

关于神经三维对偶等值面提取论文的复现

马雪奇

摘要

本文复现的论文提出了一种基于数据驱动的物体表面重建方法, Neural Dual Contouring(NDC). 与传统的 Dual Contouring 方法不同的是, NDC 不需要人为设计复杂的函数来计算物体表面的梯度信息, 而是直接采用神经网络来预测栅格中顶点的位置以及棱的符号. 该方法允许多种类型的数据作为输入, 包括有符号距离场 (Signed Distance Field, SDF)、无符号距离场 (Unsigned Distance Field, UDF)、二值体素、无向点云. 另外, 该方法可以生成非水密的物体表面. 当输入数据为无向点云时, 该方法重建的面片往往无法区分内外, 本文在该方法的基础上, 利用点云的法向量信息, 对重建的面片方向进行了矫正. 从实验结果来看, 重定向后的面片方向保持了一致性.

关键词: 表面重建; 点云重建; 机器学习

1 引言

随着我国城镇化的发展, 城市规模不断地扩大, 城市化在给人们带来生活便利的同时, 也带来了交通拥堵, 事故频发等问题. 近年来提出的“智慧城市”理念为解决上述问题提供了思路. 城市的三维场景数据为智慧城市提供了完整的三维空间信息. 如何真实、完整、高精度地重建出大规模的三维城市场景成为了“智慧城市”的重要基础.

目前. 针对规模场景的三维化, 在数字测绘和计算机图形学领域, 出现了许多三维数据获取技术, 包括无人机低空航拍, 激光雷达扫描等. 其中激光雷达扫描的点云数据具有精度高、扫描速度快等特点, 但同时也存在点云分布不均匀, 部分点云残缺等问题. 本次课程的论文复现工作 Neural Dual Contouring^[1]拟通过对点云进行处理, 采用深度学习的技术, 来重建现实生活中的三维场景, 提高三维场景重建的精度和完整性, 研究真实世界低成本数字化的可行性.

与基于 Marching Cube^[2]的重建方法相比, 基于 Dual Contouring^[3]的方法能够重建更加棱角分明的网格, 能够更好地保留建筑物的特征, 适合大规模室外建筑物的重建.

2 相关工作

2.1 基于点云的网格重建

从离散的点云中重建出多边形的网格一直是计算机图形学中最经典, 也是研究最深入地问题之一. Berger 等人^[4]对以往的网格重建工作按照不同的先验进行了分类, 包括平滑度、可见性、密集采样、基元以及数据先验. 在基于数据先验的方法中, 有一些工作是从数据中学习隐含的先验知识, 包括学习整个物体的先验^[5]以及学习 patches 的先验^[6]. 本文复现的 NDC 工作属于学习 patch 先验的方法.

除此之外, 根据输入点云有无法向量, 可以将表面重建方法分为有向点云表面重建^[7]和无向点云表面重建^[8]. Peng 等人^[9]设计了一种基于卷积的 Occupancy 神经网络, 该网络基于 UNet^[10]结构, 可以学习样本数据中的全局特征, 从而可以较好地完成补全重建任务. 另外, 该方法输入的点云不需要法向量, 因此对于输入数据的适用性较高. 本文复现的 UNDC 方法同样可以从无向的输入点云中, 重建出非水

密物体的表面。并且，通过在训练中对数据进行噪声增强，使得该方法可以有效地应对点云采样分布不均匀以及噪声干扰等情况。

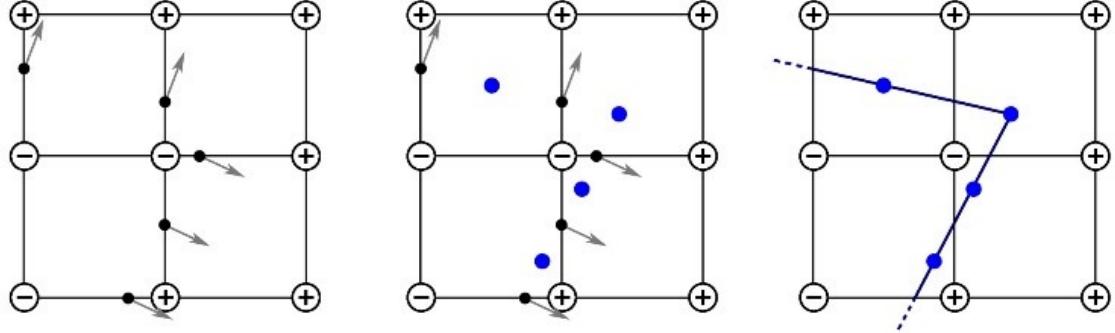


图 1: DC 输入: 角点符号、单元格边上交叉点位置以及方向。对于每一个角点符号发生变化的单元格，采用 QEM 方法在其内部计算得到一个顶点。然后，对于每一个符号发生变化的边，通过连接其周围的四个顶点，从而生成四边形面。

2.2 Dual Contouring

Ju 等人^[3]提出了 Dual Contouring 方法，该方法可以将有符号距离场 $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ 映射到多边形的网格 $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ ，如图 1 所示。该方法是通过将函数在 $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ 中离散化实现的。首先，该方法在栅格角点上对距离场 Φ 进行采样，得到每个角点的符号值；然后，对于两端符号相异的边 \mathcal{V}^3 ，找到距离场 Φ 在该边上值为 0 的位置，即交叉点；接着，通过 QEM 方法计算这些交叉点的法向；最后，根据上述信息生成四边形面。Dual Contouring 可以用以下式子来表示：

$$\mathcal{S} \in \mathbb{B}^{|\mathcal{X}|}, \mathcal{S} = f_{\mathcal{S}}(\Phi, \mathcal{G}), (\text{grid } \text{ signs}) \quad (1)$$

$$\mathcal{V}^{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}| \times 3}, \mathcal{V}^{\mathcal{E}} = f_{\mathcal{V}^{\mathcal{E}}}(\Phi, \mathcal{G}), (\text{edge } \text{ vertices}) \quad (2)$$

$$\mathcal{N}^{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}| \times 3}, \mathcal{N}^{\mathcal{E}} = f_{\mathcal{N}^{\mathcal{E}}}(\Phi, \mathcal{G}), (\text{edge } \text{ normals}) \quad (3)$$

其中， \mathcal{S} 为栅格中角点在距离场 Φ 中的符号值， $f_{\mathcal{S}} : sign(\Phi(\mathcal{X}))$ ； $f_{\mathcal{V}^{\mathcal{E}}}$ 用于计算距离场在栅格边上值为 0 的交叉点位置； $f_{\mathcal{N}^{\mathcal{E}}} : \nabla \Phi(\mathcal{V}^{\mathcal{E}})$ 为交叉点位置的梯度。而四边形面和对应的顶点可以用下式来表示：

$$\mathcal{F} \in \mathbb{B}^{|\mathcal{E}|} \quad \mathcal{F} = f_{\mathcal{F}}(\mathcal{S}) \quad (4)$$

$$\mathcal{V} \in \mathbb{R}^{|\mathcal{X}| \times 3} \quad \mathcal{V} = f_{\mathcal{V}}(\mathcal{V}^{\mathcal{E}}, \mathcal{N}^{\mathcal{E}}) \quad (5)$$

另外，只有当单元格边的符号发生变化时，才会生成四边形边：

$$f_{\mathcal{F}} : xor(\mathcal{S}_i, \mathcal{S}_j), \quad (i, j) \in \mathcal{E} \quad (6)$$

3 本文方法

3.1 Dual Contouring

本文复现的 NDC 方法, 可以通过简单的方式实现锐利的特征. 该方法基于符号预测来重建物体的面片, 可以用以下公式来表示:

$$\text{NDC}(\mathcal{I}) = \begin{cases} \mathcal{S} = f_{\mathcal{S}}(\mathcal{I}, \mathcal{G}; \theta) \\ \mathcal{V} = f_{\mathcal{V}}(\mathcal{I}, \mathcal{G}; \theta) \\ \mathcal{F} = \text{xor}(\mathcal{S}_i, \mathcal{S}_j) \end{cases} \quad (7)$$

其中用于控制是否生成面片的逻辑变量与经典的 Dual Countouring 相同. 在给定输入量 \mathcal{I} 之后, 该方法可以通过神经网络参数 θ 来预测栅格 \mathcal{G} 中顶点的位置和角点的符号. 该方法的另外一个变体为 UNDC, 其中 U 表示 unsigned(无符号). 与 NDC 不同的是, UNDC 直接预测是否生成面片, 而不用去预测角点的符号值:

$$\text{UNDC}(\mathcal{I}) = \begin{cases} \mathcal{V} = f_{\mathcal{V}}(\mathcal{I}, \mathcal{G}; \theta) \\ \mathcal{F} = f_{\mathcal{F}}(\mathcal{I}, \mathcal{G}; \theta) \end{cases} \quad (8)$$

UNDC 最大的优势在于, 该方法可以重建非水密的物体, 例如大规模的建筑场景. 同时, 该方法还可以以薄片的形式重建出网格面.

3.2 编码器

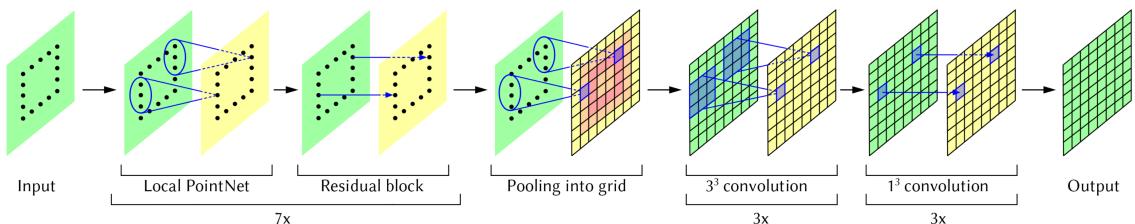


图 2: NDC 方法网络结构

以点云作为输入对象, 该网络的编码器可以分为两个部分: 点云处理模块、栅格处理模块. 前者是基于 PointNet++^[11] 实现的, 后者是通过三层 3x3x3 的卷积层以及三层 1x1x1 的卷积层实现的. 该方法的网络结构图如图 2 所示.

3.3 大场景重建方法

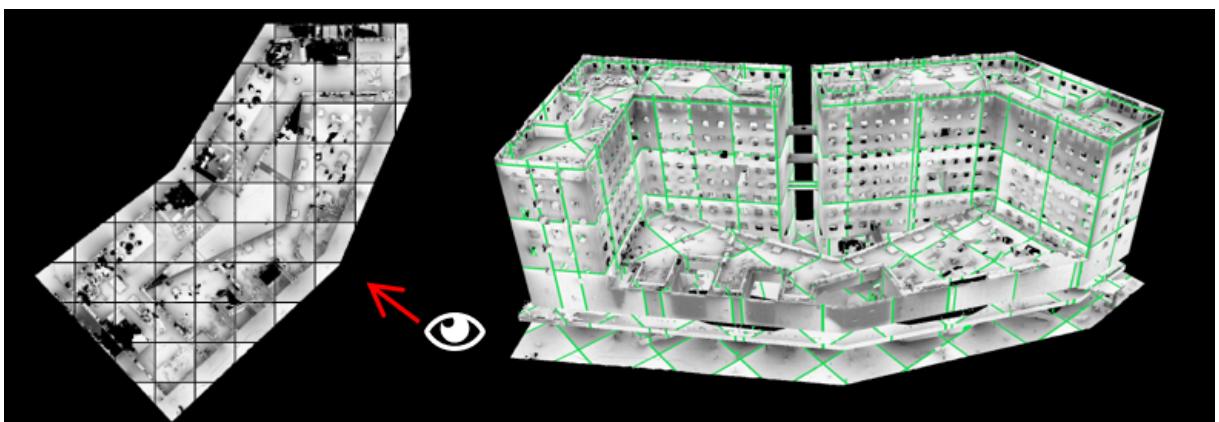


图 3: 场景空间划分示意图

对于大规模的场景,本文采用的重建方法是:首先,对大场景进行区域分割,得到若干个小区域;然后,针对每一个小区域,使用 UNDC 的方法对其进行重建.以 L7 建筑为例,其区域划分方式如图 3 所示.由于 L7 建筑相对于坐标系呈倾斜分布,因此,切分的区域往往与建筑物自身不平行.

3.4 网格法向重定向

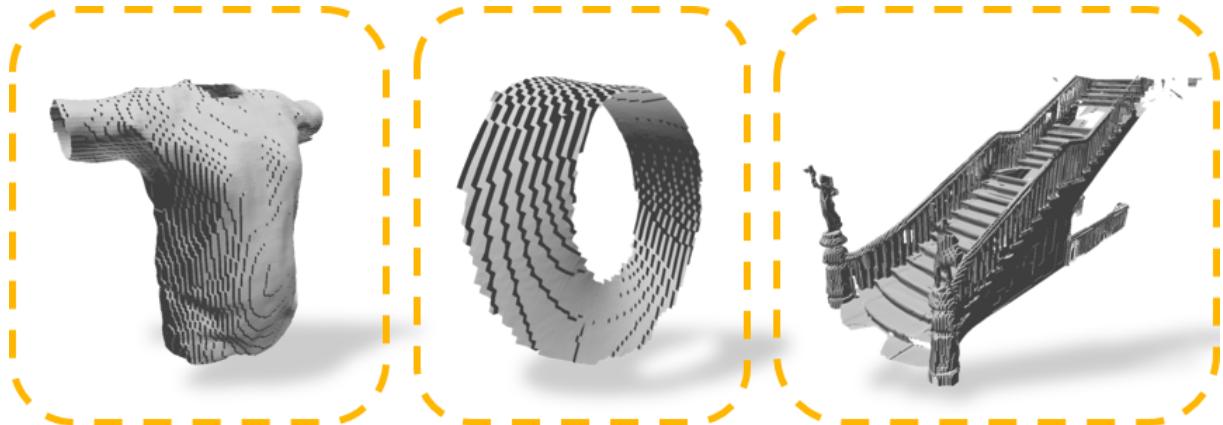


图 4: UNDC 方法重建结果

如图 4 所示,由于输入的点云为无向点云,因此 UNDC 方法重建的网格无法区分物体的内外,导致面片的法向量出现不一致的情况.

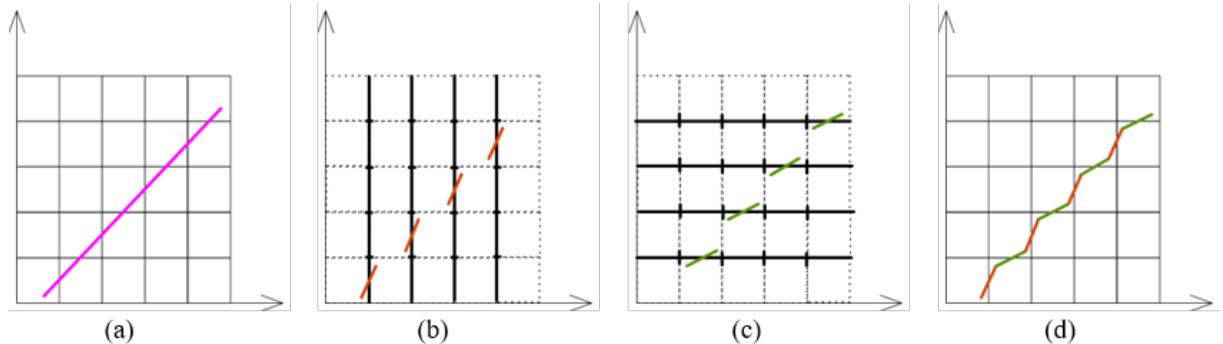


图 5: Dual Contouring 2D 示意图

由于 UNDC 无法区分场景的内外,因此该方法在生成 mesh 的时候需要提前指定面片的正方向.以图 5 为例:假设朝“右”和朝“上”为正向,为了重建出如图 5(a) 所示的紫色的真实线段,UDC 方法首先判断所有竖向的单元格边是否被切割,并使用线段将被切割的单元格边两侧的单元格连接起来,如图 5(b) 所示;由于前面假设了“右”为正向,因此,这些线段的右侧为正向;然后用同样的方式去判断所有的横线,得到的结果如图 5(c) 所示,又因为前面假设了“上”为正向,因此,这些线段的上侧为正向;最后将图 5(b) 的结果与图 5(c) 的结果拼接起来,得到最后的重建结果如图 5(d) 所示;反映到真实的三维场景中即为面片明暗相间交替出现.

为了对网格的法向量进行重新定向,我们根据激光雷达在点云采集过程中得到的法向量,利用 k-d tree 方法对所有面片的法向量进行了矫正,重定向结果见 5.3 节.

3.5 后处理

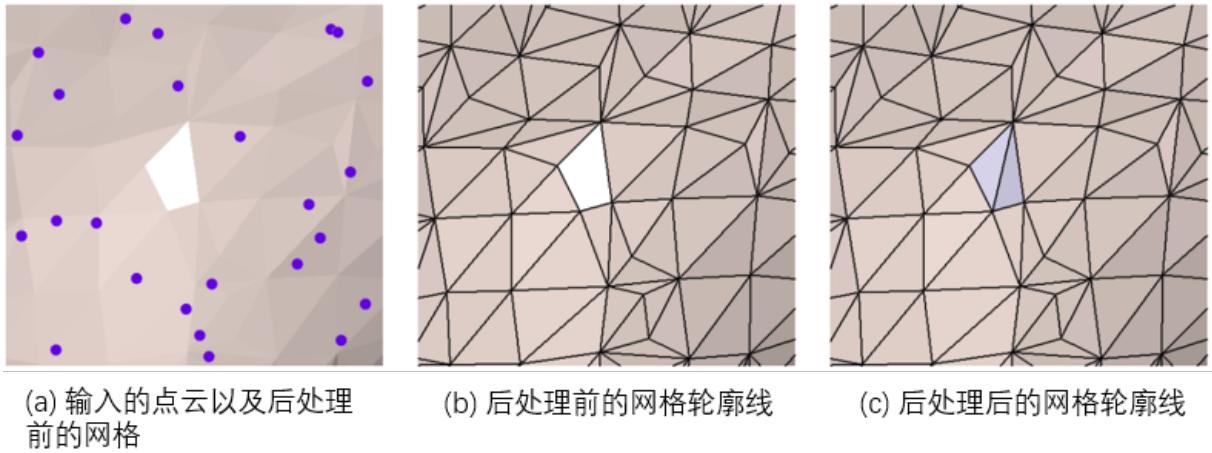


图 6: UNDC 方法后处理输出

如图 6 所示, 当 UNDC 在处理稀疏或者带有噪声的点云时, 用于预测单元格边是否被交叉的函数可能会出现错误, 导致在输出网格的时候产生一些小空洞, 因此需要采取一些后处理方法来进行弥补. 本文首先检测出周围存在四条边同时自身处于未被交叉状态的单元格边, 接着将其交叉状态修改为正, 然后对该区域进行重新网格化.

4 复现细节

4.1 与已有开源代码对比

本文复现了 NDC 中基于点云的网格重建方法, 其中, 神经网络以及 Dual Contouring 方法的实现方式与 UNDC 相似. 另外, 本文在此基础上, 对 UNDC 方法遇到的重建出的网格法向不一致的问题进行了修复.

4.2 训练数据准备

本文以 ABC 数据集为训练数据, 对 UNDC 网络进行了训练. 为了获取训练数据的真值, 本文将随机得到的三维网格样本放置在栅格中, 然后利用经典的 DC 方法来计算栅格角点的符号, 交叉点的位置和梯度, 并以此作为真值, 来对网络进行训练.

4.3 训练损失函数

在获取到真值数据之后, UNDC 方法的所有子网络均可以进行单独训练. 下式为顶点计算的损失函数:

$$\mathcal{L}_V(\theta) = \mathbb{E}_{(\mathcal{I}, \mathcal{M}_V, V_{gt}) \sim \mathcal{D}} \sum_{m,n,k} [\|\mathcal{M}_V \odot (f_V(\mathcal{I}, \mathcal{G}; \theta))\|_2^2] \quad (9)$$

这里, \odot 为 Hadamard 乘积. 对于 UNDC, 我们通过二值交叉熵 (Binary Cross Entropy) 来监督单元格边的符号, 下式为面预测的损失函数:

$$\mathcal{L}_S(\theta) = \mathbb{E}_{(\mathcal{I}, \mathcal{M}_S, S_{gt}) \sim \mathcal{D}} \sum_{m,n,k} [S_V \odot \text{BCE}(f_S(\mathcal{I}, \mathcal{G}; \theta), S_{gt})] \quad (10)$$

4.4 实验环境搭建

4.4.1 环境配置

环境要求

操作系统	Ubuntu 20.04	
开发平台	Docker	
CPU	Intel(R)Xeon(R) CPU E5-2620 v4 @ 2.10GHz 32GB	用于数据预处理
GPU	RTX 3090 24GB	用于模型训练以及网格生成
编译语言	Python, C++	Python 进行模型的训练和测试, C++ 用于加速网格提取;
DL 开发平台	Pytorch	

配置环境

- 首先在 Ubuntu 系统上安装 docker;
- 在容器中安装 Anaconda, 并创建虚拟环境;
- 安装 Pytorch 以及必要的 Python 包;
- Build Cython 文件;

4.5 界面分析与使用说明

模型训练

训练单元格棱符号预测模型

```
1 python main.py --train_bool --input_type pointcloud --method undc --
    epoch 250 --lr_half_life 100 --data_dir ./groundtruth/gt_UNDC --
    checkpoint_save_frequency 10 --point_num 4096 --grid_size 64
```

训练单元格内顶点位置预测模型

```
1 python main.py --train_float --input_type pointcloud --method undc --
    epoch 250 --lr_half_life 100 --data_dir ./groundtruth/gt_UNDC --
    checkpoint_save_frequency 10 --point_num 4096 --grid_size 64
```

模型测试

```
1 python main.py --test_bool_float --input_type udf --method undc --
    data_dir ./groundtruth/gt_UNDC
```

4.6 创新点

虽然 UNDC 方法可以基于离散点云重建出非水密的面片, 但是由于输入的点云本身无法向信息, 因此重建的面片的方法往往存在异常. 为了解决这个问题, 本文在 UNDC 的基础上, 对原方法进行了完善. 本文基于点云的法向量和 kd-tree, 对所有面片的法向进行了纠正, 从而保证重建的物体表面的法向保持一致性.

5 实验结果分析

5.1 数据集

本文在 ABC 数据集上对 UNDC 网络进行了训练。ABC 数据集中包含了丰富的 CAD 模型，并且这些模型具有锐利的边缘特征以及光滑的曲面特征。本文将数据集中分为训练集 (80%, 4280 个样本) 和测试集 (20%, 1071 个样本)，并以 L7 楼点云数据为样本进行了测试。

5.2 基于点云的表面重建

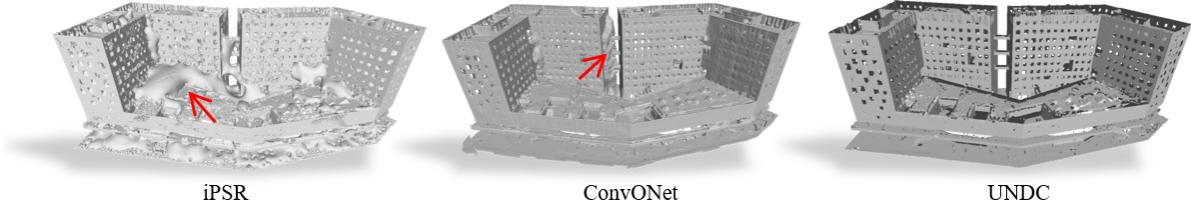


图 7: 不同方法重建结果对比

本文与另外两种大场景点云重建方法进行了对比，包括迭代泊松重建^[12]、ConvONet^[9]。本文以 L7 楼点云作为输入数据，对比结果如图 7 所示。

iPSR 虽然可以用于处理无向点云的重建，但是该方法重建的结果会出现许多的凸包；ConvONet 重建的结果同样出现了许多不必要的凸包；而 UNDC 重建的结果结构清晰，无冗余的凸包情况出现。

5.3 网格法向重定向

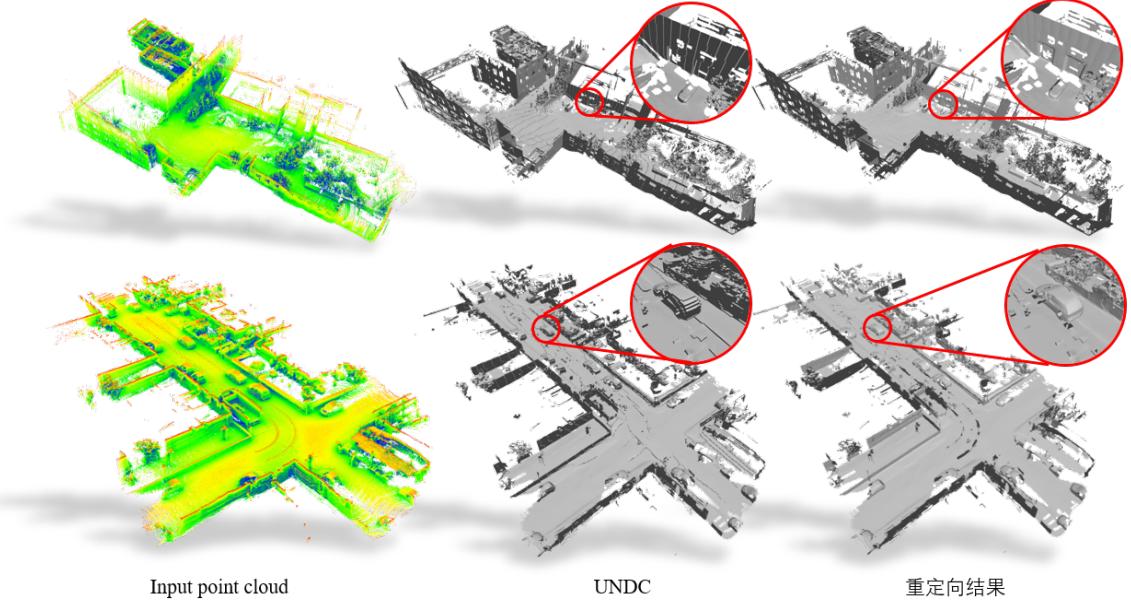


图 8: 网格重定向结果

由于 UNDC 重建的网格无法区分内外，因此本文利用激光雷达在扫描过程中获得的点云法向量对 UNDC 重建的结果进行了重定向，结果如图所示。从图 8 中可以看出，重定向之后的网格法向保持了连续性。

5.4 其他场景重建结果

本文对其他场景同样进行了实验，实验结果如图 9, 10 所示

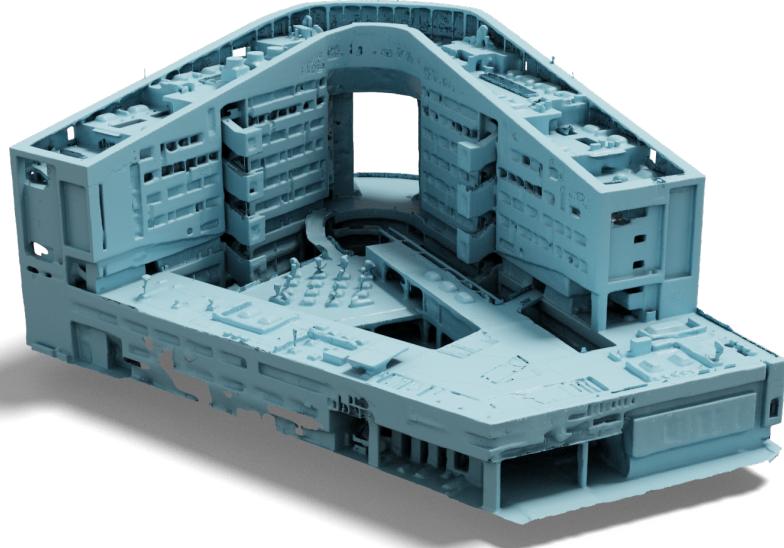


图 9: 致腾楼重建结果

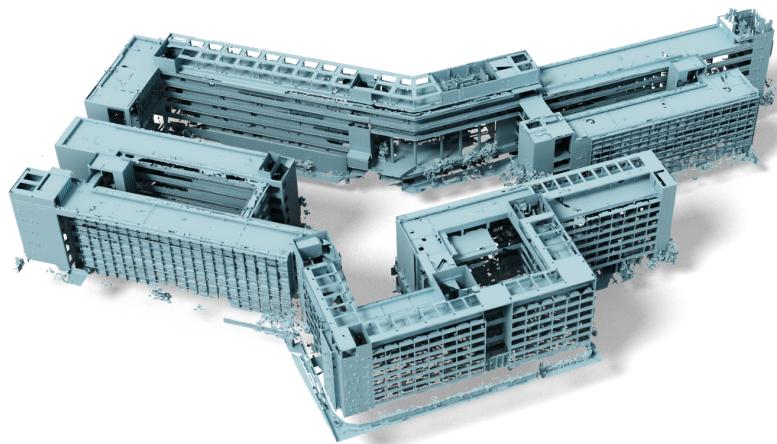


图 10: 汇文楼重建结果

6 总结与展望

UNDC 方法是一种基于数据驱动的三维等值面提取算法, 它弥补了原始 Dual Contouring 算法对输入要求较为严格 (提供梯度方向) 的不足, 并能从无符号距离场中重建出三维网格. 但是该方法因为无法区分场景的内外信息, 因此重建的网格面片法向往往会出现不一致的情况. 本文针对该问题, 利用输入点云的法向信息, 对重建后的网格法向进行了重新定向, 从而保证了网格法向的一致性. 不过, 由于输入的点云存在残缺性, 导致重建的结果在部分局部位置仍存在一些空洞. 因此, 残缺区域的补全重建问题, 将是未来的探索和研究方向.

参考文献

- [1] CHEN Z, TAGLIASACCHI A, FUNKHOUSER T, et al. Neural dual contouring[J]. ACM Trans. on Graphics (Proc. SIGGRAPH), 2022, 41(4): 1-13.

- [2] LORENSEN W E, CLINE H E. Marching cubes: A high resolution 3D surface construction algorithm [J]. ACM Trans. on Graphics (Proc. SIGGRAPH), 1987, 21(4): 163-169.
- [3] JU T, LOSASSO F, SCHAEFER S, et al. Dual contouring of hermite data[J]. ACM Trans. on Graphics, 2002, 32(3): 339-346.
- [4] BERGER M, TAGLIASACCHI A, SEVERS KY L M, et al. A survey of surface reconstruction from point clouds[J]. Computer Graphics Forum, 2017, 36(1): 301-329.
- [5] PENG S, JIANG C, LIAO Y, et al. Shape As Points: A Differentiable Poisson Solver[C]// Proc. Conf. on Neural Information Processing Systems. 2021: 13032-13044.
- [6] SHEN C H, FU H, CHEN K, et al. Structure Recovery by Part Assembly[J]. ACM Trans. on Graphics, 2012, 31(180): 1-11.
- [7] KAZHDAN M, HOPPE H. Screened poisson surface reconstruction[J]. ACM Trans. on Graphics (Proc. SIGGRAPH), 2013, 32(3): 1-13.
- [8] ATZMON M, LIPMAN Y. Sal: Sign agnostic learning of shapes from raw data[C]// Proc. IEEE/CVF Conf. on Computer Vision & Pattern Recognition. 2020: 2565-2574.
- [9] TANG J, LEI J, XU D, et al. Sa-convonet: Sign-agnostic optimization of convolutional occupancy networks[C]// Proc. Int. Conf. on Computer Vision. 2021: 6504-6513.
- [10] RONNEBERGER O, FISCHER P, BROX T. U-Net: Convolutional networks for biomedical image segmentation[C]// Proc. Int. Conf. on Medical Image Computing and Computer Assisted Intervention. 2015: 234-241.
- [11] QI C R, YI L, SU H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space [C]// Proc. Conf. on Neural Information Processing Systems. 2017: 5105-5114.
- [12] HOU F, WANG C, WANG W, et al. Iterative poisson surface reconstruction (iPSR) for unoriented points [J]. ACM Trans. on Graphics (Proc. SIGGRAPH), 2022, 41(128): 1-13.