

复现论文题目：《Iterative Poisson Surface Reconstruction (iPSR) for Unoriented Points》

1. 研究背景

随着三维点云数据处理技术的发展，表面重建已经成为计算机图形学和计算机视觉中的一个重要研究方向。表面重建旨在通过一组离散的采样点来推断出一个三维物体的表面。简单来说，表面重建是从点云数据中恢复物体的几何形状的过程，这些点通常来自扫描、传感器或计算机生成的三维模型，目标是生成一个光滑、连贯的表面，能够准确地反映物体的形状。表面重建广泛应用于三维建模、虚拟现实、医学影像、计算机视觉等领域。

近年来，已经有很多研究者在表面重建的任务上提了很多工作，这些工作典型的分类是将方法分类为显式表面重建和隐式表面重建。隐式方法可以保证生成的表面的水密性，尽管生成的表面不一定会经过采样点，但是对噪声的输入更加鲁棒，在表面重建的领域备受研究者的青睐。泊松表面重建 (PSR) [1] 是一个经典的隐式表面重建方法，其关键思想是通过找到一个隐式函数最佳地表示指示函数。假定一个带有定向法线的点云，点云上每个点的法线方向与指示函数的梯度一致，那么就可以将问题转化为泊松方程，通过求解泊松方程计算出隐式函数，从而得到一个稀疏的线性系统，生成水密的三维表面。然而，PSR 的方法强烈依赖于输入点云的法线方向，而在实际应用中，获取准确的法线方向仍然是一个无法解决的挑战，因此 PSR 的应用范围受到显著的限制。

在早期的工作当中，有一些显式重建的方法不需要依赖于点云的法线信息，而是可以从点的位置信息直接重构出表面。例如，Ball Pivoting[2]方法基于点云中点之间的几何关系，通过一个球体在点云数据中滚动，逐步建立起表面网格；Power Crust[3]方法通过在点云上构建一种称为“功率图”的几何构造，来重建点云的表面；Tight Cocone[4]方法通过最小化表面重建中点云的误差和噪声影响，从而构建一个紧致的三角网格。然而这些方法都直接依赖于 Delaunay 三角剖分或其对偶的 Voronoi 图，生成的网格直接依赖于原始点云中点的位置，因此对噪声和离群点非常敏感，且无法在稀疏点云下有效地工作。

为了克服法线方向的问题，也有一些研究者研究如何从无方向的点云中自动推断出法线方向。一个经典的计算法线方向的方法是主成分分析方法 (PCA)，然而 PCA 方法不能保证在表面上的法线方向具有一致性。Hoppe[5]等人通过利用相邻的点的法线方向平行的假设来传播法线以保证法线方向的一致性，然而这是一个基于最小生成树的贪婪算法，对邻域的大小非常敏感，如果邻域过小，法线方向容易受到局部噪声或异常点的影响，而过大的邻域导致法线传播时可能考虑了跨越表面曲率变化较大的区域，从而引入误差。后来，有一些学者[6],[7]提出了一些更可靠的传播策略用于法线方向传递，但这些方法本质上仍是贪婪的。

与基于局部方法相比，全局方法在点云法线方向一致性的保持上更具有优势。Alliez[8]等人在从 Voronoi 图中估计法线方向后，通过求解广义特征值问题来评估隐式场并定向法线，最大化各向异性 Dirichlet 能量。Mullen 等人[9]首先计算输入数据的无符号距离近似，然后通过最小化二次能量估计符号，该方法对噪声和离群点具有鲁棒性。Schertler 等人[10]提出了基于全局图优化的法线方向推断方法。大多数方法将法线方向的估计分解为两个步骤，而 Wang 等人[11]提出了一个将这两步结合在一起的变分框架。

Huang 等人[12]通过 Duchon 能量拟合表面和法线，这在稀疏均匀采样和线框样本中表现良好，但由于计算成本高，无法重建密集点云。Peng 等人[13]利用基于快速傅里叶变换的泊松求解器对指示函数进行微分，通过优化点位置和法线来最小化重建误差。Xu 等人[14]通过引入一个绕数场的正则化项，通过最小化正则化能量来同时估计和定向法线方向，确保整个点云的法线方向在全局范围内的一致性。与之前的方法相比，这种方法更可靠，因为它在统一框架中同时优化点位置和法线。然而，该方法需要数千次迭代才能收敛，因此在大规模模型重建中速度过慢。

在这篇论文中，作者提出一种无需点云定向信息的改进版泊松表面重建方法——迭代泊松表面重建 (iPSR) [15]。该方法通过迭代过程逐步优化点云的法线方向，并最终生成高质量的水密表面。通过在每次迭代中利用前一轮迭代结果计算法线，iPSR 能够在没有准确法线方向的情况下，逐步修正点云的方向信息，从而有效提升表面重建质量。iPSR 方法不仅能够避免对定向信息的强依赖，也能够适用于大规模、噪声较多、非均匀分布的点云数据，为三维表面重建问题提供了一种新的解决思路。

2. 方法框架

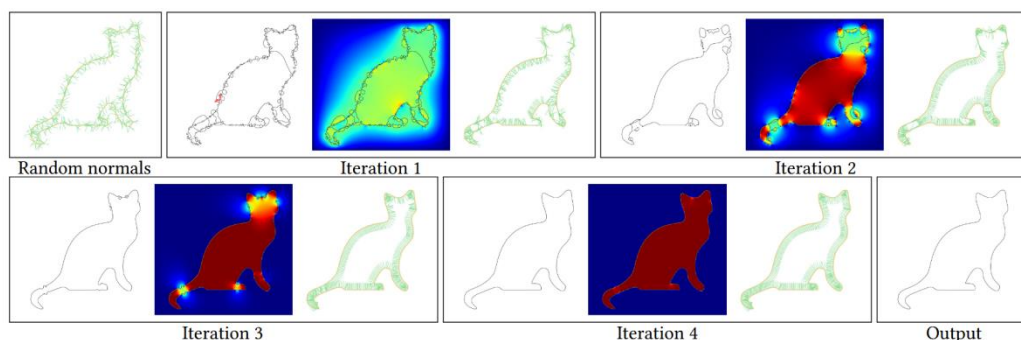


图 1 二维模型上的迭代过程

这篇论文的方法以一个无方向的点集 P 作为输入，目标是输出一个近似点云的水密网格。首先构建一个最大深度为 D 的八叉树，用于泊松重建，同时为了提高查找的效率，对所有八叉树的节点构建一棵 KD 树，并对每个节点分配一个随机的初始法向量。然后方法开始迭代，在每次的迭代中，对八叉树的所有节点应用屏蔽泊松表面重建 (SPSR) [16] 的方法，得到一个指示函数 χ ，之后通过 Marching Cube 的方法从指示函数中提取等值面，获取网格。然后，对于每个三角面片，使用叉积计算每个面片的法向量，并计算这个三角面片的面积，同时通过 KD 树查找 k 个最近的八叉树节点产生的三角面片。最后通过计算每个三角面片及其关联面片的法线的平均来更新法线。迭代终止条件是通过对比当前迭代与前一次迭代之间样本法线的方差。当法线方差的平均值小于用户指定的阈值 δ 时，算法终止。

Algorithm 1: iPSR

input : Unoriented points $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$, the maximum octree depth D , the convergence threshold δ and the screened PSR weight α

output : A watertight surface approximating the points

construct an octree of depth D to discretize \mathcal{P} ;
 $\mathcal{S} \leftarrow$ nodes of the octree;
 $n \leftarrow |\mathcal{S}|$;
construct a kd-tree for \mathcal{S} for nearest sample searching;
initialize normal for each sample $s_i \in \mathcal{S}$ randomly;
while *not convergence or exceeding maximum number of iterations* **do**
 compute indicator function χ by applying screened PSR with parameter α to \mathcal{S} ;
 extract the iso-surface \mathcal{F} with iso-value $\frac{1}{n} \sum_{j=1}^n \chi(s_j)$ by marching cubes;
 for each sample $s_j \in \mathcal{S}$ **do**
 $s_j.\text{face_list} \leftarrow \emptyset$
 end
 for each triangular face $f_j \in \mathcal{F}$ **do**
 $\vec{n}_j \leftarrow$ f_j 's inward normal, i.e., towards interior of the component f_j belonging to;
 $a_j \leftarrow$ the area of triangle f_j ;
 use kd-tree to find top- k samples in \mathcal{S} that are closest to f_j ;
 add f_j to the associated face list of each of the k samples;
 end
 for each sample $s_j \in \mathcal{S}$ **do**
 $\vec{n}(s_j) \leftarrow \sum_{i=1}^{m_j} a_{j_i} \vec{n}_{j_i}$; $/* m_j = s_j.\text{face_list.count()} */$
 normalize $\vec{n}(s_j)$;
 end
 $d \leftarrow$ the average of top 0.1% normal difference of samples between the previous and the current iterations;
 if $d < \delta$ **then**
 $\text{convergence} \leftarrow \text{true}$
 end
end
apply screened PSR to the sample points $\{s_i\}$ with predicted normals $\vec{n}(s_i)$ and return the iso-surface \mathcal{F} extracted from the computed indicator function;

3. 实验结果

3.1 论文中的实验结果

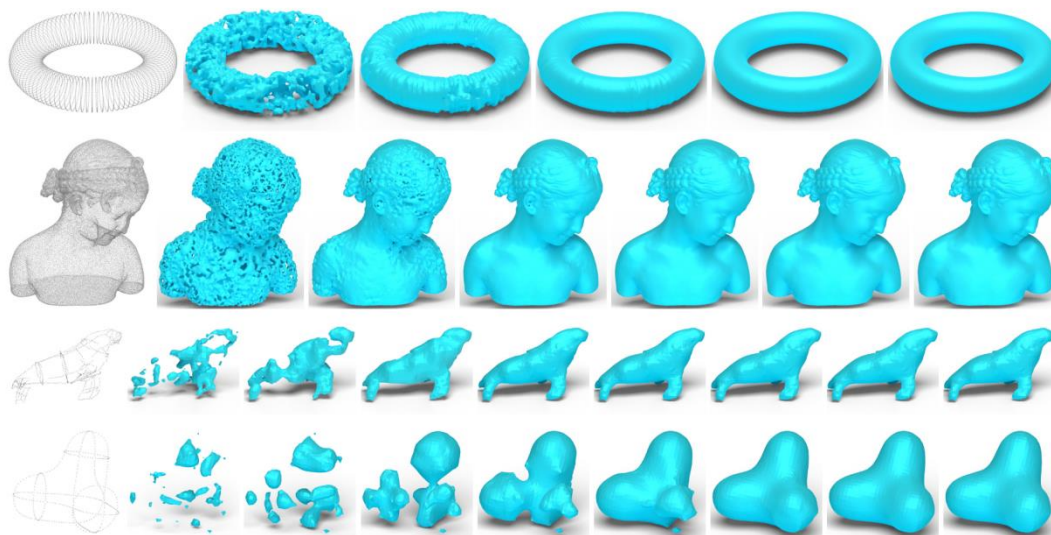


图 2 模型迭代过程和最后效果

在论文中，作者展示了一些模型的迭代过程和最后的模型效果（图 2）来说明 iPSR 方法在表面重建上的应用。

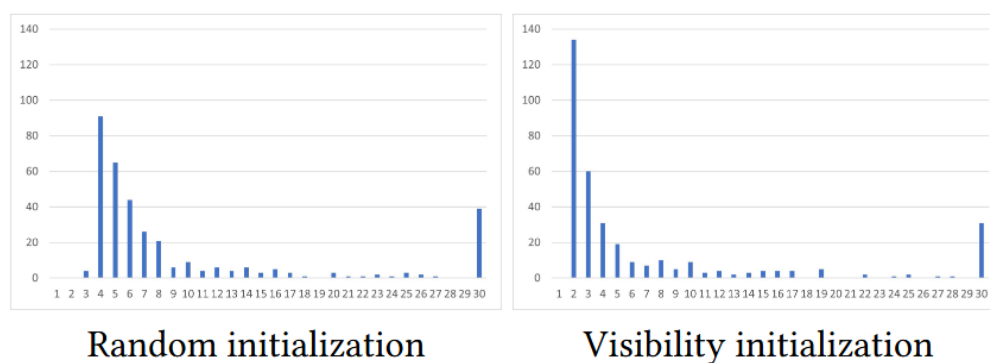


图 3 在 AIM@SHAPE 数据集上迭代收敛直方图

作者首先通过在 AIM@SHAPE 数据集来验证 iPSR 方法的有效性。结果表明，随机初始化的法向量的点云模型在 iPSR 方法中能够快速收敛，58%的模型在不超过 6 次迭代后就能达到收敛。中位数迭代次数为 6。然而，仍然还有 39 个模型需要 iPSR 超过 30 次迭代才能与随机初始化收敛。对于这些模型，进一步的迭代并没有带来显著的视觉改进。为了优化运行时间，作者将最大迭代次数设定为 30 次，即使收敛未完全达到，算法也会停止。

其次，作者展示了 iPSR 方法在大规模数据和结构化离散点云数据上的有效性。图 3 中的 Lucy 模型包含 1400 万个采样点，作者设置最大八叉树深度 $D = 14$ ，Lucy 的重建结果表明 iPSR 能够很好地保留重建表面的细节。图 4 展示了一个稀疏的、只有兔子结构的点云数据。iPSR 继承了屏蔽 Poisson 表面重建（screened PSR）的平滑逼近特性，因此对于结构化的兔子点云数据，能够生成平滑的补丁，然而如果输入模型是不完整的且无结构的，iPSR 无法智能地推断出缺失的形状。



图 4 Lucy 重建效果

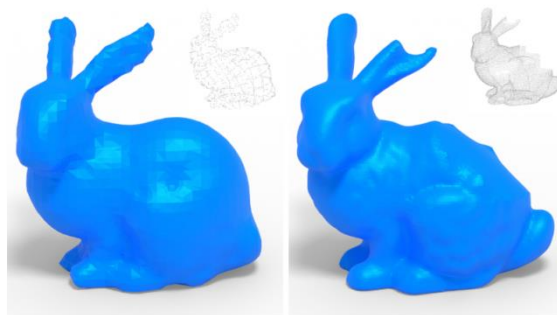


图 5 稀疏的兔子模型重建效果

作者通过对比近期的一些工作，包括对比 Dipole，可微分泊松重建 DPSR 的优化方法，以及 VIPSS 方法，来说明 iPSR 方法的效率和可扩展性。作者分别在包含 100、1K、10K、100K、1M 和 10M 个点的模型上测量了运行时间和内存消耗。从图 6 的运行时间和运行所需内存可以看出 iPSR 方法的消耗几乎是常数，可以说明 iPSR 方法的效率和可扩展性。

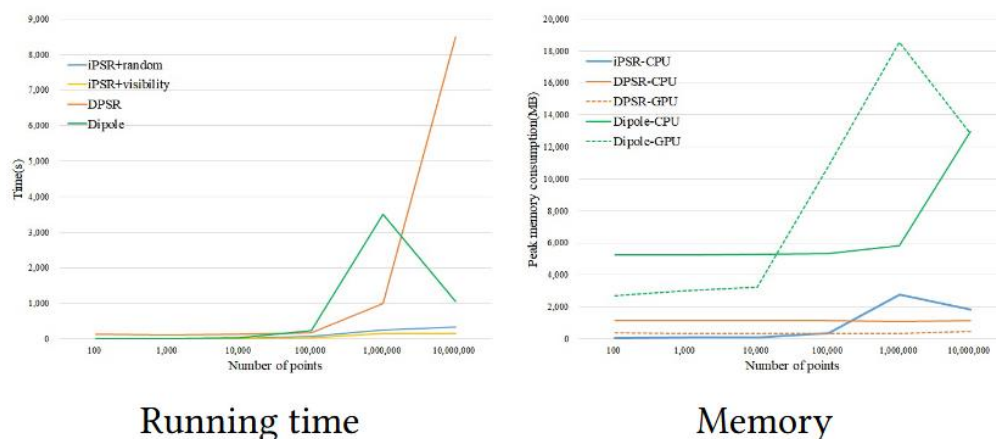


图 6 不同方法运行时间和内存消耗对比

作者也通过对比在噪声数据、高属模型、具有异常值模型、非均匀数据模型上的重建效果来说明 iPSR 方法重建表面的鲁棒性。由于 iPSR 继承了屏蔽 PSR 方法的稳健特性，iPSR 方法对噪声、异常值、高属、不均匀、缺失区域和稀疏但结构化的输入具有弹性。相比之下，其他每种方法重建出的网格模型都会有些缺点。

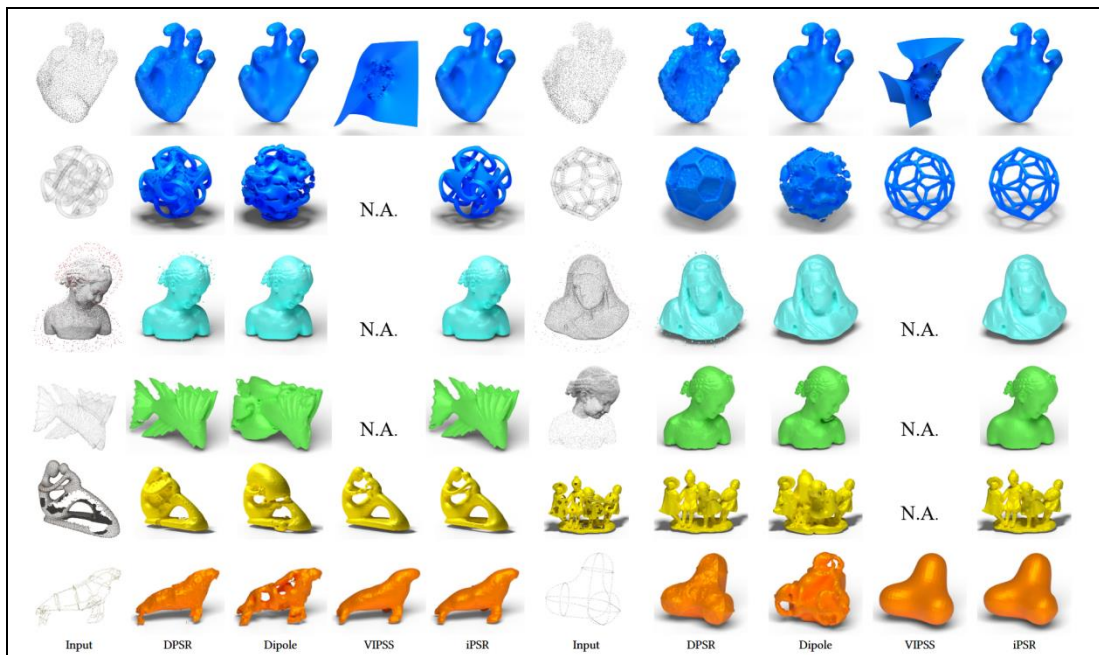


图 7. 稳健性（从上到下：噪声数据、高属模型、具有 1% 异常值的模型、非均匀点、不完整模型和稀疏点）

3.2 论文实验复现

作者已经提供了项目的源代码，链接如下：<https://github.com/houfei0801/ipsr>。在项目的复现过程，使用 Visual Studio2022 进行项目的配置。复现的目标为验证 iPSR 在其他数据集如 ABC 数据集的有效性。在本次复现中，共使用了 100 个 ABC 数据集的模型，部分表面重建的结果在图 9 中展示。图 8 中展示了这 100 个模型收敛所需要的迭代次数，对于 ABC 数据集，有 58 个模型经历 4-6 次迭代就完成收敛，这个结果与作者在 AIM@SHAPE 数据集上测试的结果一致。

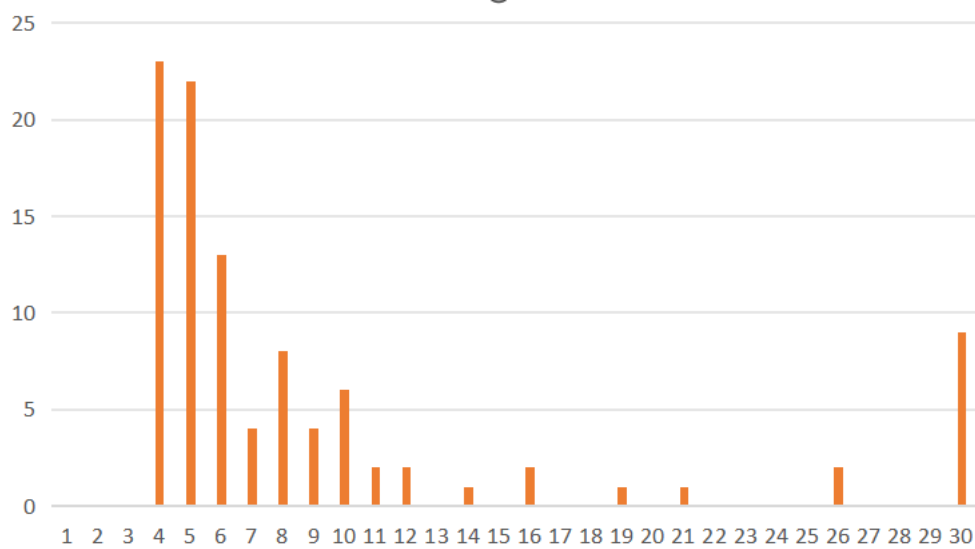


图 8 ABC 数据集部分模型收敛所需的迭代次数

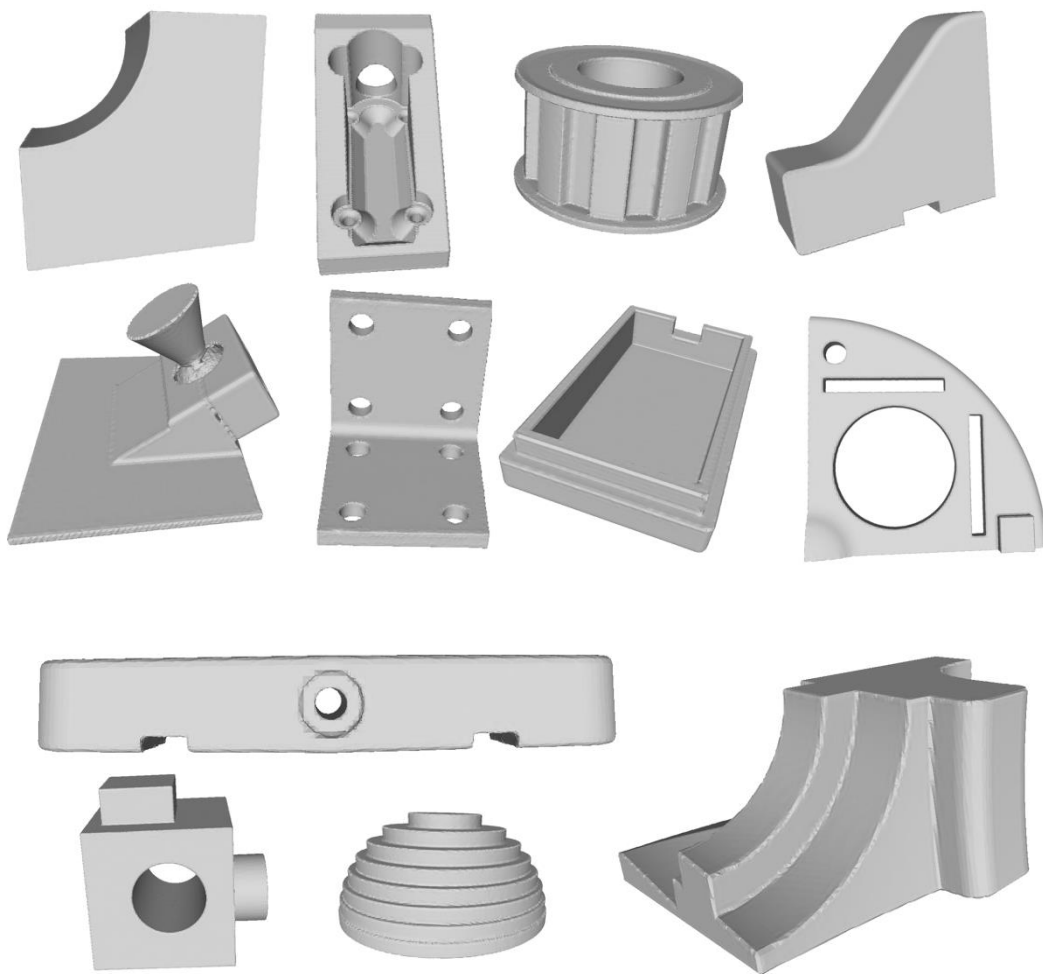


图9 复现结果

4. 讨论与改进思路

4.1 改进思路概述

PSR 方法通过求解泊松方程, 试图在整个空间中找到一个光滑且连续的解, 使系统的总能量最小化。从能量流动的角度来看, 尖锐边缘对应于局部的高频信息, 而泊松方程本质上是一个低通滤波过程, 过滤掉了这些局部高频成分, 将局部集中能量“扩散”到周围空间, 能量的平衡与扩散过程使得原本清晰的尖锐结构变得模糊, 并被过度平滑成连续的曲面。由于 PSR 的平滑特性, iPSR 也无法恢复尖锐特征。因此, 改进的思路为在 iPSR 之后引入一个后处理, 尽可能地恢复原始模型的尖锐特征。为了验证该思路的合理性, 选用了神经立方行进算法 NMC[17]对 iPSR 生成的网格进行后处理, 定性与定量地对比 iPSR 与 iPSR+NMC 方法生成的网格模型对尖锐特征恢复的情况。

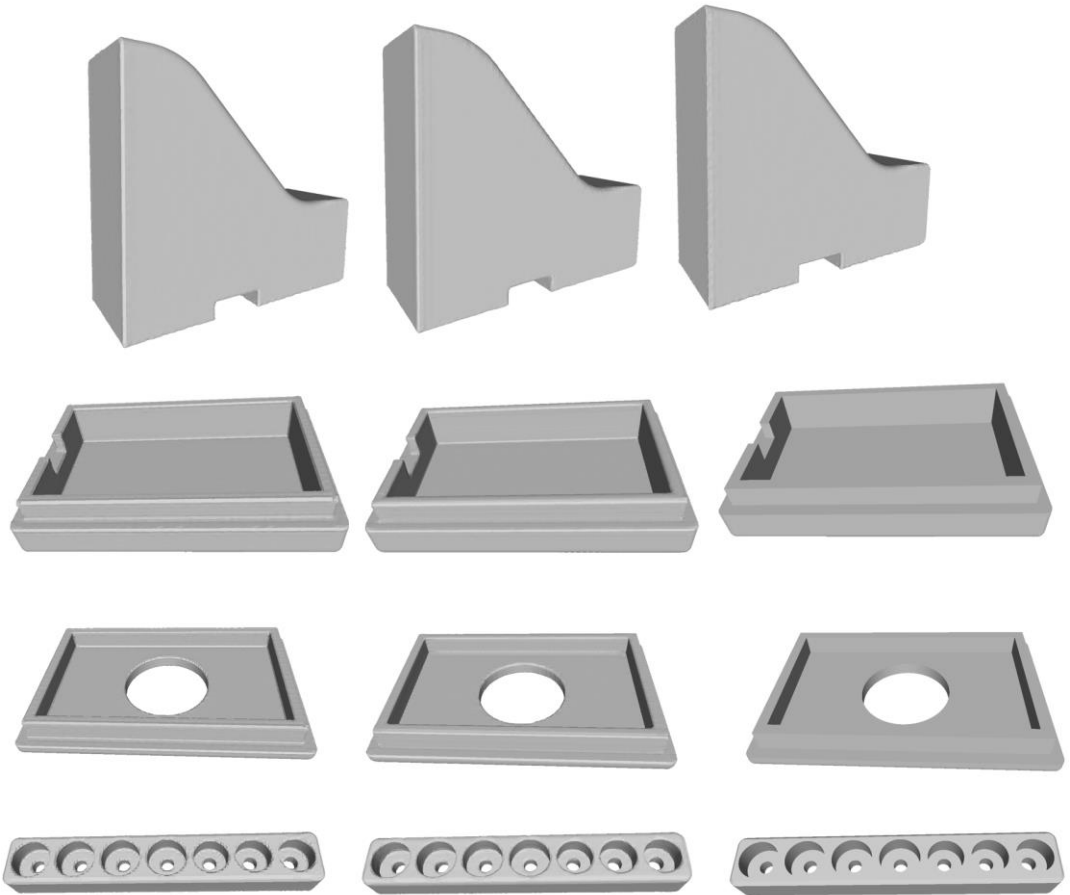
4.2 改进结果

使用改进思路之后的表面重建部分结果如图 10 所示，图 10 左列是 iPSR 的结果，中间列是 iPSR+NMC 的结果，最后 1 列是真实值网格。从结果可以定性地看出改进后的方法对边缘的恢复有一些帮助。

为了定量评估 iPSR+NMC 重建出的网格质量，选取了 CD（Chamfer Distance）、F1（F-score）作为整体网格质量的评估指标。CD 指标通过计算两个点集之间的最近距离的平均值，来衡量重建网格与真实表面之间的几何差异，CD 指标越小，说明两个点集越接近。F1 指标基于 Precision 和 Recall，来衡量生成的网格与真实值网格之间的匹配程度，F1 越高，则说明两个网格越匹配。同时使用了 BSP-Net[18]提出的 ECD（Edge Chamfer Distance）和 EF1（Edge F-Score）指标来评估重建网格尖锐边缘的质量，这两者是对传统的 CD 和 F1 的扩展，但 ECD 和 EF1 是针对边缘区域进行计算。表 1 展示了 iPSR 方法和 iPSR+NMC 方法在各指标上的对比，从结果也可以定量地得出 iPSR+NMC 可以恢复模型的尖锐特征，并且重建出较高质量的网格。

表 1 改进方法网格质量对比

	CD ($\times 10^{-5}$)↓	F1 ↑	ECD ↓	EF1 ↑
iPSR	8.7014	0.1801	0.1830	0.1076
iPSR+NMC	1.2614	0.7955	0.0285	0.2848



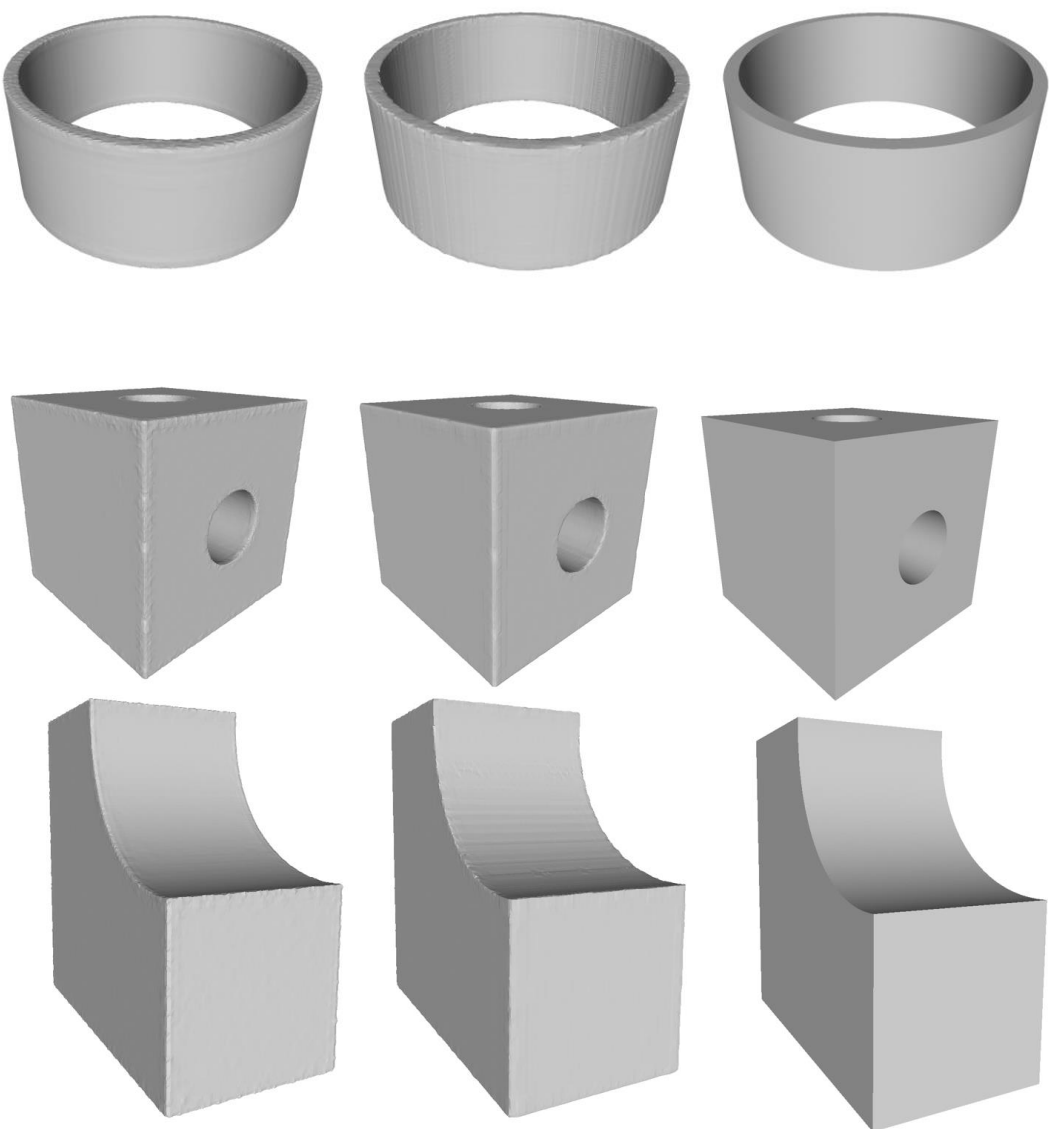


图 10 改进思路部分结果展示

参考文献:

- [1] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson Surface Reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing. 61–70
- [2] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. 1999. The Ball-Pivoting Algorithm for Surface Reconstruction. IEEE Transactions on Visualization and Computer Graphics 5, 4 (oct 1999), 349–359.
- [3] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. 2001. The Power Crust. In Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications. 249–266.
- [4] Tamal K. Dey and Samrat Goswami. 2003. Tight Cocone: A Water-Tight Surface Reconstructor. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications. 127–134

- [5] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. 1992. Surface Reconstruction from Unorganized Points. SIGGRAPH 26, 2 (jul 1992), 71–78.
- [6] Hui Xie, Jianning Wang, Jing Hua, Hong Qin, and Arie Kaufman. 2003. Piecewise C1 Continuous Surface Reconstruction of Noisy Point Clouds via Local Implicit Quadric Regression. In Proceedings of the IEEE Visualization. 91–98.
- [7] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao (Richard) Zhang. 2013. Edge-Aware Point Set Resampling. ACM Trans. Graph. 32, 1, Article 9 (feb 2013), 12 pages.
- [8] Quentin Merigot, Maks Ovsjanikov, and Leonidas J. Guibas. 2011. Voronoi-Based Curvature and Feature Estimation from Point Clouds. IEEE Transactions on Visualization and Computer Graphics 17, 6 (jun 2011), 743–756.
- [9] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. 2010. Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets. Computer Graphics Forum 29, 5 (2010), 1733–1741
- [10] Nico Schertler, Bogdan Savchynskyy, and Stefan Gumhold. 2017. Towards Globally Optimal Normal Orientations for Large Point Clouds. Comput. Graph. Forum 36, 1 (January 2017), 197–208. <https://doi.org/10.1111/cgf.12795>.
- [11] Jun Wang, Zhouwang Yang, and Falai Chen. 2012. A Variational Model for Normal Computation of Point Clouds. Vis. Comput. 28, 2 (feb 2012), 163–174.
- [12] Zhiyang Huang, Nathan Carr, and Tao Ju. 2019. Variational Implicit Point Set Surfaces. ACM Trans. Graph. 38, 4, Article 124 (July 2019), 13 page
- [13] Songyou Peng, Chiyu "Max" Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. 2021. Shape As Points: A Differentiable Poisson Solver. In NeurIPS. 13032–13044.
- [14] Rui Xu, Zhiyang Dou, Ningna Wang, Shiqing Xin, Shuangmin Chen, Mingyan Jiang, Xiaohu Guo, Wenping Wang, and Changhe Tu. 2023. Globally Consistent Normal Orientation for Point Clouds by Regularizing the Winding-Number Field. ACM Trans. Graph. 42, 4, Article 111 (August 2023), 15 pages. <https://doi.org/10.1145/3592129>
- [15] Fei Hou, Chiyu Wang, Wencheng Wang, Hong Qin, Chen Qian, and Ying He. 2022. Iterative poisson surface reconstruction (iPSR) for unoriented points. ACM Trans. Graph. 41, 4, Article 128 (July 2022), 13 pages. <https://doi.org/10.1145/3528223.3530096>
- [16] Michael Kazhdan and Hugues Hoppe. 2013. Screened Poisson Surface Reconstruction. ACM Trans. Graph. 32, 3, Article 29 (July 2013), 13 pages
- [17] Zhiqin Chen and Hao Zhang. 2021. Neural Marching Cubes. ACM Trans. Graph. 40, 6, Article 251 (dec 2021), 15 pages.
- [18] Z. Chen, A. Tagliasacchi, H. Zhang, BSP-Net: Generating Compact Meshes via Binary Space Partitioning[C]/Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2020