

# ImplicitFilter

论文：《Implicit Filtering for Learning Neural Signed Distance Functions from 3D Point Clouds》

地址：<https://arxiv.org/abs/2407.13342>

年份：ECCV 2024

## Introduction

任务：表面重建

技术贡献：

(1) 提出一种应用于 SDF 的隐式滤波器，使得物体表面光滑的同时保留了几何上的细节

## Method

文章参考了图像处理中双边滤波的思想：同时考虑将要被滤波的像素点的空域信息（domain）和值域信息（range）。对于图像滤波来说，一个通常的 intuition 是：图像在空间中变化缓慢，因此相邻的像素点会更相近。但是这个假设在图像的边缘处变得不成立。如果在边缘处也用这种思路来进行滤波的话，即认为相邻相近，则得到的结果必然会模糊掉边缘，因此考虑再利用像素点的值的大小进行补充，因为边缘两侧的点的像素值差别很大，因此会使得其加权的时候权重具有很大的差别，从而使得只考虑自己所属的一边的邻域。可以理解成先根据像素值对要用来进行滤波的邻域做一个分割或分类，再给该点所属的类别相对较高的权重，然后进行邻域加权求和，得到最终结果。

对应到 3D 中，被滤波的“像素点”就是 3D 点，“空域信息”就是点的坐标，“值域信息”就是点的法向量，文章就根据这个思想构造了针对 SDF 的滤波。

记点云为  $\mathbf{P}$ ， $\mathcal{S}_d$  是 SDF 的距离为  $d$  的 level set，即  $\mathcal{S}_d = \{\mathbf{q} | f_\theta(\mathbf{q}) = d\}$

## Level set bilateral filtering

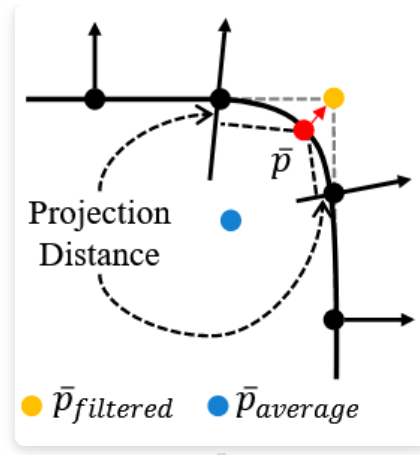
通过优化 SDF 中最基本的 loss

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

我们可以认为  $\mathbf{P}$  中所有的点都位于 level set  $\mathcal{S}_0$  上。我们先来看只考虑空域信息的高斯滤波的情况。对于  $\mathcal{S}_0$  上一点  $\bar{\mathbf{p}}$ ，其相邻点为  $\mathcal{N}(\bar{\mathbf{p}}, \mathcal{S}_0)$ ，对其应用高斯滤波：

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

其中  $\phi(\|\bar{\mathbf{p}} - \mathbf{p}_j\|) = \exp(-\frac{\|\bar{\mathbf{p}} - \mathbf{p}_j\|^2}{\sigma_p^2})$



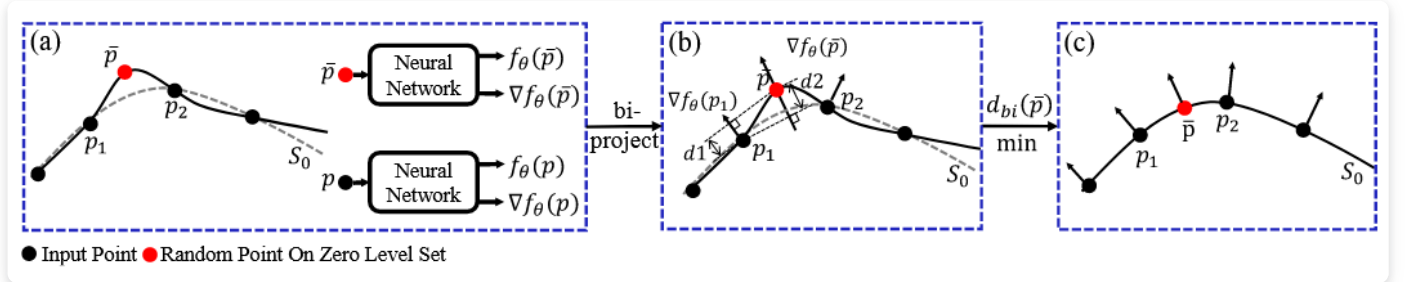
如图所示，蓝色点就是通过上述方式计算得到的，可以发现只考虑位置信息会得到错误的结果。因此，参照双边滤波的思想，引入法向量的相似性，稍作修改：

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

其中  $\psi(\mathbf{n}_{\bar{p}}, \mathbf{n}_{p_j}) = \exp(-\frac{1 - \mathbf{n}_{\bar{p}}^T \mathbf{n}_{p_j}}{1 - \cos(\sigma_n)})$ ， $|\mathbf{n}_{p_j}^T (\bar{p} - \mathbf{p}_j)|$  相当于两点距离在梯度方向上的投影，我们通过最小化上述距离，就可以得到滤波后的  $\bar{p}$ 。

作者参考了另一篇文章，在上式中再添加了一项，可以得到更好地效果：

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



作者提到，上述方法在 point cloud denoising and resampling 中已经有用到，这篇文章主要的贡献是解决了将其应用于 SDF 的 2 个困难：(1) 上面的方法要求在 level set  $S_0$  上进行采样点来进行计算，这对于 SDF 来说是困难的；(2) 上述式子的法向量是通过计算网络的梯度得到的，因此最小化上式可能会出现  $\nabla f_\theta = 0$  的解的情况，因此要对梯度加以限制。文章后半部分就是来解决这两个问题。

## Sampling points for filtering

对于第一个问题，可以参考 Neural-Pull 的做法，将采样到的其他 level set 上的点“拉”到  $S_0$  上：

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

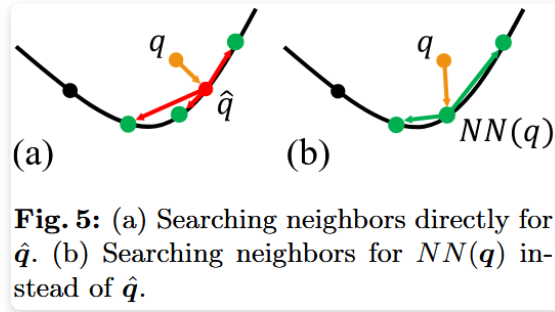
因此可以去优化以下 loss 达到前半部分所说的：

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

上面这个过程可以描述为“有了邻居点去找采样的点”，我们还可以做反过来的事情：“有了采样点去找邻居点”，就是将  $P$  中的点“拉”到  $\mathcal{S}_{f_\theta(q)}$  上，作为采样点  $q$  的邻居点。这样就可以优化整个 SDF 场：

$$L_{field} = \sum_{q \in Q} d_{bi}(q)$$

最后是对计算效率和准确度之间做了一个 trade-off：对于固定的采样点  $q$ ，“拉”到  $\mathcal{S}_0$  上的点  $\hat{q}$  会随着优化过程动态变换，因此寻找其邻居点  $\mathcal{N}(\hat{q}, \mathcal{S}_0)$  会是一个费时的事情。因此通过  $P$  上离  $q$  最近点的邻居来替代： $\mathcal{N}(\text{NN}(q), \mathcal{S}_0)$ ，如下图所示。



## Gradient constraint

保证梯度不为 0 的最直接想法就是应用 Eikonal loss，使梯度模长为 1。这里作者并没有采用这样的方法（后面消融实验也证明了用下面讲的 loss 会更好）。可以直接使用 Neural-Pull 中的 loss，因为只要保证“拉”这个操作是准确的，梯度自然就是准确的了。

因此可以采用：

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

但文章采用的是另一种 loss，是 CAP-UDF 中使用的最小化点云和 moved query points 之间的 Chamfer distance：

$$L_{CD} = \frac{1}{M} \sum_{i \in [1, M]} \min_{j \in [1, N]} \|\hat{q}_i - p_j\|_2 + \frac{1}{N} \sum_{j \in [1, N]} \min_{i \in [1, M]} \|p_j - \hat{q}_i\|_2$$

最终的 loss 就是之前提到的几个 loss 的组合：

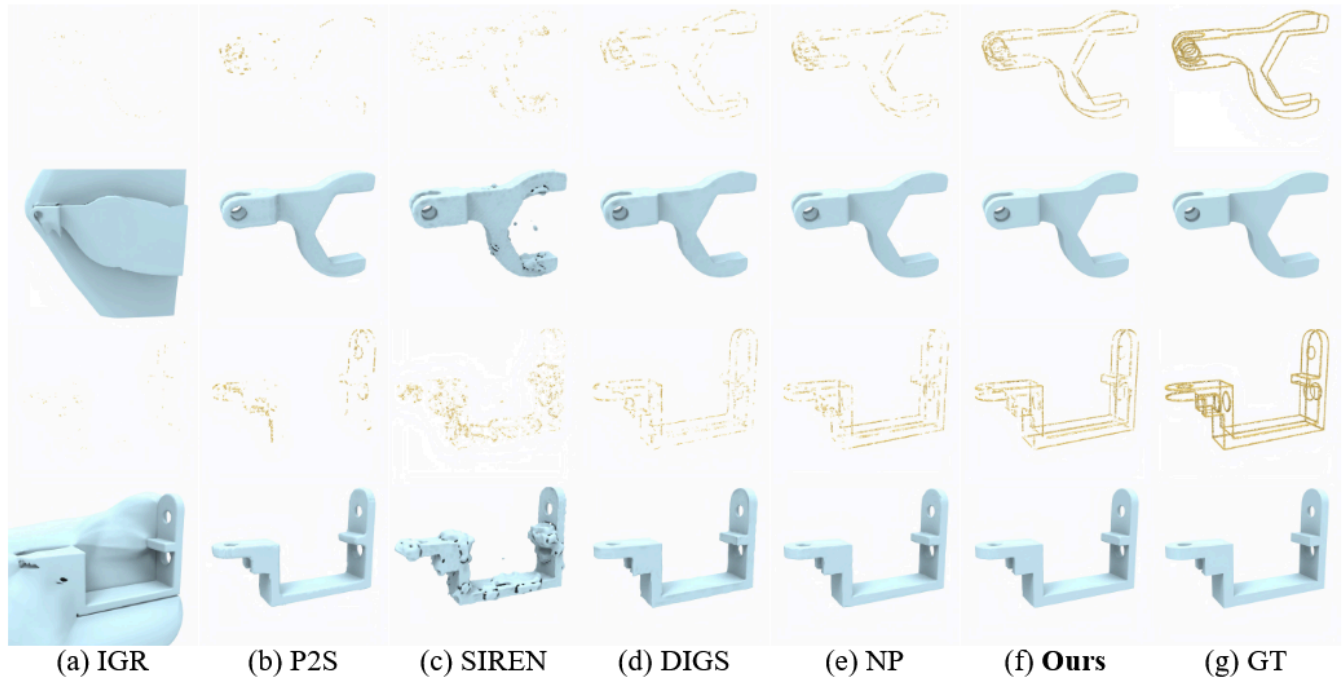
$$L = L_{zero} + \alpha_1 L_{field} + \alpha_2 L_{dist} + \alpha_3 L_{CD}$$

## Experiments

使用的数据集：ABC、FAMOUS、ShapeNet

评估指标：L1 and L2 Chamfer distance、normal consistency、F-score

实验中有一个计算 edge point 的可视化结果



**Fig. 9:** Visual comparisons of edge points and reconstruction results.

## Ablation Studies

### Effect of Eikonal loss

**Table 6:** Effect of the Eikonal term.

Loss	$CD_{L1}$	$CD_{L2}$	F-S.	NC
w/ Eikonal, w/o CD	0.009	0.021	0.738	0.899
w/ Eikonal, w/ CD	0.008	0.009	0.774	0.910
w/o Eikonal, w/ CD	<b>0.007</b>	<b>0.008</b>	<b>0.778</b>	<b>0.911</b>

测试了使用 Eikonal loss 来约束梯度的效果，不如使用  $L_{CD}$  的，我觉得可能的原因是 Eikonal loss 仅仅约束了梯度的模长，而  $L_{CD}$  则更具体一些，对点进行约束，显式地优化了 SDF 值，隐式优化了梯度，因此从几何上会更好一些。

### Effect of different losses

**Table 7:** 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	<b>0.007</b>	<b>0.008</b>	<b>0.778</b>	<b>0.911</b>

### Effect of the bidirectional projection

**Table 8:** Effect of bidirectional projection.

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