

基于快速点直方图和深度学习的断裂面分割算法

郭子旭¹ 谢晓尧¹ 刘 嵩¹ 刘旭斌¹
GUO Zixu XIE Xiaoyao LIU Song LIU Xubin

摘 要

文物修复中断裂面分割至关重要，它是后续精准拼接、还原文物形态和结构的基础步骤，能够为文物修复工作提供精确的修复边界和匹配依据，有助于最大程度地恢复文物的完整性和原有风貌，同时也有利于对文物内部结构和成分进行科学研究。针对于文物修复中断裂面分割问题，文章提出了一种基于 PointNet++ 和点快速特征直方图（fast point feature histograms, FPFH）的断裂面分割算法 FPFH-PointNet++。在一定程度上解决了大部分传统算法在面对物体表面比较复杂的情况时，断裂面也会被错误分割的问题。通过 FPFH-PointNet++ 对碎片的断裂面进行提取，并在 FragTag3D 数据集与 Presious 数据集上进行了测试。结果表明，这种方法能够提取这些数据集中大部分碎块的断裂面，在复杂表面的碎块上提取断裂面效果远超传统算法。

关键词

FPFH；深度学习；分割断裂面；PointNet++；断裂面拼接

doi: 10.3969/j.issn.1672-9528.2024.12.031

0 引言

在文物修复过程中，早期文物专家和研究员，根据史料和经验，通过人工方式对文物进行手工复原。该方法不仅费力耗时，在拼接过程中可能对文物产生二次磨损。随着计算机技术的发展，文物可以用激光扫描等方式，以数字信息为载体输入到特定的软件中进行模拟拼接，而拼接的首要步骤就是断裂面的识别和提取。断裂面的提取目前还是依赖于人工肉眼识别或一些传统算法，比如区域增长、基于特定特征（如曲率、法向量）的聚类算法。人工识别准确率高但是效率低，传统算法对复杂轮廓的碎片进行分割时准确率低。因此，本文旨在探寻一种基于深度学习方面，面对碎片轮廓比较复杂时也可以分割出完整断裂面的算法，进而能够提高整个拼接流程的效率。

1 相关工作

1.1 基于传统算法的碎片拼接方法

在传统的碎片拼接方法中，通常通过提取碎片的断裂面信息进行拼接，这是一种常见的工作流程。Huang 等人^[1]通过计算积分不变量以判断顶点的凹凸程度，从而提取碎片边缘，接着应用图割算法获取碎片面，再使用区域生长算法对断裂面顶点进行聚类，形成环形特征区域，并利用这些区域进行拼接；Mavridis 等人^[2]采用高效的数据结构和三层由粗到细的搜索策略，提升了几何匹配的效率，实现了在不依赖

特征计算的情况下，将两个碎片进行预期对齐。通过区域生长过程提取可能的断裂面，并利用平均法线偏差作为终止条件来控制生长过程；Du 等人^[3]采用了一种基于区域生长算法的交互式方法，手动提取属于完整表面的碎片，并根据其与完整模型的最佳匹配位置进行组合；Jia 等人^[4]通过点云的象征性投影距离提取关键点，并利用关键点的局部邻域信息构建多尺度协方差矩阵描述符，以描述关键点的局部表面特征；Li 等人^[5]通过多尺度曲率区分边缘点和非边缘点，基于最大和最小主曲率计算曲率，并通过拟合邻域点的二次曲面进行鲁棒估计，随后用区域生长算法将碎片分割成由尖锐边缘界定的多个表面；Savelonas 等人^[6]采用区域生长方法对输入碎片进行分割，通过计算相邻元素的二面角得分来完成分割，并在后处理阶段根据平均法线和法线方差对小段进行分割与合并；Georgios 等人^[7]利用 z-buffer 技术提取断裂面的边界线，通过将表面法线与 z 轴对齐，接着使用正交投影将表面投影到 z-buffer 中，进而提取出外边界；袁洁等人^[8]基于 Morse-Smale 复形的四边形性质，通过结合断裂面上凹凸信息的四边形曲面，更好地反映空间曲面特征，利用凹凸互补性筛选出最优匹配集，提高了拼接的准确性；何璐阳等人^[9]利用区域生长算法和改进的曲面融合算法分割破碎物体的碎片断裂面，随后通过边界提取算法获得断裂面的边缘轮廓曲线，并以此进行拼接匹配。

1.2 基于机器学习的碎片拼接方法

随着深度学习技术的发展，越来越多的相关算法被引入

1. 贵州师范大学 贵州贵阳 550001

到碎片拼接领域，以辅助这一过程。Charles 等人^[10]提出了 PointNet 结构，这是将原始点云作为深度学习输入的创新性工作，广泛应用于 3D 对象检测和语义分割。尽管 PointNet 聚合了每个点的高维局部特征，但由于多层感知机仅学习每个点的局部特征，忽略了点之间的联系，从而限制了其在复杂场景中的表现。为了解决这一问题，PointNet++^[11]引入了类金字塔特征聚合方案，尽管如此，仍未能充分利用点云输入的空间分布信息；Yao 等人^[12]借鉴 Li 的研究，先提取碎片模型的断裂面点云，然后利用孪生 PointNet 比较两个碎片断裂面之间的相似度，最终通过主成分分析（PCA）和深度最近点（DCP）拼接两碎片，实现精准匹配；Roberto 等人^[13]结合机器学习，采用 One-Class SVM 模型来提取带有油漆装饰表面的碎片，该模型用于从碎片的非装饰表面颜色中训练，通过用户评估观察值是否为异常值的方式来划分断裂面。

2 方法

碎片的拼接流程如图 1 所示，首先进行数据的预处理部分，进行快速点特征直方图(FPFH)的计算和输入格式的处理，再把规格化后的数据输入到训练好的 FPFH-PointNet++ 模型中，进行断裂面的分割提取。最后把提取出的断裂面及其点的特征 FPFH，应用到碎片断裂面匹配算法中。由于把断裂面根据特征进行匹配的算法比较常见，本文不再赘述。

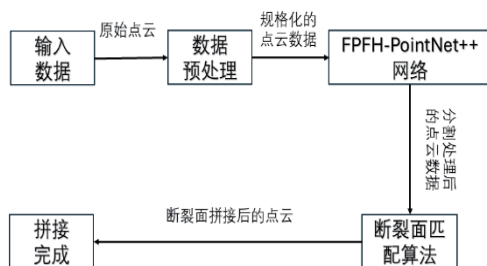


图 1 碎片拼接流程

2.1 数据预处理

2.1.1 数据归一化处理

$$PC_{\text{centroid}} = \frac{1}{N} \sum_{i=1}^N PC_i \quad (1)$$

$$PC_{\text{normalized}} = PC - PC_{\text{centroid}} \quad (2)$$

$$\text{Norm}_{\text{max}} = \text{Max} \left(\sqrt{x_i^2 + y_i^2 + z_i^2} \right) \quad (3)$$

$$PC_{\text{normalized}} = \frac{PC_{\text{normalized}}}{\text{Norm}_{\text{max}}} \quad (4)$$

式中： PC_i 为点云信息 PC 中的点； x_i 、 y_i 、 z_i 分别为 PC_i 对应的点坐标； PC_{centroid} 为点云 pc 的质心； $PC_{\text{normalized}}$ 为归一化后的点云信息。

2.1.2 FPFH 计算

点特征直方图（point feature histograms, PFH）是一种描述三维点云中点的局部几何特征的方法，也是计算快速点特征直方图（fast point feature histograms, FPFH）的前置算子。它通过计算查询点与其欧式局部邻域点之间的空间关系，绘制多维直方图，得到特征分布。FPFH 是基于 PFH 改进的算子，如图 2 所示。

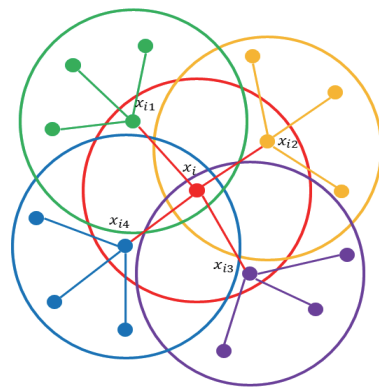


图 2 FPFH 的计算范围

对于给定包含 n 个点的点云， x_i 初始特征为欧式三维坐标，通常表示成 $x_i \in \mathbf{R}^{3 \times n}$ ， $N(x_i)$ 表示 x_i 的 k 个邻居点，对于查询点 x_i 的任意一个邻居点 x_{ij} ， $x_{ij} \in N(x_i) = \{x_{i1}, x_{i2}, \dots, x_{ik}\}$ ，如图 3 所示。

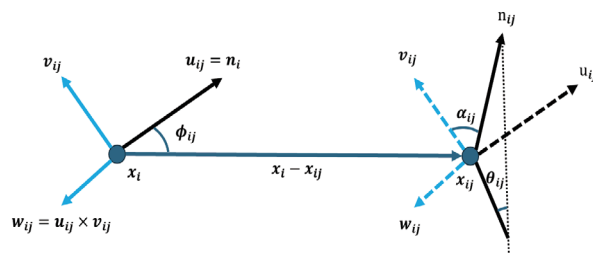


图 3 FPFH 特征描述子的计算

FPFH 算子构建如下。

（1）点对局部坐标系：在三维空间中建立点对局部坐标系，以此描述点云几何特征。

$$\begin{cases} u_{ij} = n_i \\ v_{ij} = u_{ij} \times \frac{(x_i - x_{ij})}{\|x_i - x_{ij}\|_2} \\ w_{ij} = u_{ij} \times v_{ij} \end{cases} \quad (5)$$

式中： $\times$ 表示外积； n_i 表示查询点 x_i 的法向量。在建立的 UVW 坐标系中， u_{ij} 是 x_i 点的法向量； v_{ij} 垂直于向量 u_{ij} 和 $x_i - x_{ij}$ 所在的平面； w_{ij} 同时垂直于向量 u_{ij} 和 v_{ij} 。局部坐标系的建立是为了计算后面的角度特征。

（2）简化点特征直方图（SPFH）计算：SPFH 值是 FPFH 值的组成部分，用于计算查询点 x_i 和近邻点 x_{ij} 之间的特征差异。

$$\begin{cases} \alpha_{ij} = \mathbf{n}_{ij} \cdot \mathbf{v}_{ij} \\ \phi_{ij} = \mathbf{u}_{ij} \cdot \frac{(\mathbf{x}_i - \mathbf{x}_{ij})}{d_{ij}} \\ \theta_{ij} = \arctan(\mathbf{w}_{ij} \cdot \mathbf{n}_{ij}, \mathbf{u}_{ij} \cdot \mathbf{n}_{ij}) \end{cases} \quad (6)$$

式中: $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_{ij}\|_2$ 表示点 $\mathbf{x}_i$ 和邻居点 $\mathbf{x}_{ij}$ 之间的欧式距离, $\cdot$ 表示内积。 α 、 Φ 都是归一化向量之间的点积, 它们实际物理含义是 3D 向量之间夹角的余弦值, 因此取值范围在 ± 1 。 θ_{ij} 是 $\mathbf{n}_i$ 和 $\mathbf{n}_{ij}$ 在由 UW 定义的平面上投影后形成的夹角, 其取值范围在 $\pm \pi$ 。式 (6) 求解的 α 、 Φ 、 θ 参数, 就构成简单点特征直方图 (SPFH) 描述子。SPFH 只计算查询点 $\mathbf{x}_i$ 和邻居点 $\mathbf{x}_{ij}$ 之间的特征元素。

(3) FPFH 构建: 通过将一个点的 SPFH 与其邻域内点的 SPFH 组合, 构建 FPFH。这一步通过加权邻域内点的 SPFH 来完成, 从而获得更丰富的特征描述。

$$\text{FPFH}(\mathbf{x}_i) = \text{SPFH}(\mathbf{x}_i) + \frac{1}{k} \sum_{j=1}^k \frac{1}{w_{ij}} \cdot \text{SPFH}(\mathbf{x}_{ij}) \quad (7)$$

$$w_{ij} = \frac{d_{ij}}{\sum_{j=1}^k \frac{1}{d_{ij}}} \quad (8)$$

权重 w_{ij} 的设置一般与查询点 $\mathbf{x}_i$ 和邻居点 $\mathbf{x}_{ij}$ 之间的距离有关, 距离越近, 权重越大。

2.2 FPFH-PointNet++ 网络

2.2.1 模型结构

尽管 PointNet++ 通过多尺度和多分辨率的方式提取特征, 但对于非常复杂或细小的局部结构, 其捕捉能力可能仍然不足, 这种方法可能无法很好地适应点云中的不规则分布和密度变化。FPFH 作为一种强大的局部描述符, 不仅可以用于改进点云的配准过程, 也可以用来进一步补充 PointNet++ 在特征提取中可能忽略的局部细节。而且 FPFH 可以与 PointNet++ 中的多尺度特征提取相结合, 能够提供更丰富的局部和全局特征表示。

该层次化结构由 4 个主要模块构成, 分别是 FPFH 特征提取层、多尺度分组集合抽象层、特征传播层和 Softmax 模块。这 4 个模块共

同在提取点云中的特征上起作用, 具体的结构如图 4 所示, $n \times 3$ 代表输入点的坐标信息。

FPFH 特征提取层对应图中 FPFH 模块, 对未进行下采样的点云先进行归一化, 使得所有输入到网络的点云统一尺寸, 再对点云进行局部特征 FPFH 的计算, 再把含有局部细节特征的点输入到下一层。

多尺度分组集合抽象层对应图 4 中的 SA 模块, 该模块结构能够对包含 FPFH 特征的点云进行分层抽样和特征聚合。在操作过程中, 多尺度分组集合抽象层首先通过最远点采样方法选择远距离的点, 随后在这些采样点的球形领域内选取邻域点。然后, 不同半径球形领域内的点会被分成不同的组, 并分别输入深度卷积神经网络以进一步提取点的特征。最后, 这些不同半径组的特征会被拼接起来, 并返回最远点采样的点及其对应的特征。在具体实现时, 集合抽象层首先将带有 FPFH 特征的输入点看作一个整体组, 并将其输入到深度卷积神经网络中, 以进一步提取特征。最终返回分组后的点及其提取到的特征。

特征传播层对应图 4 中的 FP 模块, 该模块负责将从采样点获取的特征传播回原始点, 以便在网络中实现上采样。在实际操作中, 特征传播层首先将未采样的点、采样点以及这些点的特征输入模块, 计算输入点与采样点之间的距离, 并基于距离对采样点的特征进行插值, 从而获取与输入点相对应的特征。接着, 将这些特征输入深度卷积神经网络进行上采样, 最终返回上采样后的点的特征数据。

Softmax 模块包含两部分, Conv 层和 Log_softmax 层, Conv 层用于对点特征进行逐点卷积操作。它对每个点独立地应用多个全连接层 (类似于 1×1 卷积) 来提取特征。Log_softmax 层将网络的输出转换为概率分布。它可以对点的特征进行归一化处理, 使输出的值则表示每个点是否划分为断裂面上的概率。

2.2.2 损失函数

在训练模型过程中, 需要使用损失函数度量模型预测结果与真实值之间的差异。本文使用的损失函数为:

$$p_i = \frac{e^{x_{i,y}}}{\sum_{j=1}^N e^{x_j}} \quad (9)$$

$$\text{Loss}_i = -\log p_i \quad (10)$$

$$\text{Loss} = \frac{\sum_{i=1}^B \text{Loss}_i}{B} \quad (11)$$

式中: $x_{i,y}$ 为第 i 个数据通过模型后输出的对应真实值的值; $\sum_{j=1}^N e^{x_j}$ 为通过模型后输出的对应所有类别的值; p_i 为第 i 个数据预测的类别概率; Loss_i 为使用第 i 个数据的损失函数值。对所有样本求平均值后, 可以得到在该 Epoch 下的损失函数值。在测试阶段, 通过训练好的模型可以预测采样后碎片点云每个点属于某个类别的概率值。因此, 对碎片点云预测次

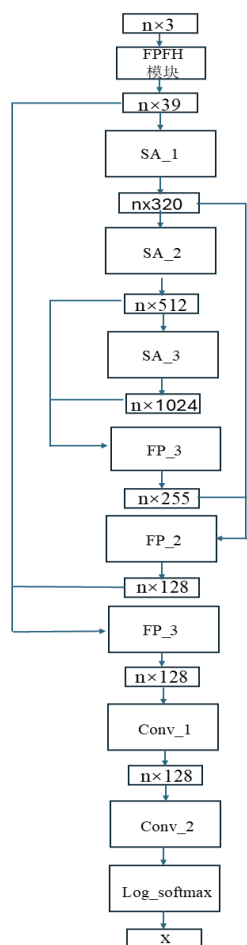


图 4 FPFH-PointNet++ 结构图

后, 取输出结果的平均值可以得到最终点云中每个点所对应类别的概率。通过这些概率值, 可以确定每个点最终属于的类别, 具体公式为:

$$\text{Pred} = \frac{\sum_{i=1}^n \text{Pred}_i}{n} \quad (12)$$

式中: Pred_i 为第 i 次预测的点云中每个点分别属于断裂面和非断裂面的概率值; Pred 为最后输出的概率值。在本实验中, 将 n 设置为 3。

3 实验

3.1 数据集

本文主要使用 FragTag3D 数据集^[14]训练与测试模型, 该数据集是模拟破碎物体的数据集, 专门用于文物拼接、点云配准领域。它通过软件 Blender 中的碎片修改插件, 基于 Thingy10k 数据集^[15]与 Precious 3D 数据集^[16]。碎片与碎片之间的重叠区域信息需使用对应颜色进行标注。目前, FragTag3D 总共包含 294 个碎片模型, 其中, 292 个来自 Thingy10k 的模型, 2 个来自于 3DPuzzle^[1]的模型。

3.2 实验设备及参数

本文方法主要是在搭载 Intel i5-12600KF 处理器、32 GB 内存、12 GB NVIDIA GTX 4070 显卡的电脑平台上进行实验。在软件环境方面, 则使用了 Ubuntu 20.04、Python 3.8.13、PyTorch 2.1、CUDA12.1。在实验中, 将 FragTag3D 数据集的数据按照的比例分成训练集、验证集、测试集。在模型的训练阶段, 首先, 需要输入的碎片模型文件, 为防止局部信息丢失, 先进行归一化和 FPFH 的计算。输入到 SA 模块时, 为了满足运行时显存空间的要求, 对点云进行随机采样, 保留 4096 个点用于训练; 然后, 需要使用批量大小为 20 的值进行训练, 采用 Adam 优化器优化网络的 200 个 Epoch (其初始学习率为 0.001, 在每 20 个 Epoch 之后缩小 1 倍学习率); 最后, 当学习率缩小到小于 0.000 01 时, 保持学习率为 0.000 01。而在模型的测试阶段, 使用的是批量大小为 1 的值进行测试, 在测试数据上使用的与训练时的数据相近。为此, 在将这些数据输入训练完成的模型后, 需要输出模型预测的颜色标签, 再结合输入的数据生成提取断裂面后的点云文件。

3.3 实验结果

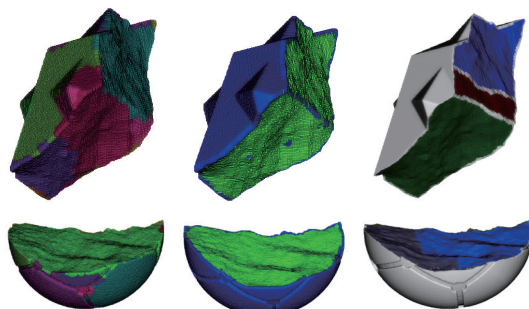
3.3.1 模型效果展示

mIoU, 即均交并比, 是一种在语义分割领域中衡量模型预测与实际标注之间的相似性的指标, 其是通过测量模型预测的分割区域与真实标注的交集与并集之间的比例进行衡量, 计算公式为:

$$\text{mIoU} = \frac{\text{TP}}{\text{FP}} \quad (13)$$

式中: FP 为人工标记断裂面后, 断裂面上点的数量; TP 为模型正确预测出的位于断裂面上点的数量; 将测试数据输入模型后, 计算 mIoU 得到的结果为 0.85。

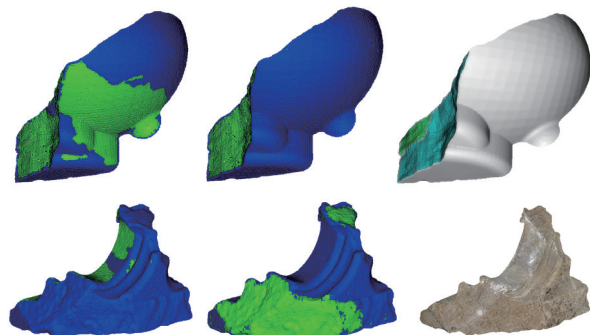
在图 5 中, 分别展示了 FragTag3D 数据集其中两块碎片分别利用传统算法和深度学习网络进行分割后的效果。左侧是传统算法分割后的点云, 中间是深度学习网络分割后的点云, 右侧是原点云。从图 5 中可知, 在分割断裂面时, 传统分割算法在面对表面形状较为复杂的碎块时, 会出现过度分割甚至时错误分割的情况, 分割效果不理想, 而深度学习网络分割效果会更好一些。



(a) 传统算法 (b) 深度学习网络 (c) 初始点云

图 5 传统算法和深度学习网络的分割效果

在图 6 中, 分别显示了 FragTag3D (上) 和 Precious (下) 中的碎块应用两种深度学习网络后的分割效果。左侧是 PointNet++ 分割后的点云, 中间是改进后网络 FPFH-PointNet++ 分割后的点云, 右侧是原点云。从图 6 可知, 在分割断裂面时, 尽管 PointNet++ 网络应用了多尺度的手段, 利用点的位置信息, 对点提取高维特征, 但这种高维特征在局部几何特征的表达上仍然有所不足, 特别在面对断裂面分割这种需求时, 这种不足更加明显。而 FPFH-PointNet++ 补充了 PointNet++ 在特征提取中可能忽略的局部细节, 在一定程度上弥补了在断裂面分割方面存在的缺陷。



(a) PointNet++ (b) FPFH-PointNet++ (c) 初始点云

图 6 PointNet++ 和 FPFH-PointNet++ 网络的分割效果

3.3.2 泛化能力展示

在图 7~10 中, 上左、上右、下左分别是真实文物数据集 Precious 中的碎块通过传统算法, PointNet++ 网络和 FPFH-PointNet++ 网络分割后的效果, 下右是文物碎块初始点云。在图 7~8 中, 传统算法在面对碎块表面比较简单的时候, 其分割效果还算理想。但是对于图 9~10 中的碎块时, 传统算法会对物体表面会发生过度分割甚至错误分割。在图 7~10 中, PointNet++ 由于提取局部几何信息的不足, 无法从碎块中正确的分割出断裂面, 而 FPFH-PointNet++ 弥补了 PointNet++ 不能充分提取局部几何信息的不足, 能够正确的分割出断裂面。虽然 FPFH-PointNet++ 的分割效果仍然不及人工分割的效果, 但是在面对大批量碎块需要进行断裂面分割和匹配时, FPFH-PointNet++ 网络的效率远高于人工。

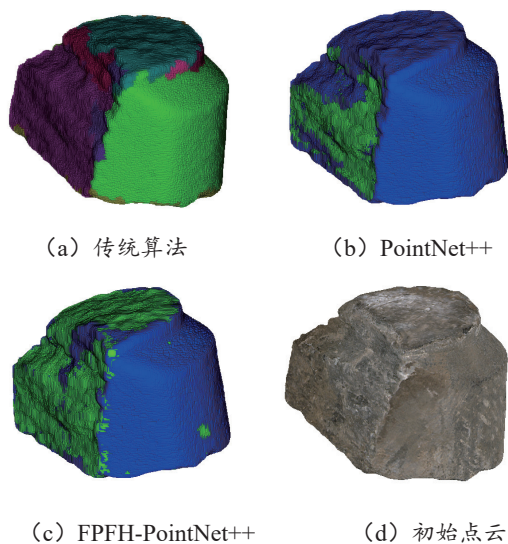


图 7 Precious 碎块 DoraColumnBase6 分割效果

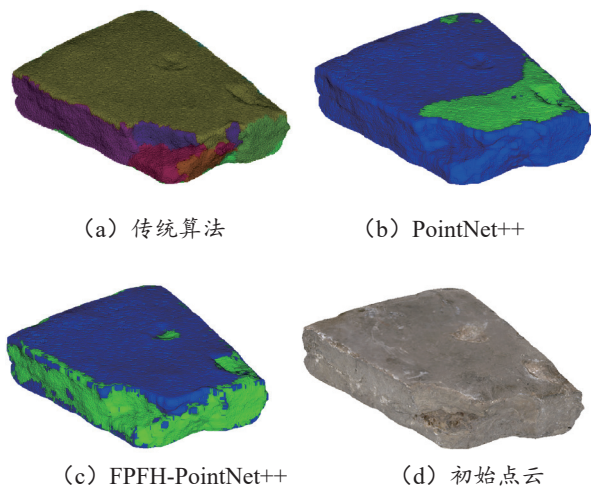


图 8 Precious 碎块 Tombstone1 分割效果

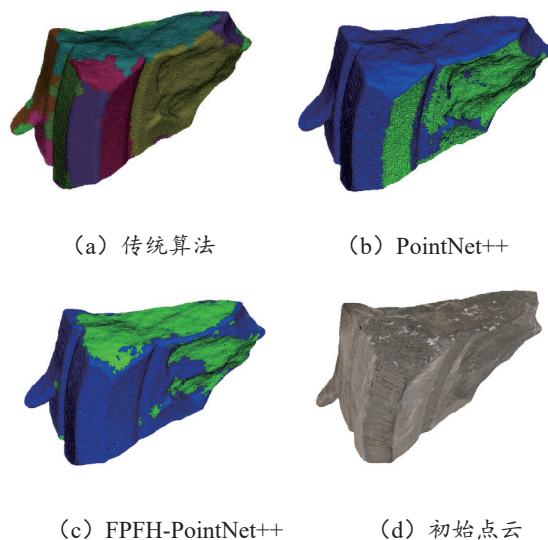


图 9 Precious 碎块 DoraArch2 分割效果

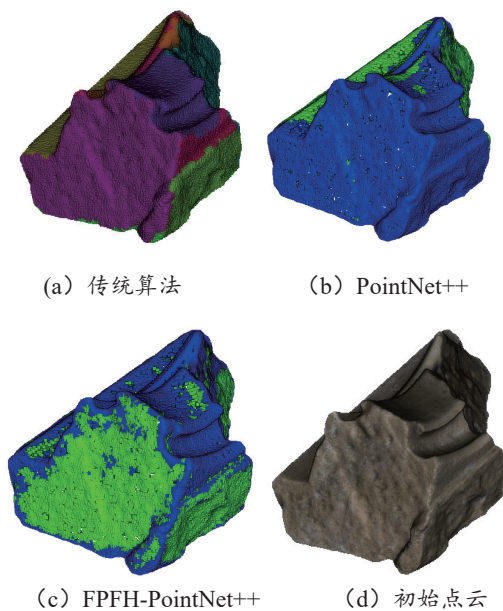


图 10 Precious 碎块 DoraEmbrasure5 分割效果

4 总结

本文在碎片拼接领域, 提出了一种把传统算法和深度学习网络相结合的碎片拼接方案。首先通过传统算法, 对 Frag-Tag3D 数据集进行处理, 计算出 FPFH, 然后将处理后的数据输入到 FPFH-PointNet++ 深度学习网络进行训练、检验、测试, 也在 Precious 数据集上评估该算法的泛化性。最后把提取出的断裂面上的点及其对应 FPFH 特征应用于传统匹配算法, 进行点云断裂面的配准。

不难看出, 这种思路用于其他断裂面上点的特征提取然后以该特征作为拼接配准依据算法中, 提取出来的特征不仅可以参与断裂面的拼接过程, 也可在断裂面拼接阶段前用到深度学习网络对非断裂面进行剔除中, 从而达到提高效率,

提高准确率。

但在进一步实验中发现, 本方法存在一定的局限性, 首先, 在碎片表面侵蚀比较严重, 断裂面上点的局部特征不明显的情况下, 断裂面分割准确率会出现明显下降, 这也是很多依靠断裂面提取特征作为配准算法的痛点。其次, 本文主要工作是基于深度学习网络进行整个断裂面和非断裂面之间分割, 而整个断裂面可能由多个小的断裂面组合而成。虽然完整分割后的断裂面轮廓较为简单, 再用传统算法进一步细化分割断裂面可以达到较好的效果, 但步骤未免繁琐, 因此可以进一步探索如何通过深度学习网络利用断裂面信息, 对断裂面的分割进行细化。

参考文献:

- [1]HUANG Q X, SIMON F, NATASHA G, et al. Reassembling fractured objects by geometric matching[M].New York: ACM Siggraph 2006 papers, 2006.
- [2]MAVRIDIS P, ANDREADIS A, PAPAIOANNOU G. Fractured object reassembly via robust surface registration[C]//Eurographics (Short Papers).[S.l.]:[s.n.], 2015: 21-24.
- [3]DU G G, ZHOU M Q, YIN C L, et al. Classification and reassembly of archaeological fragments[J].Proceedings of the Symposium on VR Culture and Heritage-Volume 2. 2016: 67-70.
- [4]JIA C Q, HE L G, YANG X W, et al. Developing a reassembling algorithm for broken objects[J]. IEEE access, 2020, 8: 220320-220334.
- [5]LI Q H, GENG G H, ZHOU M Q. Pairwise matching for 3D fragment reassembly based on boundary curves and concave-convex patches[J]. IEEE access, 2019, 8: 6153-6161.
- [6]SAVELONAS M A, ANDREADIS A, PAPAIOANNOU G, et al. Exploiting unbroken surface congruity for the acceleration of fragment Reassembly[C]//Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage.[S.l.]:[s.n.], 2017: 137-144.
- [7]GEORGIOU P, EVAGGELIA K. On the automatic assemblage of arbitrary broken solid artefacts[J]. Image and vision computing, 2003, 21(5): 401-412.
- [8]袁洁, 周明全, 耿国华, 等. 基于 Morse-Smale 拓扑特征的文物碎片拼接算法 [J]. 自动化学报, 2018,44(8):1486-1495.
- [9]何璐阳, 谢晓尧, 刘建成, 等. 基于轮廓曲线特征的破碎物体重组算法 [J]. 信息技术与信息化, 2024(2):79-83.
- [10]CHARLES R Q, SU H, MO K C, et al. Pointnet: deep learning on point sets for 3D classification and segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE, 2017: 652-660.
- [11]CHARLES R Q, SU H, LI Y, et al. Pointnet++: deep hierarchical feature learning on point sets in a metric space[DB/OL].(2017-06-07)[2024-08-21]. <https://doi.org/10.48550/arXiv.1706.02413>.
- [12]YAO W M, CHU T, TANG W L, et al. SPPD: a novel reassembly method for 3D terracotta warrior fragments based on fracture surface information[J]. ISPRs international journal of geo-information, 2021, 10(8): 525.
- [13]ROBERTO D L H, MAARTEN V. A hybrid approach to reassemble ancient decorated block fragments through a 3D puzzling engine[J]. Remote sensing, 2020, 12(16): 2526.
- [14]SU H T. FragTag3D: a 3D object fragment dataset with annotations for overlapping regions[EB/OL].[2024-07-21]. <https://bailanciaga.github.io/FragTag3D>.
- [15]ZHOU Q N, JACOBSON A. Thingi10K: a dataset of 10,000 3D-printing models[DB/OL].(2016-05-16)[2024-08-19]. <https://doi.org/10.48550/arXiv.1605.04797>.
- [16]THEOHARIS T. Papaioannou, PRESIOUS 3D cultural heritage fragments[EB/OL].[2024-08-09]. <https://www.ntnu.edu/presious/3d-data-sets>.

【作者简介】

郭子旭 (1996—), 男, 河北邯郸人, 硕士研究生, 研究方向: 计算机视觉、点云处理, email:15100025235@163.com。

谢晓尧 (1952—), 男, 贵州贵阳人, 博士, 教授, 研究方向: 网络通信、信息安全与人工智能, email:xyx@gznu.edu.cn。

刘嵩 (1983—), 男, 贵州贵阳人, 博士, 副教授, 研究方向: 计算机视觉、三维重建, email:songliu@gznu.edu.cn。

刘旭斌 (1997—), 男, 广东韶关人, 硕士研究生, 研究方向: 计算机视觉、深度学习, email:21010210660@gznu.edu.cn。

(收稿日期: 2024-09-03)