

001 **Implicit Filtering for Learning Neural Signed**
002 **Distance Functions from 3D Point Clouds**

001
002

003 Anonymous ECCV 2024 Submission

003

004 Paper ID #960

004

005 **Abstract.** Neural signed distance functions (SDFs) have shown power-
006 ful ability in fitting the shape geometry. However, inferring continuous
007 signed distance fields from discrete unoriented point clouds still remains
008 a challenge. The neural network typically fits the shape with a rough
009 surface and omits fine-grained geometric details such as shape edges and
010 corners. In this paper, we propose a novel non-linear implicit filter to
011 smooth the implicit field while preserving high-frequency geometry de-
012 tails. Our novelty lies in that we can filter the surface (zero level set) by
013 the neighbor input points with gradients of the signed distance field. By
014 moving the input raw point clouds along the gradient, our proposed im-
015 plicit filtering can be extended to non-zero level sets to keep the promise
016 consistency between different level sets, which consequently results in a
017 better regularization of the zero level set. We conduct comprehensive ex-
018 periments in surface reconstruction from objects and complex scene point
019 clouds, the numerical and visual comparisons demonstrate our improve-
020 ments over the state-of-the-art methods under the widely used bench-
021 marks.

022 1 Introduction

022

023 Reconstructing surfaces from 3D point clouds is an important task in 3D com-
024 puter vision. Recently signed distance functions (SDFs) learned by neural net-
025 works have been a widely used strategy for representing high-fidelity 3D geome-
026 try. These methods train the neural networks to predict the signed distance for
027 every position in the space by signed distances from ground truth or inferred
028 from the raw 3D point cloud. With the learned signed distance field, we can
029 obtain the surface by running the marching cubes algorithm [23] to extract the
030 zero level set.

023
024
025
026
027
028
029
030

031 Without signed distance ground truth, inferring the correct gradient and
032 distance for each query point could be hard. Since the gradient of the neural
033 network also indicates the direction in which the signed distance field changes,
034 recent works [1–3, 13, 25, 33] typically add constraints on the network gradient
035 to learn a stable field. In terms of the rate at which the field is changing, the
036 eikonal term [1, 2, 5, 33] is widely used to ensure the norm of the gradient to be one
037 everywhere. For the gradient direction constraint, some methods [3, 10] use the
038 direction from the query point to the nearest point on the surface as guidance.
039 Leveraging the continuity of the neural network and the gradient constraint, all

031
032
033
034
035
036
037
038
039

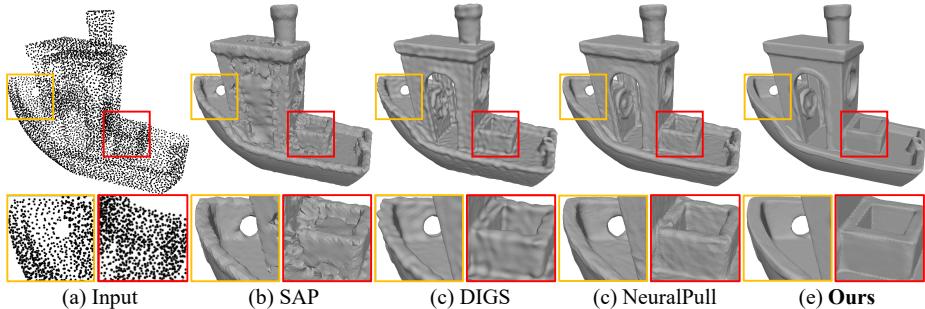


Fig. 1: Visualization of the comparisons on FAMOUS dataset [11]. Our implicit filter can improve the reconstruction by removing the noise and keeping the geometric details compared with other methods.

these methods could reconstruct discrete points. However, the continuity cannot guarantee the prediction is correct everywhere. Therefore, reconstructed surfaces of previous methods usually contain noise and ignore geometry details when there are not enough points to guide the reconstruction, as shown in Fig. 1.

The above issue arises from the fact that these methods overlook the geometric information within the neighborhood but only focus on adding constraints on individual points to optimize the network. To resolve this issue, we introduce the bilateral filter for implicit fields that reduces surface noise while preserving the high-frequency geometric characteristics of the shape. Our designed implicit filter takes into account both the position of point clouds and the gradient of learned implicit fields. Based on the assumption of all input points lying on the surface, we can filter noise points on the zero level set by minimizing the weighted projection distance to gradients of the neighbor input points. Moreover, by moving the input points along the gradient of the field to other level sets, we can easily extend the filter to the whole field. This helps constrain the signed distance field near the surface and achieve better consistency through different level sets. To evaluate the effectiveness of our proposed implicit filtering, we validate it under widely used benchmarks including object and scene reconstructions. Our contributions are listed below.

- We introduce the implicit filter on SDFs to smooth the surface while preserving geometry details for learning better neural networks to represent shapes or scenes.
- We improve the implicit filter by extending it to non-zero level sets of signed distance fields. This regularization of the field aligns different level sets and provides better consistency within the whole SDF field.
- Both object and scene reconstruction experiments validate our implicit filter, demonstrating its effectiveness and ability to produce high-fidelity reconstruction results, surpassing the previous state-of-the-art methods.

068 2 Related Work

069 With the rapid development of deep learning, neural networks have shown great
070 potential in surface reconstruction from 3D point clouds. In the following, we
071 briefly review methods related to implicit learning for 3D shapes and reconstruc-
072 tions from point clouds.

073 **Implicit Learning from 3D Supervision.** The most commonly used strategy
074 to train the neural network is to learn priors in a data-driven manner. These
075 methods require signed distances or occupancy labels as 3D supervision to learn
076 global priors [6, 11, 22, 26, 27] or local priors [7, 16, 19, 30, 35, 36, 38]. With large-
077 scale training datasets, the neural network can perform well with similar shapes,
078 but may not generalize well to unseen cases with large geometric variations.
079 These models often have limited inputs that can be difficult to scale for varying
080 sizes of point clouds.

081 **Implicit Learning from Raw Point Clouds.** Different from the supervised
082 methods, we can learn implicit functions by overfitting neural networks on sin-
083 gle point clouds globally or locally to learn SDFs. These unsupervised methods
084 rely on neural networks to infer implicit functions without learning any priors.
085 Therefore, apart from the guidance of original input point clouds, we also need
086 constraints on the direction [3, 4, 10] or the norm [1, 2] of the gradients, spec-
087 cially designed priors [4, 24], or differentiable poisson solver [29] to infer SDFs.
088 This unsupervised approach heavily depends on the fitting capability and con-
089 tinuity of neural networks. However, these SDFs lack accuracy because there is
090 no reliable guidance available for each query point across the entire space when
091 working with discrete point clouds. Therefore, deducing the correct geometry for
092 free space becomes particularly crucial. Our implicit filtering enhances SDFs by
093 inferring the geometric details through the implicit field information of neighbor
094 points.

095 **Feature Preserving Point Cloud Reconstruction.** Early works [15, 20, 28]
096 reconstruct point clouds with sharp features usually by point cloud consolidation.
097 The key idea of these methods is to enhance the quality of point clouds with
098 sharp features. One popular category is the local projection operation (LOP) [21]
099 and its variants [14, 15, 20, 31]. The projection operator provides a stable and
100 easily generalizable method for point cloud filtering, which is also the foundation
101 of our implicit filter. The difference lies in that we do not need any normal
102 or other priors and our filtering can be directly applied to implicit fields to
103 extract high-fidelity meshes. Some other learning-based methods [41, 42] try to
104 consolidate point clouds with edge points in a data-driven manner. Although
105 capable of generating high-quality point clouds, these methods still require a
106 proper reconstruction method [12] to inherit the details in meshes.

107 With the advancement of deep learning in point cloud reconstruction, some
108 approaches [5, 33, 37] also explored to employ neural networks to reconstruct high-
109 precision models. FFN [34], SIREN [33], and IDF [40] introduce high-frequency
110 features into the neural network in different ways to preserve the geometric de-
111 tails of the reconstructed shape. DIGS [5] and EPI [37] smooth the surface by
112 using the divergence as guidance to alleviate the implicit surface roughness. Com-

pared with these methods, we first introduce local geometric features through filtering to optimize the implicit field, so that we can achieve higher accuracy.

3 Method

Nerual SDFs overview. In this section we will briefly describe the concept we used in our implicit filtering. We focus on the SDF $f : \mathbf{R}^3 \rightarrow \mathbf{R}$ inferred from the point cloud $\mathbf{P} = \{\mathbf{p}_i | \mathbf{p}_i \in \mathbf{R}^3\}_{i=1}^N$ without ground truth signed distances and normals. f predicts a signed distance $s \in \mathbf{R}$ for an arbitrary query point \mathbf{q} , as formulated by $s = f_\theta(\mathbf{q})$, where θ denotes the parameters of the neural network.

The level set \mathcal{S}_d of SDF is defined as a set of continuous query points with the same signed distance d , formulated as $\mathcal{S}_d = \{\mathbf{q} | f_\theta(\mathbf{q}) = d\}$. The goal of our implicit filtering is to smooth each level set with geometry details. Then we can extract the zero level set which also represents the surface as a mesh by running the marching cubes algorithm [23].

Level set bilateral filtering. Filtering for 2D images replaces the intensity of each pixel with the weighted intensity values from nearby pixels. Different from images, the resolution of implicit fields is infinite and we need to find the neighborhood on each level set for filtering. By minimizing the following loss function,

$$L_{dist} = \frac{1}{N} \sum_{i=1}^N |f_\theta(\mathbf{p}_i)|, \quad (1)$$

we can approximate that all points in \mathbf{P} are located on level set \mathcal{S}_0 , which makes it feasible to find neighbor points for any position on this level set as shown in Fig. 3(a). For a given point $\bar{\mathbf{p}}$ on \mathcal{S}_0 , one simple strategy of filtering is to average positions of neighbor points $N(\bar{\mathbf{p}}, \mathcal{S}_0) \subset \mathbf{P}$ on \mathcal{S}_0 by a Gaussian filter based on relative positions the same in 2D as follows:

$$\bar{\mathbf{p}}_{\text{average}} = \frac{\sum_{\mathbf{p}_j \in N(\bar{\mathbf{p}}, \mathcal{S}_0)} \mathbf{p}_j \phi(||\bar{\mathbf{p}} - \mathbf{p}_j||)}{\sum_{\mathbf{p}_j \in N(\bar{\mathbf{p}}, \mathcal{S}_0)} \phi(||\bar{\mathbf{p}} - \mathbf{p}_j||)}, \quad (2)$$

where the Gaussian function ϕ is defined as $\phi(||\bar{\mathbf{p}} - \mathbf{p}_j||) = \exp\left(-\frac{||\bar{\mathbf{p}} - \mathbf{p}_j||^2}{\sigma_p^2}\right)$.

However, as depicted in Fig. 2, it is evident that this weighted mean position yields excessively smooth surfaces, causing sharp features and details to be further obscured. To keep the geometric details, our filtering operator suggests measuring the projection distance to the gradient of neighbor points as shown in Fig. 2 and Fig. 3(b). When calculating weights, it is vital to account for both

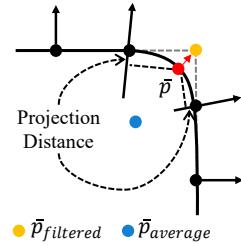


Fig. 2: By minimizing the weighted projection distance, our filter can preserve the sharp feature while the average method will lead to a wrong result.

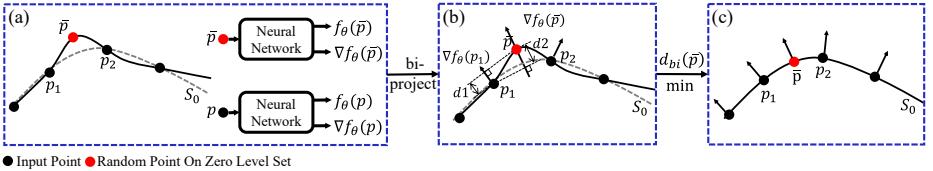


Fig. 3: Overview of filtering the zero level set. (a) We assume all input points lying on the surface and compute gradients as normals. (b) Calculating bidirectional projection distances $d_1 = |\mathbf{n}_{\bar{p}}^T (\bar{p} - \mathbf{p}_j)|$, $d_2 = |\mathbf{n}_{\bar{p}}^T (\bar{p} - \mathbf{p}_j)|$ and the weights in Eq. (4). (c) By minimizing Eq. (4), we can remove the noise on the zero level set. The gradient ∇f_θ in this figure defaults to be regularized.

the impact of relative positions and the gradient similarity. Following the principles of bilateral filtering, to compute the filtered point for \bar{p} , we simply need to minimize the following distance equation:

$$d(\bar{p}) = \frac{\sum_{\mathbf{p}_j \in N(\bar{p}, S_0)} |\mathbf{n}_{\bar{p}}^T (\bar{p} - \mathbf{p}_j)| \phi(||\bar{p} - \mathbf{p}_j||) \psi(\mathbf{n}_{\bar{p}}, \mathbf{n}_{\mathbf{p}_j})}{\sum_{\mathbf{p}_j \in N(\bar{p}, S_0)} \phi(||\bar{p} - \mathbf{p}_j||) \psi(\mathbf{n}_{\bar{p}}, \mathbf{n}_{\mathbf{p}_j})}, \quad (3)$$

where the gradient $\mathbf{n}_{\bar{p}}$, $\mathbf{n}_{\mathbf{p}_j}$ and the Gaussian function ψ are defined as $\mathbf{n}_{\bar{p}} = \frac{\nabla f_\theta(\bar{p})}{\|\nabla f_\theta(\bar{p})\|}$, $\mathbf{n}_{\mathbf{p}_j} = \frac{\nabla f_\theta(\mathbf{p}_j)}{\|\nabla f_\theta(\mathbf{p}_j)\|}$, $\psi(\mathbf{n}_{\bar{p}}, \mathbf{n}_{\mathbf{p}_j}) = \exp\left(-\frac{1 - \mathbf{n}_{\bar{p}}^T \mathbf{n}_{\mathbf{p}_j}}{1 - \cos(\sigma_n)}\right)$.

In addition to projection to the gradient $\mathbf{n}_{\mathbf{p}_j}$, we observe that the projection distance to $\mathbf{n}_{\bar{p}}$ can assist in learning a more stable gradient for point \bar{p} which is also adopted in EAR [15]. Taking into account the bidirectional projection, our final bilateral filtering operator can be formulated as follows:

$$d_{bi}(\bar{p}) = \frac{\sum_{\mathbf{p}_j \in N(\bar{p}, S_0)} \left(|\mathbf{n}_{\bar{p}}^T (\bar{p} - \mathbf{p}_j)| + |\mathbf{n}_{\bar{p}}^T (\bar{p} - \mathbf{p}_j)| \right) \phi(||\bar{p} - \mathbf{p}_j||) \psi(\mathbf{n}_{\bar{p}}, \mathbf{n}_{\mathbf{p}_j})}{\sum_{\mathbf{p}_j \in N(\bar{p}, S_0)} \phi(||\bar{p} - \mathbf{p}_j||) \psi(\mathbf{n}_{\bar{p}}, \mathbf{n}_{\mathbf{p}_j})}. \quad (4)$$

Although similar filtering methods have been widely studied in applications such as point cloud denoising and resampling [15, 42], there are two critical problems when applying these methods in implicit fields:

1. Filtering the zero level set needs to sample points on the level set S_0 , which necessitates the resolution of the equation $f_\theta = 0$, or the utilization of the marching cubes algorithm [23]. Both methods pose challenges in achieving fast and uniform point sampling. For the randomly sampled point q on non-zero level set $S_{f_\theta(q)}$, we can also not filter this level set since there are no neighbor points on $S_{f_\theta(q)}$.
2. The normals utilized in our filtering are derived from the gradients of the neural network f_θ . While the network typically offers reliable gradients, we may find that $\nabla f_\theta = 0$ is also the optimal solution to the minimum value of Eqs. (3) and (4). This degenerate solution is unexpected, as it implies a scenario where there is no surface when the gradient is zero everywhere.

We will concentrate on addressing the two issues in the subsequent sections.

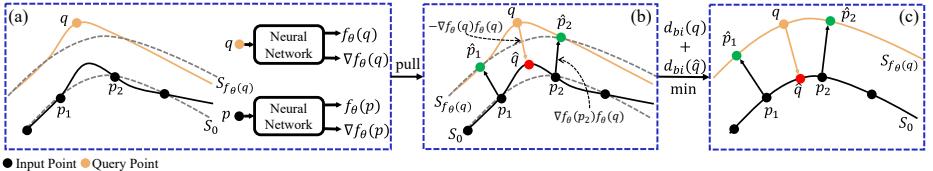


Fig. 4: Overview of sampling points. (a) Sampling query points near the surface. (b) Pulling the query point to the zero level set and input points to the level set where the query point is located. (c) Applying the filter on each level set. The gradient ∇f_θ in this figure defaults to be regularized.

Sampling points for filtering. Inspired by NeuralPull [3], we can pull a query point to the zero level set by the gradient of the neural network f_θ . For a given query point q as input, the pulled location \hat{q} can be formulated as follows:

$$\hat{q} = q - f_\theta(q)\nabla f_\theta(q)/\|\nabla f_\theta(q)\|. \quad (5)$$

The point q and \hat{q} lie respectively on level set $S_{f_\theta(q)}$ and S_0 as illustrated in Fig. 4(b). By adopting the sampling strategy in NeuralPull, we can generate samples $\mathbf{Q} = \{q_i | q_i \in \mathbf{R}^3\}_{i=1}^M$ on different level sets near the surface and pull them to S_0 by Eq. (5), to obtain $\hat{\mathbf{Q}} = \{\hat{q}_i | \hat{q}_i = q_i - f_\theta(q_i)\nabla f_\theta(q_i)/\|\nabla f_\theta(q_i)\|, q_i \in \mathbf{Q}\}_{i=1}^M$. Hence, we can filter the zero level set by minimizing Eq. (4) across all pulled query points $\hat{\mathbf{Q}}$, which is equivalent to optimizing the following loss:

$$L_{zero} = \sum_{\hat{q} \in \hat{\mathbf{Q}}} d_{bi}(\hat{q}), \quad (6)$$

where for each $\hat{q} \in \hat{\mathbf{Q}}$, $N(\hat{q}, S_0)$ denotes finding the neighbors of \hat{q} within the input points \mathbf{P} , since \mathbf{P} is assumed to be located on S_0 .

This filtering mechanism can be easily extended to non-zero level sets in a similar inverse manner. To be more specific, as for level set $S_{f_\theta(q)}$, the neighbor points for query point $q \in \mathbf{Q}$ are required. These points should lie on the level set $S_{f_\theta(q)}$ same as q , allowing us to filter the level set $S_{f_\theta(q)}$ using the same filter as described in Eq. (4).

However, obtaining $N(q, S_{f_\theta(q)})$ in \mathbf{P} is not feasible, since all input points \mathbf{P} are situated on the zero level set instead of the $S_{f_\theta(q)}$ level set. To address this issue, we propose a technique for identifying neighbors of q on level set $S_{f_\theta(q)}$, by projecting the input points \mathbf{P} inversely onto the specific level set $S_{f_\theta(q)}$ based on the gradient, as depicted in Fig. 4(b). The projected neighbor points can be represented as in Eq. (7). Filtering across multiple level sets helps to enhance the performance of our method by optimizing the consistency between different level sets within the SDF field. We further showcase this evidence in the ablation study detailed in Section Sec. 4.4.

$$N(q, S_{f_\theta(q)}) = \{\hat{p} | \hat{p} = p + f_\theta(q) \frac{\nabla f_\theta(p)}{\|\nabla f_\theta(p)\|}, p \in N(\hat{q}, S_0)\}. \quad (7)$$

Based on the above analysis, we can filter the level sets $S_{f_\theta(\mathbf{q})}$ by minimizing Eq. (4) over all sample points \mathbf{Q} through Eq. (7), equivalent to optimizing the following loss:

$$L_{field} = \sum_{\mathbf{q} \in \mathbf{Q}} d_{bi}(\mathbf{q}). \quad (8)$$

It is worth noting that for a fixed query point \mathbf{q} , the pulled query point $\hat{\mathbf{q}}$ dynamically changes while training the neural network, which results in a time-consuming process to repeatedly conduct neighbor searching for $\hat{\mathbf{q}}$. To handle this matter, we substitute the $N(\hat{\mathbf{q}}, \mathcal{S}_0)$ with $N(NN(\mathbf{q}), \mathcal{S}_0)$, where $NN(\mathbf{q}, \mathcal{S}_0)$ denotes the nearest point of \mathbf{q} within the input point cloud \mathbf{P} . While this substitution may introduce a slight bias to our network, we have included experimental results in our supplementary material to demonstrate that this trade-off between efficiency and accuracy is reasonable.

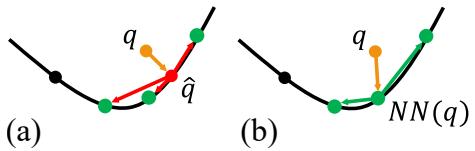


Fig. 5: (a) Searching neighbors directly for $\hat{\mathbf{q}}$. (b) Substituting $N(\hat{\mathbf{q}}, \mathcal{S}_0)$ with $N(NN(\mathbf{q}), \mathcal{S}_0)$.

Gradient constraint. The other problem of implicit filtering is gradient degeneration. Overfitting the neural network requires the SDF to be geometrically initialized. We can consider the initialized implicit field as the noisy field and apply our filter directly to train the network from the beginning to fit the raw point cloud by removing the ‘noise’. However, if the denoise target is too complex, gradient degeneration will occur during the training process. Therefore, we need to add a constraint to the gradient of the SDF.

There are two ways for training the neural network to pull the query points onto the surface according to NeuralPull [3] and CAP-UDF [43]. One is minimizing the distance between each query point \mathbf{q} and its nearest point $NN(\mathbf{q}) \in \mathbf{P}$ as formulated below:

$$L_{pull} = \frac{1}{M} \sum_{i \in [1, M]} \|\hat{\mathbf{q}}_i - NN(\mathbf{q}_i)\|_2. \quad (9)$$

The other is minimizing the Chamfer distance between moved query points and the raw point cloud:

$$L_{CD} = \frac{1}{M} \sum_{i \in [1, M]} \min_{j \in [1, N]} \|\hat{\mathbf{q}}_i - \mathbf{p}_j\|_2 + \frac{1}{N} \sum_{j \in [1, N]} \min_{i \in [1, M]} \|\mathbf{p}_j - \hat{\mathbf{q}}_i\|_2. \quad (10)$$

A stable SDF can be trained by the losses above since they are trying to move the query points to be in the same distribution with the point cloud, which can provide the constraint for our implicit filter. Here we choose L_{CD} since the filtered points are likely not the nearest points and L_{CD} is a more relaxed constraint.

238 Loss function. Finally, our loss function is formulated as:

$$239 \quad L = L_{zero} + \alpha_1 L_{field} + \alpha_2 L_{dist} + \alpha_3 L_{CD}, \quad (11)$$

240 where α_1, α_2 , and α_3 is the balance weights for our implicit filtering loss.

241 Implementation details. We employ a neural network similar to OccNet [26]
 242 and the geometric network initialization proposed in SAL [1] with a smaller
 243 radius the same as GP [10] to learn the SDF. We use the strategy in NeuralPull [3]
 244 to sample queries around each point p in P . We set the weight α_3 to 10 to
 245 constrain the learned SDF and α_1 and α_2 to 1. The parameters σ_n, σ_p are set to
 246 $15^\circ, \max_{p_j \in N(\bar{p}, \mathcal{S}_{f_\theta}(\bar{p}))} (||\bar{p} - p_j||)$ respectively.

247 4 Experiments

248 We conducted experiments to assess the performance of our implicit filter for sur-
 249 face reconstruction from raw point clouds. The results are presented for general
 250 shapes in Sec. 4.1, real scanned raw data including 3D objects in Sec. 4.2, and
 251 complex scenes in Sec. 4.3. Additionally, ablation experiments were carried out
 252 to validate the theory and explore the impact of various parameters in Sec. 4.4.

253 4.1 Surface Reconstruction for Shapes

254 **Datasets and metrics.** For
 255 surface reconstruction of gen-
 256 eral shapes from raw point
 257 clouds, we conduct evalua-
 258 tions on three widely used
 259 datasets including a subset of
 260 ShapeNet [8], ABC [18], and
 261 FAMOUS [11]. We use the
 262 same setting with NeuralPull
 263 [3] for the dataset ShapeNet.
 264 For datasets ABC and FA-
 265 MOUS, we use the train/test splitting released by Points2Surf [11] and we sample
 266 points directly from the mesh in the ABC dataset without other mesh prepro-
 267 cessing to keep the sharp features.

268 For evaluating the performance, we follow NeuralPull to sample 1×10^5 points
 269 from the reconstructed surfaces and the ground truth meshes on the ShapeNet
 270 dataset and sample 1×10^4 on the ABC and FAMOUS datasets. For the evalua-
 271 tion metrics, we use L1 and L2 Chamfer distance (CD_{L1} and CD_{L2}) to measure
 272 the error. Moreover, we adopt normal consistency (NC) and F-score to evalua-
 273 ate the accuracy of the reconstructed surface, the threshold is the same with
 274 NeuralPull.

254 **Table 1:** Comparisons on ABC and Famous
 255 datasets. The threshold of F-score (F-S.) is 0.01.

Methods	ABC			FAMOUS		
	CD_{L2}	CD_{L1}	F-S.	CD_{L2}	CD_{L1}	F-S.
P2S [11]	0.298	0.015	0.598	0.012	0.008	0.752
IGR [13]	2.675	0.063	0.448	1.474	0.044	0.573
NP [3]	0.095	0.011	0.673	0.100	0.012	0.746
PCP [4]	0.252	0.023	0.373	0.037	0.014	0.435
SIREN [33]	0.022	0.012	0.493	0.025	0.012	0.561
DIGS [5]	0.021	0.010	0.667	0.015	0.008	0.772
Ours	0.011	0.009	0.691	0.008	0.007	0.778

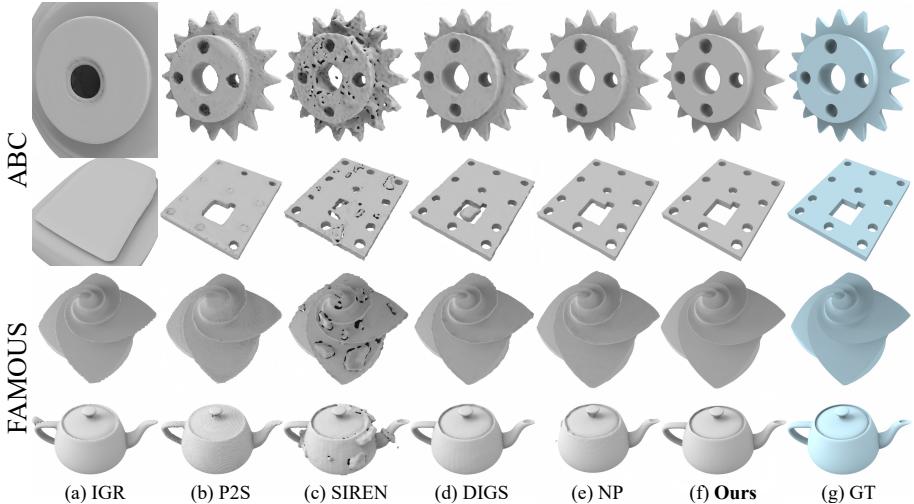


Fig. 6: Visual comparisons of surface reconstruction on ABC and FAMOUS datasets. Our method can reconstruct objects with sharp edges and less noise compared with other methods.

Comparisons. To evaluate the validity of our implicit filter, we compare our method with a variety of methods including SPSR [17], Points2Surf (P2S) [11], IGR [13], NeuralPull (NP) [3], LPI [9], PCP [4], GridPull (GP) [10], SIREN [33], DIGS [5]. The quantitative results on ABC and FAMOUS datasets are shown in Tab. 1, and selectively visualized in Fig. 6. Our model reaches state-of-the-art performance on both datasets, accomplishing the goal of eliminating noise on each level set while preserving the geometric details. To more intuitively validate the efficacy of our filtering, we visualize the level sets on a cross section in Fig. 7. We also report the results on ShapeNet which contains over 3000 objects in terms of CD_{L2} in Tab. 2, NC in Tab. 3, and F-Score with thresholds of 0.002 and 0.004 in Tab. 4 and Tab. 5. Our method outperforms previous methods over most classes. The visualization comparisons in Fig. 8 shows that our method can reconstruct a smoother surface with fine details.

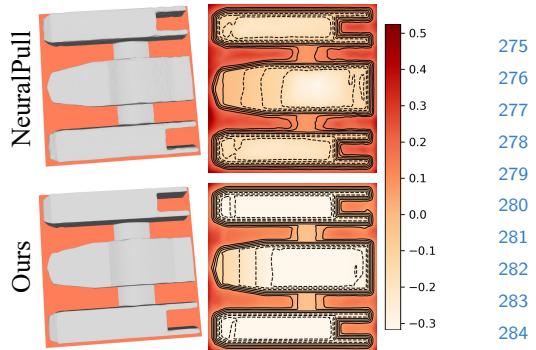


Fig. 7: Visualization of level sets on a cross section.

4.2 Surface Reconstruction for Real Scans

Dataset and metrics. For surface reconstruction of real point cloud scans, we follow VisCo [32] to evaluate our method under the Surface Reconstruction

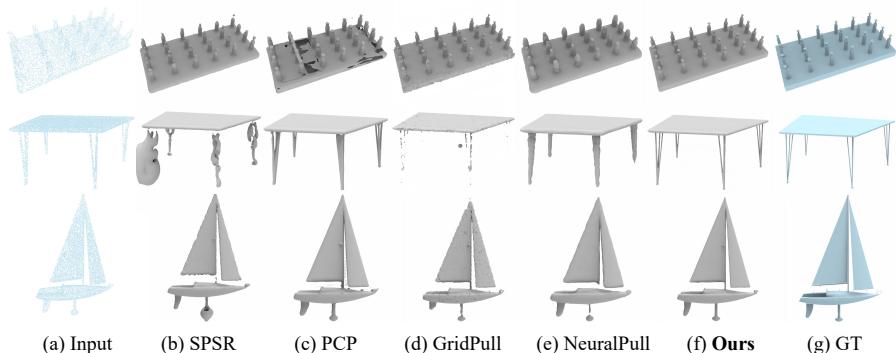


Fig. 8: Visual comparisons of surface reconstruction on ShapeNet dataset.

Benchmarks (SRB) [39]. We use Chamfer and Hausdorff distances (CD_{L1} and HD) between the reconstruction meshes and the ground truth. Furthermore, we report their corresponding one-sided distances ($d_{\overrightarrow{C}}$ and $d_{\overrightarrow{H}}$) between the reconstructed meshes and the input noisy point cloud.

Comparisons. We compare our method with state-of-the-art methods under the real scanned SRB dataset, including IGR [13], SPSR [17], Shape As Points (SAP) [29], NeuralPull (NP) [3], and GridPull (GP) [10]. The numerical comparisons are shown in Tab. 6, where we achieve the best accuracy in most cases. The visual comparisons in Fig. 9 demonstrate that our method can reconstruct a continuous and smooth surface with geometry details.

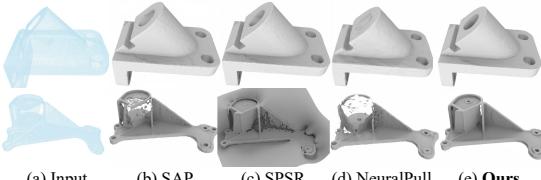


Fig. 9: Visual comparisons on SRB dataset.

4.3 Surface Reconstruction for Scenes

Dataset and metrics. To further demonstrate the advantage of our method in the surface reconstruction of real scene scans, we conduct experiments using the 3D Scene dataset. The 3D Scene dataset is a challenging real-world dataset with complex topology and noisy open surfaces. We uniformly sample 1000 points per m^2 of each scene as the input and follow PCP [4] to sample 1M points on both the reconstructed and the ground truth surfaces. We leverage L1 and L2 Chamfer distance (CD_{L1}, CD_{L2}) and normal consistency (NC) to evaluate the reconstruction quality.

Comparisons. We compare our method with the state-of-the-art methods ConvONet [30], LIG [16], DeepLS [7], NeuralPull (NP) [3], PCP [4], GridPull

Table 2: Comparisons on ShapeNet dataset in terms of $CD_{L2} \times 100$.

Class	SPSR [17]	NP [3]	LPI [9]	PCP [4]	GP [10]	Ours
Display	0.273	0.039	0.0080	0.0087	0.0082	0.0009
Lamp	0.227	0.080	0.0172	0.0380	0.0347	0.0019
Airplane	0.217	0.008	0.0060	0.0065	0.0007	0.0045
Cabinet	0.363	0.026	0.0179	0.0153	0.0112	0.0055
Vessel	0.254	0.022	0.0092	0.0079	0.0033	0.0005
Table	0.383	0.060	0.0436	0.0131	0.0052	0.0025
Chair	0.293	0.054	0.0187	0.0110	0.0043	0.0070
Sofa	0.276	0.012	0.0164	0.0086	0.0015	0.0027
Mean	0.286	0.038	0.0171	0.0136	0.0086	0.0032

Table 3: Comparisons on ShapeNet dataset in terms of NC.

Class	SPSR [17]	NP [3]	LPI [9]	PCP [4]	GP [10]	Ours
Display	0.889	0.964	0.9780	0.9775	0.9847	0.9880
Lamp	0.876	0.930	0.9503	0.9450	0.9693	0.9692
Airplane	0.848	0.947	0.9560	0.9490	0.9614	0.9800
Cabinet	0.880	0.930	0.9576	0.9600	0.9689	0.9711
Vessel	0.861	0.941	0.9564	0.9546	0.9667	0.9802
Table	0.833	0.908	0.9527	0.9595	0.9755	0.9780
Chair	0.850	0.937	0.9545	0.9580	0.9733	0.9740
Sofa	0.892	0.951	0.9713	0.9680	0.9792	0.9829
Mean	0.866	0.939	0.9596	0.9590	0.9723	0.9779

(GP) [10]. The numerical comparisons in Tab. 7 demonstrate our superior performance in all scenes even compared with the local-based methods. We further present visual comparisons in Fig. 10. The visualization further shows that our method can achieve smoother with high-fidelity surfaces in complex scenes. It should be noted that the surface we extract here is not the zero level set but the 0.001 level set since the scene is not watertight. For NeuralPull we use the threshold of 0.005 instead of 0.001 to extract the complete surface therefore the mesh looks thicker.

4.4 Ablation Studies

We conduct ablation studies on the FAMOUS dataset to demonstrate the effectiveness of our proposed implicit filter and explore the effect of some important hyperparameters. We report the performance in terms of L1 and L2 Chamfer distance ($CD_{L1}, CD_{L2} \times 10^3$), normal consistency (NC), and F-Score (F-S.).

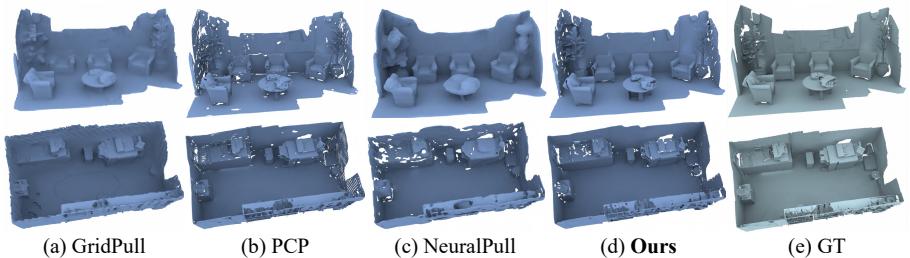
Effect of level set filtering. To justify the effectiveness of each term in our loss function. We report the results trained by different combinations in Tab. 8. The CD loss function is more applicable for training SDF from raw point clouds.

Table 4: Comparisons on ShapeNet dataset in terms of F-Score with a threshold of 0.002.

Class	SPSR [17]	NP [3]	LPI [9]	PCP [4]	GP [10]	Ours
Display	0.468	0.989	0.9978	0.9939	0.9963	0.9998
Lamp	0.455	0.891	0.9889	0.9382	0.9455	0.9986
Airplane	0.415	0.996	0.9989	0.9942	0.9976	0.9959
Cabinet	0.392	0.980	0.9849	0.9888	0.9901	0.9955
Vessel	0.415	0.985	0.9955	0.9935	0.9956	0.9998
Table	0.233	0.922	0.9789	0.9969	0.9977	0.9986
Chair	0.382	0.954	0.98977	0.9970	0.9979	0.9952
Sofa	0.499	0.968	0.9946	0.9943	0.9974	0.9973
Mean	0.407	0.961	0.9912	0.9871	0.9896	0.9976

Table 5: Comparisons on ShapeNet dataset in terms of F-Score with a threshold of 0.004.

Class	SPSR [17]	NP [3]	LPI [9]	PCP [4]	GP [10]	Ours
Display	0.666	0.991	0.9993	0.9958	0.9963	0.9971
Lamp	0.648	0.924	0.9954	0.9402	0.9538	0.9968
Airplane	0.619	0.997	0.9998	0.9972	0.9989	0.9966
Cabinet	0.598	0.989	0.9938	0.9939	0.9946	0.9999
Vessel	0.633	0.99	0.9985	0.9958	0.9972	0.9994
Table	0.442	0.973	0.9866	0.9985	0.9990	0.9988
Chair	0.617	0.969	0.994	0.9991	0.9990	0.9995
Sofa	0.725	0.974	0.9982	0.9987	0.9992	0.9999
Mean	0.618	0.976	0.9957	0.9899	0.9923	0.9985

**Fig. 10:** Visual comparisons of surface reconstruction on 3D Scene dataset.

The zero-level filter can help remove the noise and keep the geometric features. Filtering across non-zero level sets can improve the overall consistency of the entire signed distance field. Since we assume all input points lie on the surface, the function L_{dist} is also necessary. Fig. 11 shows a 2D comparison of these losses,

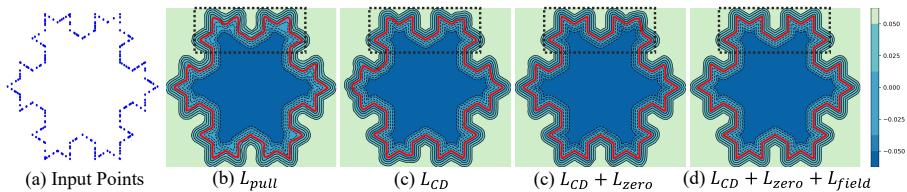
340
341
342
343

Table 6: Comparisons on SRB dataset.

		SPSR [17]	IGR [13]	SIREN [33]	VisCo [32]	SAP [29]	NP [3]	GP [10]	Ours
Anchor	CD_{L1}	0.60	0.22	0.32	0.21	0.12	0.122	0.093	0.052
	HD	14.89	4.71	8.19	3.00	2.38	3.243	1.804	1.232
	$d_{\vec{C}}$	0.60	0.12	0.10	0.15	0.08	0.061	0.066	0.025
	$d_{\vec{H}}$	14.89	1.32	2.432	1.07	0.83	3.208	0.460	0.265
Daratech	CD_{L1}	0.44	0.25	0.21	0.21	0.26	0.375	0.062	0.051
	HD	7.24	4.01	4.30	4.06	0.87	3.127	0.648	0.751
	$d_{\vec{C}}$	0.44	0.08	0.09	0.14	0.04	0.746	0.039	0.028
	$d_{\vec{H}}$	7.24	1.59	1.77	1.76	0.41	3.267	0.293	0.423
DC	CD_{L1}	0.27	0.17	0.15	0.15	0.07	0.157	0.066	0.041
	HD	3.10	2.22	2.18	2.22	1.17	3.541	1.103	0.815
	$d_{\vec{C}}$	0.27	0.09	0.06	0.09	0.04	0.242	0.036	0.019
	$d_{\vec{H}}$	3.10	2.61	2.76	2.76	0.53	3.523	0.539	0.724
Gargoyle	CD_{L1}	0.26	0.16	0.17	0.17	0.07	0.080	0.063	0.044
	HD	6.80	3.52	4.64	4.40	1.49	1.376	1.129	1.089
	$d_{\vec{C}}$	0.26	0.06	0.08	0.11	0.05	0.063	0.045	0.022
	$d_{\vec{H}}$	6.80	0.81	0.91	0.96	0.78	0.475	0.700	0.246
Lord Quas	CD_{L1}	0.20	0.12	0.17	0.12	0.05	0.064	0.047	0.030
	HD	4.61	1.17	0.82	1.06	0.98	0.822	0.569	0.554
	$d_{\vec{C}}$	0.20	0.07	0.12	0.07	0.04	0.053	0.031	0.014
	$d_{\vec{H}}$	4.61	0.98	0.76	0.64	0.51	0.508	0.370	0.230

Table 7: Comparisons on 3D Scene dataset, $CD_{L2} \times 1000$.

	Burgers			Loung			Copyroom			Stonewall			Totempole		
	CD_{L2}	CD_{L1}	NC												
ConvOnNet [30]	27.46	0.079	0.907	9.54	0.046	0.894	10.97	0.045	0.892	20.46	0.069	0.905	2.054	0.021	0.943
LIG [16]	3.055	0.045	0.835	9.672	0.056	0.833	3.61	0.036	0.810	5.032	0.042	0.879	9.58	0.062	0.887
DeepLS [7]	0.401	0.017	0.920	6.103	0.053	0.848	0.609	0.021	0.901	0.320	0.015	0.954	0.601	0.017	0.950
GP [10]	1.367	0.028	0.873	4.684	0.053	0.827	2.327	0.030	0.857	2.234	0.024	0.913	2.278	0.034	0.878
PCP [4]	1.339	0.031	0.929	0.432	0.014	0.934	0.405	0.014	0.914	0.266	0.014	0.957	1.089	0.029	0.954
NP [3]	0.897	0.025	0.883	0.855	0.022	0.887	0.479	0.018	0.862	0.434	0.018	0.929	1.604	0.032	0.923
Ours	0.133	0.011	0.934	0.120	0.008	0.926	0.111	0.009	0.913	0.082	0.009	0.957	0.203	0.013	0.944

**Fig. 11:** The 2D level sets show the distance field learned by different losses. The red lines represent the learned zero level set.

showing that our filter loss functions can reconstruct a field that is aligned at all level sets and maintains geometric characteristics.

Effect of the bidirectional projection. To validate our bidirectional projection distance, we report the results in Tab. 9. The numerical comparisons show that projecting the distance to both normals can improve the reconstruction

349
350

quality. Note that the effect of using $d(\bar{p})$ is also much better than the methods without our filter.

349
350**Table 8:** Effect of different losses.

Loss	CD_{L1}	CD_{L2}	F-S.	NC
L_{pull}	0.012	0.083	0.742	0.884
L_{CD}	0.010	0.031	0.757	0.891
$L_{CD} + L_{zero}$	0.008	0.018	0.772	0.905
$L_{CD} + L_{zero} + L_{field}$	0.008	0.011	0.769	0.908
Ours	0.007	0.008	0.778	0.911

Table 9: Effect of bidirectional projection.

	$d(\bar{p})$	$d_{bi}(\bar{p})$
CD_{L1}	0.010	0.007
CD_{L2}	0.024	0.008
F-S.	0.726	0.778
NC	0.890	0.911

Weight of level set projection loss. We explore the effect of the Chamfer distance loss function by adjusting the weight α_3 in Eq. (11). We report our results with different candidates $\{0, 1, 10\}$ in Tab. 10, where 0 means we do not use the L_{CD} to constraint the gradient. The comparisons in Tab. 10 show that although our implicit filter can directly learn SDFs, it is better to adopt the L_{CD} for a more stable field. However, if the weight is too large, the filtering effect will decrease. It is recommended to select weights ranging from 1 to 10, which is usually adequate. For the weights α_1 and α_2 , setting them to 1 is always necessary.

Effect of filter parameters. We compare the effect of different parameters σ_n, σ_p in the bilateral filtering in Tab. 11. The diagonal weight for σ_p means the length of the diagonal of the bounding box for the local patch mentioned in [42]. The results indicate that the method is relatively robust to parameter variation within a certain range.

Table 10: Effect of weight α_3 .

α_3	CD_{L1}	CD_{L2}	F-S.	NC
0	0.008	0.013	0.758	0.903
1	0.007	0.011	0.772	0.910
10	0.007	0.008	0.778	0.911
100	0.008	0.009	0.774	0.909

Table 11: Effect of filter parameters σ_n and σ_p .

		CD_{L1}	CD_{L2}	F-S.	NC
σ_n	15°	0.007	0.008	0.778	0.911
	30°	0.007	0.011	0.771	0.907
	45°	0.008	0.012	0.764	0.903
	60°	0.008	0.010	0.767	0.901
σ_p	max	0.007	0.008	0.778	0.911
	diagonal	0.008	0.011	0.763	0.904

365 5 Conclusion

366 We introduce implicit filtering on SDFs to reduce the noise of the signed distance
 367 field while preserving geometry features. We filter the distance field by minimizing
 368 the weighted bidirectional projection distance, where the sample points on
 369 the zero level set and neighbor points on non-zero level sets can be generated
 370 by the pulling procedure. By leveraging the Chamfer distance loss function, we
 371 address the issue of gradient degeneration in our filter operator. The visual and
 372 numerical comparisons demonstrate our effectiveness and superiority over state-
 373 of-the-art methods in SDF inference.

366
367
368
369
370
371
372
373

374

References

374

- Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) **1, 3, 8**
- Atzmon, M., Lipman, Y.: SALD: sign agnostic learning with derivatives. In: 9th International Conference on Learning Representations, ICLR 2021 (2021) **1, 3**
- Baorui, M., Han, Z., Liu, Y., Zwicker, M.: Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. Cornell University - arXiv,Cornell University - arXiv (Nov 2020) **1, 3, 6, 7, 8, 9, 10, 11, 12, 13**
- Baorui, M., Yu-Shen, L., Matthias, Z., Zhizhong, H.: Surface reconstruction from point clouds by learning predictive context priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) **3, 8, 9, 10, 11, 12, 13**
- Ben-Shabat, Y., Hewa Koneputugodage, C., Gould, S.: Digs: Divergence guided shape implicit neural representation for unoriented point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19323–19332 (2022) **1, 3, 8, 9**
- Boulch, A., Marlet, R.: Poco: Point convolution for surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6302–6314 (June 2022) **3**
- Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. pp. 608–625. Springer (2020) **3, 10, 13**
- Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015) **8**
- Chao, C., Yu-shen, L., Zhizhong, H.: Latent partition implicit with surface codes for 3d representation. In: European Conference on Computer Vision (ECCV) (2022) **9, 11, 12**
- Chen, C., Liu, Y.S., Han, Z.: Gridpull: Towards scalability in learning implicit representations from 3d point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2023) **1, 3, 8, 9, 10, 11, 12, 13**
- Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J., Wimmer, M.: Points2Surf: Learning implicit surfaces from point clouds. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 108–124. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_7 **2, 3, 8, 9**
- Fleishman, S., Cohen-Or, D., Silva, C.T.: Robust moving least-squares fitting with sharp features. ACM transactions on graphics (TOG) **24**(3), 544–552 (2005) **3**
- Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. Cornell University - arXiv,Cornell University - arXiv (Feb 2020) **1, 8, 9, 10, 13**
- Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. ACM transactions on graphics (TOG) **28**(5), 1–7 (2009) **3**

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

- 424 15. Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.: Edge-aware
425 point set resampling. ACM transactions on graphics (TOG) **32**(1), 1–12 (2013) 3, 5 424
426
427 16. Jiang, C.M., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local
428 implicit grid representations for 3d scenes. In: Proceedings IEEE Conf. on Computer
429 Vision and Pattern Recognition (CVPR) (2020) 3, 10, 13 425
430
431 17. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions
432 on Graphics (ToG) **32**(3), 1–13 (2013) 9, 10, 11, 12, 13 426
433
434 18. Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa,
435 M., Zorin, D., Panozzo, D.: Abc: A big cad model dataset for geometric deep
436 learning. In: Proceedings of the IEEE/CVF conference on computer vision and
437 pattern recognition. pp. 9601–9611 (2019) 8 427
438
439 19. Li, S., Gao, G., Liu, Y., Liu, Y.S., Gu, M.: Gridformer: Point-grid transformer
440 for surface reconstruction. In: Proceedings of the AAAI Conference on Artificial
441 Intelligence (2024) 3 428
442
443 20. Liao, B., Xiao, C., Jin, L., Fu, H.: Efficient feature-preserving local projection
444 operator for geometry reconstruction. Computer-Aided Design **45**(5), 861–
445 874 (2013). <https://doi.org/https://doi.org/10.1016/j.cad.2013.02.003>, 439
<https://www.sciencedirect.com/science/article/pii/S0010448513000341> 3 440
446
447 21. Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H.: Parameterization-free projection
448 for geometry reconstruction. ACM Transactions on Graphics (TOG) **26**(3),
449 22–es (2007) 3 441
450
451 22. Liu, S.L., Guo, H.X., Pan, H., Wang, P.S., Tong, X., Liu, Y.: Deep implicit moving
452 least-squares functions for 3d reconstruction. In: Proceedings of the IEEE/CVF
453 Conference on Computer Vision and Pattern Recognition. pp. 1788–1797 (2021) 3 442
454
455 23. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface
456 construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field,
457 pp. 347–353 (1998) 1, 4, 5 443
458
459 24. Ma, B., Liu, Y.S., Han, Z.: Reconstructing surfaces for sparse point clouds with
460 on-surface priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision
461 and Pattern Recognition. pp. 6315–6325 (2022) 3 444
462
463 25. Ma, B., Zhou, J., Liu, Y.S., Han, Z.: Towards better gradient consistency for neural
464 signed distance functions via level set alignment. In: Conference on Computer
465 Vision and Pattern Recognition (CVPR) (2023) 1 445
466
467 26. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy
468 networks: Learning 3d reconstruction in function space. In: Proceedings IEEE Conf.
469 on Computer Vision and Pattern Recognition (CVPR) (2019) 3, 8 446
470
471 27. Mi, Z., Luo, Y., Tao, W.: Ssrnet: Scalable 3d surface reconstruction network. In:
472 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-
473 nition. pp. 970–979 (2020) 3 447
474
475 28. Öztireli, A.C., Guennebaud, G., Gross, M.: Feature preserving point set surfaces
476 based on non-linear kernel regression. In: Computer graphics forum. vol. 28, pp.
477 493–501. Wiley Online Library (2009) 3 448
478
479 29. Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape as
480 points: A differentiable poisson solver. Advances in Neural Information Processing
481 Systems **34**, 13032–13044 (2021) 3, 10, 13 449
482
483 30. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional
484 occupancy networks. In: European Conference on Computer Vision (ECCV) (2020)
485 3, 10, 13 450
486
487 31. Preiner, R., Mattausch, O., Arikan, M., Pajarola, R., Wimmer, M.: Continuous
488 projection for fast l1 reconstruction. ACM Trans. Graph. **33**(4), 47–1 (2014) 3 451
489
490

- 475 32. Pumarola, A., Sanakoyeu, A., Yariv, L., Thabet, A., Lipman, Y.: Visco grids: Sur-
476 face reconstruction with viscosity and coarea grids. Advances in Neural Information
477 Processing Systems **35**, 18060–18071 (2022) **9, 13**
- 478 33. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural
479 representations with periodic activation functions. Neural Information Processing
480 Systems, Neural Information Processing Systems (Jun 2020) **1, 3, 8, 9, 13**
- 481 34. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Sing-
482 hal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn
483 high frequency functions in low dimensional domains. Advances in Neural Infor-
484 mation Processing Systems **33**, 7537–7547 (2020) **3**
- 485 35. Tang, J., Lei, J., Xu, D., Ma, F., Jia, K., Zhang, L.: Sa-convolnet: Sign-agnostic op-
486 timization of convolutional occupancy networks. In: Proceedings of the IEEE/CVF
487 International Conference on Computer Vision (2021) **3**
- 488 36. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Stoll, C., Theobalt, C.: Patch-
489 nets: Patch-based generalizable deep implicit 3d shape representations. In: Com-
490 puter Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28,
491 2020, Proceedings, Part XVI 16. pp. 293–309. Springer (2020) **3**
- 492 37. Wang, X., Cheng, Y., Wang, L., Lu, J., Xu, K., Xiao, G.: Edge preserving implicit
493 surface representation of point clouds. arXiv preprint arXiv:2301.04860 (2023) **3**
- 494 38. Wang, Z., Zhou, S., Park, J.J., Paschalidou, D., You, S., Wetzstein, G., Guibas,
495 L., Kadambi, A.: Alto: Alternating latent topologies for implicit 3d reconstruction.
496 arXiv preprint arXiv:2212.04096 (2022) **3**
- 497 39. Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., Panozzo, D.: Deep
498 geometric prior for surface reconstruction. In: Proceedings of the IEEE/CVF con-
499 ference on computer vision and pattern recognition. pp. 10130–10139 (2019) **10**
- 500 40. Yifan, W., Rahmann, L., Sorkine-Hornung, O.: Geometry-consistent neural shape
501 representation with implicit displacement fields. arXiv preprint arXiv:2106.05187
502 (2021) **3**
- 503 41. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Ec-net: an edge-aware point
504 set consolidation network. In: Proceedings of the European conference on computer
505 vision (ECCV). pp. 386–402 (2018) **3**
- 506 42. Zhang, D., Lu, X., Qin, H., He, Y.: Pointfilter: Point cloud filtering via encoder-
507 decoder modeling. IEEE Transactions on Visualization and Computer Graphics
508 **27**(3), 2015–2027 (2020) **3, 5, 14**
- 509 43. Zhou, J., Ma, B., Liu, Y.S., Fang, Y., Han, Z.: Learning consistency-aware unsigned
510 distance functions progressively from raw point clouds. In: Advances in Neural
511 Information Processing Systems (NeurIPS) (2022) **7**