. Some common application examples include surface reconstruction from point clouds and multi-view reconstruction from images. For most cases, the observations are noisy and incomplete. This leads to fundamental geometric and topological problems in the final reconstructed surface, as the network overfits to the imperfect data. We observe that this problem remains, and can become more prominent with the recent powerful architec-
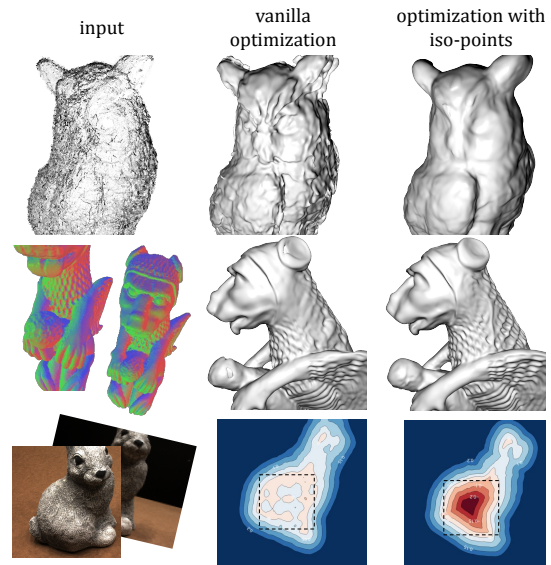

Figure 1: We propose a hybrid neural surface representation with implicit functions and *iso-points*, wich leads to accurate and robust surface reconstruction from imperfect data. The iso-points allows us to augment existing optimization pipelines in a variety of ways: geometry-aware regularizers are incorporated to reconstruct a surface from a noisy point cloud (first row); geometric details are preserved in multi-view reconstruction via feature-aware sampling (second row); iso-points can serve as a 3D prior to improve the topological accuracy of the reconstructed surface (third row). The input data are respectively: reconstructed point cloud [12] of model 122 of DTU-MVS dataset [21], multi-view rendered images of DOG-WINGED model from Sketchfab dataset [55] and multi-view images of model 55 of DTU-MVS dataset

tures, *e.g*. sine activations [45] and Fourier features [41].

We show examples of problems in fitting neural implicit functions in Figure 1. When fitting a neural surface to a noisy point cloud, "droplets" and bumps emerge where there are outlier points and holes (first row); when fitting a surface to image observations, fine-grained geometric features are not captured due to under-sampling (second row); topological noise is introduced when inadequate views are available for reconstruction (third row).

In this work, we propose to alleviate these problems by introducing a *hybrid neural surface representation* using *iso-points*. The technique converts from an implicit neural surface to an explicit one via sampling iso-points, and goes back to the implicit representation via optimization. The

two-way conversion is performed on-the-fly during training to introduce geometry-aware regularization and optimization. This approach unlocks a large set of fundamental tools from geometry processing to be incorporated for accurate and robust fitting of neural surfaces.

A key challenge is to extract the iso-points on-the-fly efficiently and flexibly during the training of a neural surface. Extending several techniques from point-based geometry processing, we propose a multi-stage strategy, consisting of projection, resampling, and upsampling. We first obtain a sparse point cloud on the implicit surface via projection, then resample the iso-points to fix severely under-sampled regions, and finally upsample to obtain a dense point cloud that covers the surface. As all operations are GPU friendly and the resampling and upsampling steps require only local point distributions, the entire procedure is fast and practical for training.

We illustrate the utility of the new representation with a variety of applications, such as multi-view reconstruction and surface reconstruction from noisy point clouds. Quantitative and qualitative evaluations show that our approach allows for fast convergence, robust optimization, and accurate reconstruction of details and topology.

2. Related work

We begin by discussing existing representations of implicit surfaces, move on to the associated optimization and differentiable rendering techniques given imperfect input observations, and finally review methods proposed to sample points on implicit surfaces.

Implicit surface representations. Implicit functions are a flexible representation for surfaces in 3D. Traditionally, implicit surfaces are represented globally or locally with radial basis functions (RBF) [6], moving least squares (MLS) [28], volumetric representation over uniform grids [9], or adaptive octrees [25]. Recent works investigate neural implicit surface representations, i.e., using deep neural networks to encode implicit function [40, 45], which achieves promising results in reconstructing surfaces from 3D point clouds [3, 44, 11] or images [30, 52, 38].

Compared with simple polynomial or Gaussian kernels, implicit functions defined by nested activation functions, e.g., MLPs [7] or SIREN [45], have more capability in representing complex structures. However, the fiting of such neural implicit function requires clean supervision points [51] and careful optimization to prevent either overfitting to noise or underfitting to details and structure.

Optimizing neural implicit surfaces with partial observations. Given raw 3D data, Atzmon et al. [3, 4] use sign agnostic regression to learn neural implicit surfaces without using a ground truth implicit function for supervision. Gropp et al. [14] use the Eikonal term for implicit geometric regularization and provide a theoretical analysis of the plane reproduction property possessed by the neural zero level set surfaces. Erler et al. [11] propose a patch-based framework that learns both the local geometry and the global inside/outside information, which outperforms existing data-driven methods. None of these methods exploit an explicit sampling of the implicit function to improve the optimization. Poursaeed et al. [42] use two different encoder-decoder networks to simultaneously predict both an explicit atlas [15] and an implicit function. In contrast, we propose a hybrid representation using a single network.

When the input observations are in the form of 2D images, differentiable rendering allows us to use 2D pixels to supervise the learning of 3D implicit surfaces through automatic differentiation and approximate gradients [23, 46]. The main challenge is to render the implicit surface and compute reliable gradients at every optimization step efficiently. Liu et al. [34] accelerate the ray tracing process via a coarse-to-fine sphere tracing algorithm [16], and use an approximate gradient in back propagation. In [33], a ray-based field probing and an importance sampling technique are proposed for efficient sampling of the object space. Although these methods greatly improve rendering efficiency, the sampling of ray-based algorithms, i.e., the intersection between the ray and the iso-surface, are intrinsically irregular and inefficient. Most of the above differentiable renderers use ray casting to generate the supervision points. We propose another type of supervision points by sampling the implicit surface in-place.

Sampling implicit surfaces. In 1992, Figueiredo et al. [10] proposed a powerful way to sample implicit surfaces using dynamic particle systems that include attraction and repulsion forces. Witkin and Heckbert [50] further developed this concept by formulating an adaptive repulsion force. While the physical relaxation process is expensive, better initialization techniques have been proposed, such as using seed flooding on the partitioned space [27] or the octree cells [43]. Huang et al. [19] resample point set surfaces to preserve sharp features by pushing points away from sharp edges before upsampling. When sampling a neural implicit surface, existing works such as Atzmon et al. [2] project randomly generated 3D points onto the iso-surface along with the gradient of the neural level set. However, such sampled points are unevenly distributed, and may leave parts of the surface under-sampled or over-sampled.

3. Method

Given a neural implicit function $f_t(\mathbf{p}; \theta_t)$ representing the surface $\mathcal{S}_t$, where θ_t are the network parameters at the t-th training iteration, our goal is to efficiently generate and utilize a dense and uniformly distributed point set on the zero level set, called *iso-points*, which faithfully represents the current implicit surface $\mathcal{S}_t$. Intuitively, we can deploy

the iso-points back into the ongoing optimization to serve various purposes, *e.g.* improving the sampling of training data and providing regularization for the optimization, leading to a substantial improvement in the convergence rate and the final optimization quality.

In this section, we first focus on how to extract the iso-points via projection and uniform resampling. We then explain how to utilize the iso-points for better optimization in practical scenarios.

3.1. Iso-surface sampling

As shown in Figure 2, our iso-surface sampling consists of three stages. First, we project a set of initial points $\mathcal{Q}_t$ onto the zero level set to get a set of *base iso-points* $\tilde{\mathcal{Q}}_t$. We then resample $\tilde{\mathcal{Q}}_t$ to avoid clusters of points and fill large holes. Finally, we upsample the points to obtain dense and uniformly distributed iso-points $\mathcal{P}_t$.

Projection. Projecting a point onto the iso-surface can be seen as using Newton's method [5] to approximate the root of the function starting from a given point. Atzmon *et al.* [2] derive the projection formula for functions modeled by generic networks. For completeness, we recap the steps here, focusing on $f : \mathbb{R}^3 \rightarrow \mathbb{R}$.

Given an implicit function $f(\mathbf{p})$ representing a signed distance field and an initial point $\mathbf{q}_0 \in \mathbb{R}^3$, we can find an iso-point $\mathbf{p}$ on the zero level set of f using Newton iterations: $\mathbf{q}_{k+1} = \mathbf{q}_k - J_f(\mathbf{q}_k)^+ f(\mathbf{q}_k)$, where J_f^+ is the Moore-Penrose pseudoinverse of the Jacobian. In our case, J_f is a row 3-vector, so that $J_f^+(\mathbf{q}_k) = \frac{J_f^\mathsf{T}(\mathbf{q}_k)}{\|J_f(\mathbf{q}_k)\|^2}$, where the Jacobian $J_f(\mathbf{q}_k)$ can be conveniently evaluated in the network via backpropagation.

However, for some contemporary network designs, such as sine activation functions [44] and positional encoding [37], the signed distance field can be very noisy and the gradient highly non-smooth. Directly applying Newton's method then causes overshooting and oscillation. While one could attempt more sophisticated line search algorithms, we instead address this issue with a simple clipping operation to bound the length of the update, *i.e.*

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \tau\left(\frac{J_f^\mathsf{T}(\mathbf{q}_k)}{\|J_f(\mathbf{q}_k)\|^2} f(\mathbf{q}_k)\right), \tag{1}$$

where $\tau(\mathbf{v}) = \frac{\mathbf{v}}{\|\mathbf{v}\|}\min(\|\mathbf{v}\|, \tau_0)$. We set $\tau_0 = \frac{D}{2|\mathcal{Q}_t|}$ with D denoting the diagonal length of the shape's bounding box.

In practice, we initialize $\mathcal{Q}_t$ with randomly sampled points at the beginning of the training and then with iso-points $\mathcal{P}_{t-1}$ from the previous training iteration. Similar to [2], at each training iteration, we perform a maximum of 10 Newton iterations and terminate as soon as all points have converged, *i.e.* $|f(\mathbf{q}_k)| < \epsilon, \forall \mathbf{q} \in \mathcal{Q}_t$. The termination threshold ϵ is set to 10^{-4} and gradually reduced to 10^{-5} during training.
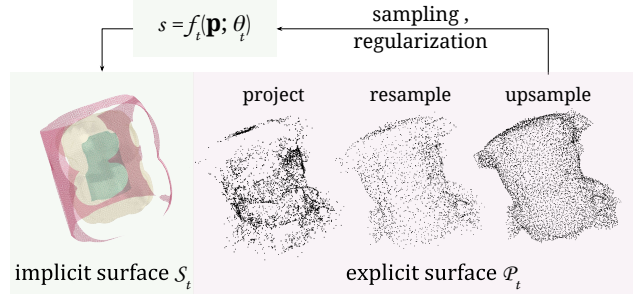

Figure 2: Overview of our hybrid representation. We efficiently extract a dense, uniformly distributed set of *iso-points* as an explicit representation for a neural implicit surface. Since the extraction is fast, iso-points can be integrated back into the optimization as a 3D geometric prior, enhancing the optimization.

Uniform resampling. The projected base iso-points $\tilde{\mathcal{Q}}_t$ can be sparse and hole-ridden due to the irregularity present in the neural distance field, as shown in Figure 2. Such irregular sample distribution prohibits us from many downstream applications described later.

The resampling step aims at avoiding over- and undersampling by iteratively moving the base iso-points away from high-density regions, *i.e.*

$$\tilde{\mathbf{q}} \leftarrow \tilde{\mathbf{q}} - \alpha\mathbf{r}, \tag{2}$$

where $\alpha = \sqrt{D/|\tilde{\mathcal{Q}}_t|}$ is the step size. The update direction $\mathbf{r}$ is a weighted average of the normalized translations between $\tilde{\mathbf{q}}$ and its K-nearest points (we set $K = 8$):

$$\mathbf{r} = \sum_{\tilde{\mathbf{q}}_i \in \mathcal{N}(\tilde{\mathbf{q}})} w(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}})\frac{\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}}{\|\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}\|}. \tag{3}$$

The weighting function is designed to gradually reduce the influence of faraway neighbors, specifically

$$w(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}) = e^{-\frac{\|\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}\|^2}{\sigma_p}}, \tag{4}$$

where the density bandwidth σ_p is set to be $16D/|\tilde{\mathcal{Q}}_t|$.

Upsampling. Next, we upsample the point set to the desired density while further improving the point distribution. Our upsampling step is based on edge-aware resampling (EAR) proposed by Huang *et al.* [19]. We explain the key steps and our main differences to EAR as follow.

First, we compute the normals as the normalized Jacobians and apply bilateral normal filtering, just as in EAR. Then, the points are pushed away from the edges to create a clear separation. We modify the original optimization formulation with a simpler update consisting of an attraction and a repulsion term. The former pulls points away from

the edge and the latter prevents the points from clustering.

$$\Delta\mathbf{p}_{\text{attraction}} = \frac{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \phi(\mathbf{n}_i, \mathbf{p}_i - \mathbf{p})(\mathbf{p} - \mathbf{p}_i)}{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \phi(\mathbf{n}_i, \mathbf{p}_i - \mathbf{p})}, \quad (5)$$

$$\Delta\mathbf{p}_{\text{repulsion}} = 0.5 \frac{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} w(\mathbf{p}_i, \mathbf{q})(\mathbf{p}_i - \mathbf{p})}{\sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} w(\mathbf{p}_i, \mathbf{q})}, \quad (6)$$

$$\mathbf{p} \leftarrow \mathbf{p} - \tau(\Delta\mathbf{p}_{\text{repulsion}}) - \tau(\Delta\mathbf{p}_{\text{attraction}}), \quad (7)$$

where $\phi(\mathbf{n}_i, \mathbf{p} - \mathbf{p}_i) = e^{-\frac{(\mathbf{n}_i^{\mathsf{T}}(\mathbf{p} - \mathbf{p}_i))^2}{\sigma_p}}$ is the anisotropic projection weight, $\mathbf{n}_i$ is the point normal of neighbor $\mathbf{p}_i$ and w is the spatial weight defined in (4). We use the same directional clipping function τ as before to bound the two update terms individually, which improves the stability of the algorithm for sparse point clouds.

By design, new points are inserted in areas with low density or high curvature. The trade-off is controlled by a unifying priority score $P(\mathbf{p}) = \max_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} B(\mathbf{p}, \mathbf{p}_i)$, where B is a distance measure (see [19] for the exact definition). Denoting the point with the highest priority as $\mathbf{p}^*$, a new point is inserted at the midpoint between $\mathbf{p}^*$ and neighbor $\mathbf{p}_{i*}^*$, where $\mathbf{p}_{i*}^* = \arg\max_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p}^*)} B(\mathbf{p}^*, \mathbf{p}_i)$. In the original EAR method, the insertion is done iteratively, requiring an update of the neighborhood information and recalculation of B after every insertion. Instead, to allow parallel computation at GPU, we approximate this process by simultaneously inserting a maximum of $|\mathcal{P}|/10$ points each step. In this way, however, inserting the midpoint of point pairs would lead to duplicated new points. Thus, we insert asymmetrically at $\frac{1}{3}(\mathbf{p}_{i*}^* + 2\mathbf{p}^*)$ instead.

We then project the upsampled iso-points to the iso-surface. As shown in Figure 2, the final iso-points successfully reflect the 3D geometry of the current implicit surface.

Compared with using marching cubes to extract the iso-surface, our adaptive sampling is efficient. Since both the resampling and upsampling steps only require information of local neighborhood, we implement it on the GPU. Furthermore, since we use the iso-points from the previous iteration for initialization, the overall point distribution improves as the training stabilizes, requiring fewer (or even zero) resampling and upsampling steps at later stages.

3.2. Utilizing iso-points in optimization

We introduce two scenarios of using iso-points to guide neural implicit surface optimization: (i) importance sampling for multi-view reconstruction and (ii) regularization when reconstructing neural implicit surfaces from raw input point clouds.

Iso-points for importance sampling. Optimizing a neural implicit function to represent a high-resolution 3D shape requires abundant training samples – specifically, many supervision points sampled close to the iso-surface to capture the fine-grained geometry details accurately. However, for

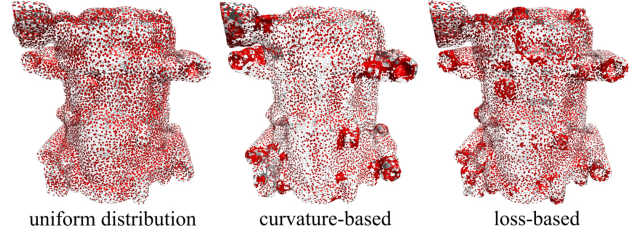uniform distribution   curvature-based   loss-based

Figure 3: Examples of importance sampling based on different saliency metrics. The default uniform iso-points treat different regions of the iso-surface equally. We compute different saliency metrics on the uniform iso-points, based on which more samples can be gathered in the salient regions. The curvature-based metric emphasizes geometric details, while the loss-based metric allows for hard example mining.

applications where the explicit 3D geometry is not available during training, the question of how to generate training samples remains mostly unexplored.

We exploit the geometry information and the prediction uncertainty carried by the iso-points during training. The main idea is to compute a saliency metric on iso-points, then add more samples in those regions with high saliency. To this end, we experiment with two types of metrics: curvature-based and loss-based. The former aims at emphasizing geometric features, typically reflected by high curvature. The latter is a form of hard example mining, as we sample more densely where the higher loss occurs, as shown in Figure 3.

Since the iso-points are uniformly distributed, the curvature can be approximated by the norm of the Laplacian, *i.e.* $\mathcal{R}_{\text{curvature}}(\mathbf{p}) = \|\mathbf{p} - \sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \mathbf{p}_i\|$. For the loss-based metric, we project the iso-points on all training views and compute the average loss at each point, *i.e.* $\mathcal{R}_{\text{loss}}(\mathbf{p}) = \frac{1}{N} \sum_i^N \text{loss}(\mathbf{p})$, where N is the number of occurrences of $\mathbf{p}$ in all views.

As both metrics evolve smoothly, we need not update them in each training iteration. Denoting the iso-points at which we computed the metric as $\mathcal{T}$ and the subset of template points with high metric values by $\mathcal{T}^* = \{\mathbf{t}^*\}$, the metric-based insertion for each point $\mathbf{p}$ in the current iso-point set $\mathcal{P}_t$ can be written as

$$\mathbf{p}_{\text{new},i} = \tfrac{2}{3}\mathbf{p} + \tfrac{1}{3}\mathbf{p}_i, \forall \mathbf{p}_i \in \mathcal{N}(\mathbf{p}), \text{ if } \min_{\mathbf{t}^*} \|\mathbf{p} - \mathbf{t}^*\| \leq \sigma.$$

The neighborhood radius σ is the same one used in (3).

Iso-points for regularization. The access to an explicit representation of the implicit surface also enables us to incorporate geometry-motivated priors into the optimization objective, exerting finer control of the reconstruction result.

Let us consider fitting a neural implicit surface to a point cloud. Depending on the acquisition process, the point cloud may be sparse, distorted by noise, or distributed highly unevenly. Similar to previous works [48, 49], we observe that the neural network tends to reconstruct a smooth surface at the early stage of learning, but then starts to pick
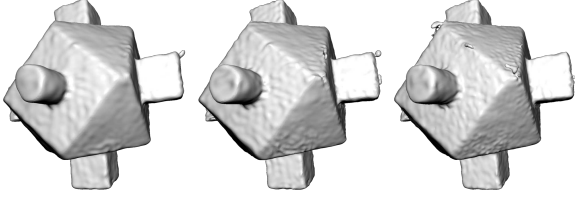
Figure 4: Progression of overfitting. When optimizing a neural implicit surface on a noisy point cloud, the network initially outputs a smooth surface, but increasingly overfits to the noise in the data. Shown here are the reconstructed surfaces after 1000, 2000 and 5000 iterations. The input point cloud is acquired in-house using an Artec Eva scanner.
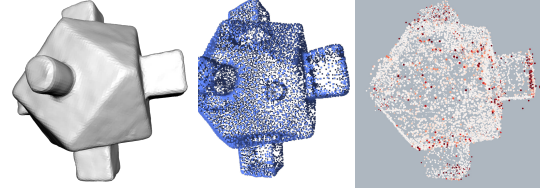

Figure 5: Iso-points for regularization. At the early stage of the training, the implicit surface is smooth (left), and we extract iso-points (middle) as a reference shape, which can facilitate various regularization terms. In the example on the right, we use the iso-points to reduce the influence of outliers (shown in red).

up higher frequency signals from the corrupted observations and overfits to noise and other caveats in the data, as shown in Figure 4. This characteristic is consistent across network architectures, including those designed to accommodate high-frequency information, such as SIREN.

Existing methods that address overfitting include early stopping, weight decay, drop-out *etc*. [13]. However, whereas these methods are generic tools designed to improve network generalizability for training on large datasets, we propose a novel regularization approach that is specifically tailored to training neural implicit surfaces and serves as a platform to integrate a plethora of 3D priors from classic geometry processing.

Our main idea is to use the iso-points as a consistent, smooth, yet dynamic approximation of the reference geometry. Consistency and smoothness ensure that the optimization does not fluctuate and overfit to noise; the dynamic nature lets the network pick up consistent high-frequency signals governed by underlying geometric details.

To this end, we extract iso-points after a short warmup training (*e.g.* 300 iterations). Because of the aforementioned smooth characteristic of the network, the noise level in the initial iso-points is minimal. Then, during subsequent training, we update the iso-points periodically (*e.g.* every 1000 iterations) to allow them to gradually evolve as the network learns more high-frequency information.

The utility of the iso-points includes, but is not limited to 1) serving as additional training data to circumvent data scarcity, 2) enforcing additional geometric constraints, 3) filtering outliers in the training data.

Specifically, for sparse or hole-ridden input point clouds, we take advantage of the uniform distribution of iso-points and augment supervision in undersampled areas by enforcing the signed distance value on all iso-points to be zero:

$$\mathcal{L}_{\text{isoSDF}} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} |f(\mathbf{p})|. \tag{8}$$

Given the iso-points, we compute their normals from their local neighborhood using principal component analysis (PCA) [18]. We then increase surface smoothness by enforcing consistency between the normals estimated by PCA

and those computed from the gradient of the network, *i.e.*

$$\mathcal{L}_{\text{isoNormal}} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} (1 - |\cos(J_f^{\mathsf{T}}(\mathbf{p}), \mathbf{n}_{\text{PCA}})|). \tag{9}$$

The larger the PCA neighborhood is, the smoother the reconstruction becomes. Optionally, additional normal filtering can be applied after PCA to reduce over-smoothing and enhance geometric features.

Finally, we can use the iso-points to filter outliers in the training data. Specifically, given a batch of training points $\mathcal{Q} = \{\mathbf{q}\}$, we compute a per-point loss weight based on their alignment with the iso-points. Here, we choose to use bilateral weights to take both the Euclidean and the directional distance into consideration. Denoting the normalized gradient of an iso-point $\mathbf{p}$ and a training point $\mathbf{q}$ as $\mathbf{n_p}$ and $\mathbf{n_q}$, respectively, the bilateral weight can written as

$$v(\mathbf{q}) = \min_{\mathbf{p} \in \mathcal{P}} \phi(\mathbf{n_p}, \mathbf{p} - \mathbf{q}) \psi(\mathbf{n_p}, \mathbf{n_q}), \quad \text{with} \tag{10}$$

$$\psi(\mathbf{n_p}, \mathbf{n_q}) = e^{-\left(1 - \frac{1 - \mathbf{n_p^{\mathsf{T}} n_q}}{1 - \cos(\sigma_n)}\right)^2}, \tag{11}$$

where σ_n regulates the sensitivity to normal difference; we set $\sigma_n = 60°$ in our experiments. This loss weight can be incorporated into the existing loss functions to reduce the impact of outliers. A visualization of the outliers detected by this weight is shown in Figure 5.

4. Results

Iso-points can be incorporated into the optimization pipelines of existing state-of-the-art methods for implicit surface reconstruction. In this section, we demonstrate the benefits of the specific techniques introduced in Section 3.2.

We choose state-of-the-art methods as the baselines, then augment the optimization with iso-points. Results show that the augmented optimization outperforms the baseline quantitatively and qualitatively.

4.1. Sampling with iso-points

We evaluate the benefit of utilizing iso-points to generate training samples for multi-view reconstruction. As the baseline, we employ the ray-tracing algorithm from a state-of-the-art neural implicit renderer IDR [53], which generates

sampling with ray-tracing

sampling with iso-points

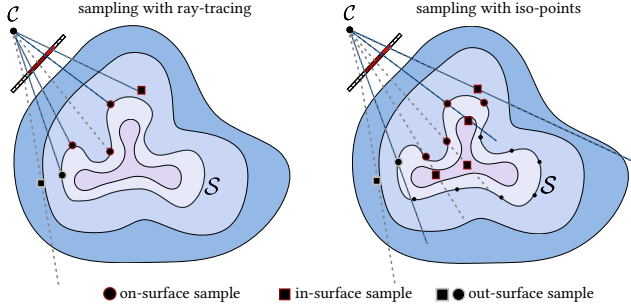● on-surface sample   ■ in-surface sample   ■● out-surface sample

Figure 6: A 2D illustration of two sampling strategies for multi-view reconstruction. Ray-tracing (left) generates training samples by shooting rays from camera $\mathcal{C}$ through randomly sampled pixels; depending on whether an intersection is found and whether the pixel lies inside the object silhouette, three types of samples are generated: on-surface, in-surface and out-surface points. We generate on-surface samples directly from iso-points, obtaining evenly distributed samples on the implicit surface, and also use the iso-points to generate more reliable in-surface samples.

	baseline (ray-tracing)	uniform	curvature-based	loss-based
CD $\cdot 10^{-4}$(position)	17.24	1.80	1.83	**1.71**
CD $\cdot 10^{-1}$(normal)	1.51	1.10	0.99	**0.95**

Table 1: Quantitative effect of importance sampling with iso-points. Compared to the baseline, which generates training points via ray-marching, we use iso-points to draw more attention on the implicit surface. The result is averaged over 10 models selected from the Sketchfab dataset [55].

training samples by ray-marching from the camera center through uniformly sampled pixels in the image. As shown in Figure 6, three types of samples are used for different types of supervision: on-surface samples, which are ray-surface intersections inside the object's 2D silhouette, in-surface samples, which are points with the lowest signed distance on the non-intersecting rays inside 2D silhouette, and out-surface samples, which are on the rays outside the 2D silhouette either at the surface intersection or at the position with the lowest signed distance.

On this basis, we incorporate the iso-points directly as on-surface samples. We can direct the learning attention by varying the distribution of iso-points using the saliency metrics described in Section 3.2. The iso-points also provide us prior knowledge to generate more reliable in-surface samples. More specifically, as shown in Figure 6 (right), we generate the three types of samples as follows: a) on-surface samples: we remove occluded iso-points by visibility testing using a point rasterizer, and select those iso-points whose projections are inside the object silhouette; b) in-surface samples: on the camera rays that pass through the on-surface samples, we determine the point with the lowest signed distance on the segment between the on-surface sample and the farther intersection with the object's bounding sphere. c) out-surface samples: we shoot camera rays through pixels outside the object silhouette, and choose the point with the lowest signed distance.

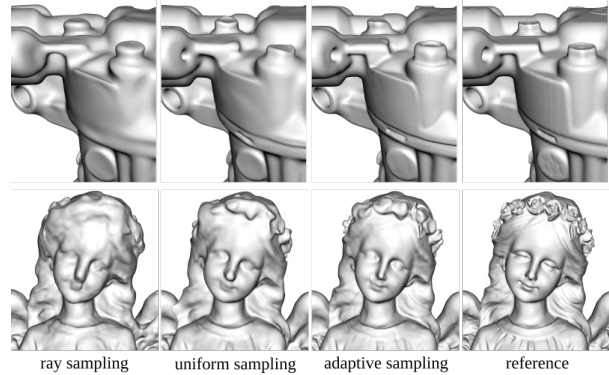ray sampling   uniform sampling   adaptive sampling   reference

Figure 7: Qualitative comparison between sampling strategies for multi-view reconstruction. Using the same optimization time and similar total sample count, sampling the surface points with uniformly distributed iso-points considerably improves the reconstruction accuracy. A loss based importance sampling further improves the recovery of small-scale structures. The models shown here are the COMPRESSOR and ANGEL2 from the Sketchfab dataset [55].

Below, we demonstrate two benefits of the proposed sampling scheme.

Surface details from importance sampling. First, we examine the effect of drawing on-surface samples using iso-points by comparing the optimization results under fixed optimization time and the same total sample count.

As inputs, we render 512×512 images per object under known lighting and material from varied camera positions. When training with the iso-points, we extract 4,000 iso-points after 500 iterations, then gradually increase the density until reaching 20,000 points. To match the sample count, in the ray-tracing variation, we randomly draw 2,048 pixels per image and then increase the sample count until reaching 10,000 pixels. We use a 3-layer SIREN model with the frequency multipliers set to 30, and optimize with a batch size of 4.

We evaluate our method quantitatively using 10 watertight models from the Sketchfab dataset [55]. As shown in Table 1, we compute 2-way chamfer point-to-point distance ($\|\mathbf{p}_i - \mathbf{p}_j\|^2$) and normal distance ($1 - \cos(\mathbf{n}_i, \mathbf{n}_j)$) on 50K points, uniformly sampled from the reconstructed meshes.

Results show that using uniform iso-points as on-surface samples compares favorably against the baseline, especially in the normal metric. It suggests that we achieve higher fidelity on the finer geometric features, as our surface samples overcome under-sampling issues occurring at small scale details. We also see that importance sampling with iso-points and loss based upsampling exhibits a substantial advantage over other variations, demonstrating the effectiveness of smart allocation of the training samples according to the current learning state. In comparison, curvature-based sampling performs similarly to the baseline, but notability worse than with the uniform iso-points. We observe that the iso-points, in this case, are highly concentrated on a
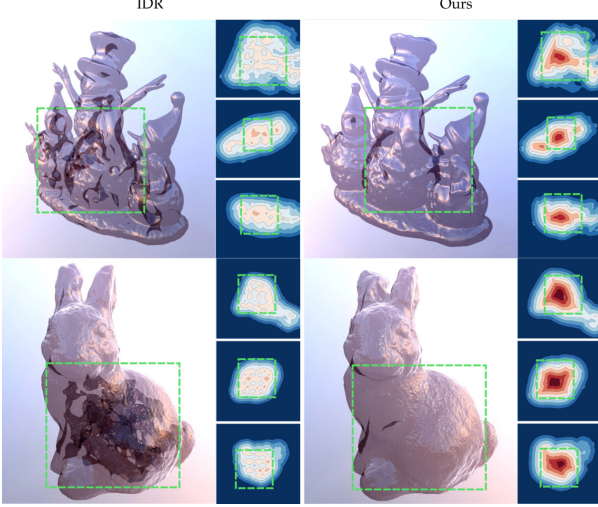
Figure 8: Topological correctness of the reconstructed surface in multi-view reconstruction. The erroneous inner structures of IDR are indicated by the artifacts in the images rendered by Blender [17] (using glossy and transparent material) and by the contours of the signed distance field on cross-sections.

few spots on the surface and ignore regions where the current reconstruction is problematic (Figure 3).

The improvement is more pronounced qualitatively, as shown in Figure 7. Sampling on-surface with uniform iso-points clearly enhances reconstruction accuracy compared to the baseline with ray-tracing. The finer geometric details further improve with loss-based importance sampling.

Topological correctness from 3D prior. IDR can reconstruct impressive geometric details on the DTU dataset [21], but a closer inspection shows that the reconstructed surface contains a considerable amount of topological errors inside the visible surface. We use iso-points to improve the topological accuracy of the reconstruction.

We use the same network architecture and training protocol as IDR, which samples 2048 pixels from a randomly chosen view in each optimization iteration. We use uniform iso-points in this experiment. To keep a comparable sample count, we subsample the visible iso-points to obtain a maximum of 1500 on-surface samples per iteration. Since our strategy automatically creates more in-surface samples (as shown in Figure 6), we halve the loss weight on the in-surface samples compared to the original implementation.

We visualize the topology of the reconstructed surface in Figure 8. To show the inner structures of the surface, we render it in transparent and glossy material with a physically-based renderer [17] and show the back faces of the mesh. Dark patches in the rendered images indicate potentially erroneous light transmission caused by inner structures. Similarly, we also show the contour lines of the iso-surface on a cross-section to indicate the irregularity of the reconstructed implicit function. In both visualizations, the incorrect topology in IDR reconstruction is

apparent. In contrast, our sampling enables more accurate reconstruction of the signed distance field inside and outside the surface with more faithful topological structure.

4.2. Regularization with iso-points

We evaluate the benefit of using iso-points for regularization. As an example application, we consider surface reconstruction from a noisy point cloud.

As our baseline method, we use the publicly available SIREN codebase and adopt their default optimization protocol. The noisy input point clouds are either acquired with a 3D scanner (Artec Eva) or reconstructed [12] from the multi-view images in the DTU dataset.

In each optimization iteration, the baseline method randomly samples an equal number of oriented surface points $\mathcal{Q}_s = \{\mathbf{q}_s, \mathbf{n}_s\}$ from the input point cloud and unoriented off-surface points $\mathcal{Q}_o = \{\mathbf{q}_o\}$ from bounding cube's interior. The optimization objective is comprised of four parts:

$$\mathcal{L} = \gamma_{\text{onSDF}}\mathcal{L}_{\text{onSDF}} + \gamma_{\text{normal}}\mathcal{L}_{\text{normal}} + \gamma_{\text{offSDF}}\mathcal{L}_{\text{offSDF}} + \gamma_{\text{eikonal}}\mathcal{L}_{\text{eikonal}},$$

where

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

$$\mathcal{L}_{\text{offSDF}} = \frac{\sum_{\mathbf{q}_o \in \mathcal{Q}_o} e^{-\alpha |f(\mathbf{q}_o)|}}{|\mathcal{Q}_o|}, \quad \mathcal{L}_{\text{eikonal}} = \frac{\sum_{\mathbf{q} \in \mathcal{Q}_o \cup \mathcal{Q}_s} |1 - \|J_f^{\mathsf{T}}(\mathbf{q})\|\|}{|\mathcal{Q}_s| + |\mathcal{P}_o|} \text{ and}$$

$\gamma_{\text{onSDF}} = 1000, \gamma_{\text{normal}} = 100, \gamma_{\text{offSDF}} = 50, \gamma_{\text{eikonal}} = 100$. We alter this objective with the outlier-aware loss weight defined in (11), and then add the regularizations on iso-points $\mathcal{L}_{\text{isoSDF}}$ (8) and $\mathcal{L}_{\text{isoNormal}}$ (9). The final objective becomes

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

where the loss terms with on-surface points are weighted as follows:

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

The iso-points are initialized by subsampling the input point cloud by 1/8 and updated every 2000 iterations.

The comparison with the baseline, i.e., vanilla optimization without regularization, is shown in Figure 9. For the DTU-MVS data, we also conduct quantitative evaluation following the standard DTU protocol as shown in Table 2, i.e., L1-Chamfer distance between the reconstructed and reference point cloud within a predefined volumetric mask [1]. For clarity, we also show the results of screened Poisson reconstruction [26] and Points2Surf [11]. The former reconstructs a watertight surface from an oriented point set by solving local Poisson equations; the latter fits an implicit neural function to an unoriented point set in a global-to-local manner. Compared with the baseline and screened Poisson, our proposed regularizations significantly
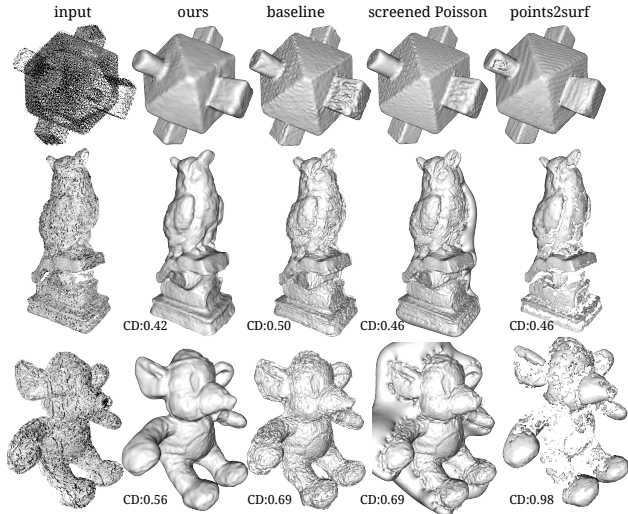
Figure 9: Implicit surface reconstruction from noisy and sparse point clouds. From left to right: input, reconstruction with our proposed regularizations, baseline reconstruction without our regularizations, screened Poisson reconstruction and Points2Surf. CD denotes L1-Chamfer distance. The sparse point cloud in the first row is acquired with an Artec Eva scanner, whereas the inputs in the second row and third row are reconstructed from DTU dataset (model 105 and 122) using [12].

ID	ours	baseline	point2surf	screened Poisson
55	**0.37**	0.41	0.56	0.42
69	**0.59**	0.65	0.61	0.63
105	**0.56**	0.69	0.98	0.69
110	0.54	**0.51**	0.61	0.55
114	**0.38**	0.45	0.45	0.37
118	**0.45**	0.49	0.59	0.55
122	**0.42**	0.50	0.46	0.46
Average	0.53	0.61	0.52	**0.47**

Table 2: Quantitative evaluation for surface reconstruction from a noisy sparse point cloud. We evaluate the two-way L1-chamfer distance on a subset of the DTU-MVS dataset.

suppresses noise. Points2Surf can handle noisy input well, but the sign propagation appears to be sensitive to the point distribution, leading to holes in the reconstructed mesh. Moreover, since their model does not use the points' normal information, the reconstruction lacks detail.

4.3. Performance analysis

The main overhead in our approach is the projection step. One newton iteration requires a forward and a backward pass. On average, the projection terminates within 4 iterations. This procedure is optimized by only considering points that are not yet converged at each iteration.

Empirically, the computation time of extracting the iso-points once is typically equivalent to running 3 training iterations. In practice, as we extract iso-points only periodically, the total optimization time only increases marginally.

In the case of multi-view reconstruction, since the ray-marching itself is an expensive operation, involving multiple forward passes per ray, the overhead of our approach

is much less notable. As discussed, we filter the occluded iso-points before the projection, which also saves optimization time. The trade-off between optimization speed and quality is depicted in a concrete example in Fig. 10, where we plot the evolution of the chamfer distance during the optimization of the COMPRESSOR model (first row of Fig. 7). Compared to the baseline optimization with ray-tracing, it is evident that the iso-points augmented optimization consistently achieves better results at every timestamp. In other words, with iso-points we can reach a given quality threshold faster.
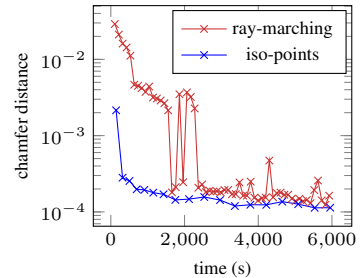

Figure 10: Validation error in relation to optimization time. The first time stamp is at the 100-th iteration.

5. Conclusion and Future Work

In this paper, we leverage the advantages of two interdependent representations: neural implicit surfaces and iso-points, for 3D learning. Implicit surfaces can represent 3D structures in arbitrary resolution and topology, but lack an explicit form to adapt the optimization process to input data. Iso-points, as a point cloud adaptively distributed on the underlying surface, are fairly straightforward and efficient to manipulate and analyze the underlying 3D geometry.

We present effective algorithms to extract and utilize iso-points. Extensive experiments show the power of our hybrid representation. We demonstrate that iso-points can be readily employed by state-of-the-art neural 3D reconstruction methods to significantly improve optimization efficiency and reconstruction quality.

A limitation of the proposed sampling strategy is that it is mainly determined by the geometry of the underlying surface and does not explicitly model the appearance. In the future, we would like to extend our hybrid representation to model the joint space of geometry and appearance, which can in turn allow us to apply path-tracing for global illumination, bridging the gap between existing neural rendering approaches and classic physically based rendering. Another promising direction is utilizing the new representation for consistent space-time reconstructions of articulated objects.

Acknowledgement

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16, 2016. 7

[2] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. In *Proc. IEEE Int. Conf. on Neural Information Processing Systems (NeurIPS)*, pages 2034–2043, 2019. 2, 3

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 2565–2574, 2020. 2

[4] Matan Atzmon and Yaron Lipman. Sald: Sign agnostic learning with derivatives, 2020. 2

[5] Adi Ben-Israel. A newton-raphson method for the solution of systems of equations. *Journal of Mathematical analysis and applications*, 15(2):243–252, 1966. 3

[6] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. of SIGGRAPH*, pages 67–76, 2001. 2

[7] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5939–5948, 2019. 1, 2

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *Proc. Euro. Conf. on Computer Vision*, pages 628–644. Springer, 2016. 1

[9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, pages 303–312, 1996. 2

[10] Luiz Henrique de Figueiredo, Jonas de Miranda Gomes, Demetri Terzopoulos, and Luiz Velho. Physically-based methods for polygonization of implicit surfaces. In *Graphics Interface*, volume 92, pages 250–257, 1992. 2

[11] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *Proc. Euro. Conf. on Computer Vision*, pages 108–124. Springer, 2020. 2, 7

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 32(8):1362–1376, 2009. 1, 7, 8

[13] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT press Cambridge, 2016. 5

[14] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. arXiv preprint arXiv:2002.10099, 2020. 2

[15] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 216–224, 2018. 2

[16] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 2

[17] Roland Hess. *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Focal Press, 2010. 7

[18] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, pages 71–78, 1992. 5

[19] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 32(1):1–12, 2013. 2, 3, 4

[20] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Proc. IEEE Int. Conf. on Neural Information Processing Systems (NeurIPS)*, pages 2802–2812, 2018. 1

[21] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 406–413. IEEE, 2014. 1, 7

[22] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 1251–1261, 2020. 1

[23] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. arXiv preprint arXiv:2006.12057, 2020. 2

[24] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 3907–3916, 2018. 1

[25] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proc. Eurographics Symp. on Geometry Processing*, volume 7, 2006. 2

[26] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. on Graphics*, 32(3):1–13, 2013. 7

[27] Florian Levet, Xavier Granier, and Christophe Schlick. Fast sampling of implicit surfaces by particle systems. In *IEEE International Conference on Shape Modeling and Applications*, pages 39–39. IEEE, 2006. 2

[28] David Levin. The approximation power of moving least-squares. *Mathematics of computation*, 67(224):1517–1531, 1998. 2

[29] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. arXiv preprint arXiv:1706.07036, 2017. 1

[30] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. SDF-SRN: Learning signed distance 3D object reconstruction from static images. arXiv preprint arXiv:2010.10505, 2020. 2

[31] Hsueh-Ti Derek Liu, Michael Tao, and Alec Jacobson. Paparazzi: surface editing by way of multi-view image processing. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 37(6):221–1, 2018. 1

[32] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *Proc. Int. Conf. on Computer Vision*, pages 7708–7717, 2019. 1

[33] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without supervision. In *Proc. IEEE Int. Conf. on Neural Information Processing Systems (NeurIPS)*, pages 8295–8306, 2019. 2

[34] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 2019–2028, 2020. 2

[35] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. arXiv preprint arXiv:1906.07751, 2019. 1

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4460–4470, 2019. 1

[37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. arXiv preprint arXiv:2003.08934, 2020. 3

[38] 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. 2

[39] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)*, 38(6):1–17, 2019. 1

[40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 165–174, 2019. 1, 2

[41] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 1

[42] Omid Poursaeed, Matthew Fisher, Noam Aigerman, and Vladimir G Kim. Coupling explicit and implicit surface representations for generative 3D modeling. arXiv preprint arXiv:2007.10294, 2020. 2

[43] João Proença, Joaquim A Jorge, and Mario Costa Sousa. Sampling point-set implicits. In *Eurographics Symposium on Point-Based Graphics*, pages 11–18, 2007. 2

[44] Vincent Sitzmann, Julien NP Martel, Alexander W Bergman, David B Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. arXiv preprint arXiv:2006.09661, 2020. 2, 3

[45] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Proc.*

[46] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. arXiv preprint arXiv:2004.03805, 2020. 2

[47] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 2626–2634, 2017. 1

[48] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 9446–9454, 2018. 4

[49] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 10130–10139, 2019. 4

[50] Andrew P Witkin and Paul S Heckbert. Using particles to sample and control implicit surfaces. In *ACM Trans. on Graphics (Proc. of SIGGRAPH)*, pages 269–277, 1994. 2

[51] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-Monte Carlo sampling for deep implicit field based 3D reconstruction with symmetry. arXiv preprint arXiv:2007.13393, 2020. 2

[52] Lior Yariv, Matan Atzmon, and Yaron Lipman. Universal differentiable renderer for implicit neural representations, 2020. 2

[53] 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. 5

[54] 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

[55] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019. 1, 6