

DOI:10.3969/j.issn.1003-5060.2025.08.005

基于RBF神经网络的点云孔洞修复算法研究

张郭昌, 檀结庆, 彭凯军

(合肥工业大学 数学学院, 安徽 合肥 230601)

摘要:针对逆向工程中三维点云模型产生的孔洞问题,文章提出一种基于径向基函数(radial basis function, RBF)神经网络映射的点云孔洞修复优化算法。首先对点云数据进行预处理,检测孔洞边界点和拓展边界点,并对边界点进行平滑处理;然后将孔洞边界点投影到不同坐标轴平面上,选择投影面积最大的孔洞坐标系作为映射坐标系,将孔洞映射到坐标系中,使用支持向量机(support vector machine, SVM)进行初步修复;最后使用RBF神经网络映射点云,优化孔洞修复算法进行孔洞填充。实验结果表明,该算法能快速有效地填充孔洞,填充效果优于其他算法,且在填充孔洞的同时保证其拓扑性。

关键词:点云数据;孔洞填充;支持向量机(SVM);神经网络

中图分类号:TP391.41

文献标志码:A

文章编号:1003-5060(2025)08-1039-06

Research on point cloud hole repair algorithm based on RBF neural network

ZHANG Guochang, TAN Jieqing, PENG Kaijun

(School of Mathematics, Hefei University of Technology, Hefei 230601, China)

Abstract: Aiming at the problem of holes generated by three-dimensional point cloud model in reverse engineering, this paper proposes an optimized point cloud hole repair algorithm based on radial basis function (RBF) neural network mapping. Firstly, the point cloud data is preprocessed, the hole boundary points and the extended boundary points are detected, and the boundary points are smoothed. After that, the hole boundary points are projected onto different coordinate planes, and the hole coordinate system with the largest projection area is selected as the mapping coordinate system. The hole is mapped into the coordinate system, and the support vector machine (SVM) is used for preliminary repair. Finally, the RBF neural network is used to map the point cloud to fill the hole. Experimental results show that the algorithm can quickly and effectively fill holes, outperforming other algorithms in filling effect while preserving topology throughout the filling process.

Key words: point cloud data; hole filling; support vector machine (SVM); neural network

0 引言

使用三维激光扫描仪对物体进行扫描时可以得到三维点云数据集。然而,由于三维模型的不规则性,获取的点会出现部分缺失现象,从而导致点云数据的曲面重建不完整,在三维模型中产生孔洞。因此,为了保持三维模型的完整性,孔洞填

充成为三维重建中的关键步骤之一。

国内外研究学者针对此问题已经提出了多种孔洞填充算法,这些算法可以分为两大类,分别是基于三角网格的孔洞填充算法和基于散乱点云的孔洞填充算法。

三角网格填充孔的研究在几何建模、计算机图形学和图像处理等领域得到了广泛的研究和应

收稿日期:2023-05-06;修回日期:2023-05-26

基金项目:国家自然科学基金资助项目(62172135)

作者简介:张郭昌(1998—),男,湖北天门人,合肥工业大学硕士生;

檀结庆(1962—),男,安徽望江人,博士,合肥工业大学教授,博士生导师,通信作者, E-mail: jieqingtan@126.com.

用。文献[1]介绍了基于 Delaunay 三角剖分填充三角形网格孔洞的工作,通过三角剖分定位孔洞信息,获取孔洞领域的几何和拓扑信息,填充孔洞并细化孔;文献[2]提出了一种基于网格变形的修复算法,在识别了数字化的孔之后,保留了表面的曲率;文献[3]使用理想法线计算方案来近似计算新三角形的理想法线,根据理想法线和孔的边界顶点,通过解泊松方程得到每个新顶点的 3 个坐标;此外,径向基函数(radial basis function, RBF)的变换也被广泛用于三角形网格孔的填充中^[4-5]。

上述文献的方法是基于三角网格化操作,但重建网格较为耗时,且由于点云的散乱性,对带孔洞的点云模型进行三角剖分较为困难。基于此,文献[6-7]基于点云边界的特征线对点云进行操作填充孔洞;文献[8]利用粗糙的 Bezier 曲线来反求 Bezier 曲线的控制点,进行孔洞填充,最后对点进行细化处理,平滑曲面;文献[9-10]利用 RBF 的曲面优势来映射点云孔洞,进行孔洞填充;文献[11]将最小二乘和支持向量机(support vector machine, SVM)相结合来改进孔洞填充算法,通过粒子群算法优化参数,填充三维空间孔洞;文献[12]通过空间投影得到残缺曲线,通过 SVM 训练得到填充点;文献[13]在平面中采用蒙特卡洛随机取点来获取二维平面填充点,再使用 RBF 神经网络训练得到三维平面填充点。

基于此,本文提出一种基于 RBF 神经网络映射的点云孔洞修复优化算法,并通过实验进行对比,验证该算法的有效性。

1 基本理论

1.1 神经网络下点云的映射

神经网络下点云的映射是指在神经网络中输入点云数据之后,将点云映射到高维空间中,使得点云能转换成便于处理的形式。

点云的映射可以表示为:

$$\mathbf{Y}_i = \phi(\mathbf{X}_i | i \in N) \quad (1)$$

其中: \mathbf{X}_i 为第 i 个点的输入坐标; \mathbf{Y}_i 为第 i 个点的输出坐标; N 为点云数据集; $\phi(\cdot)$ 为映射函数或局部感知器。

通过将点云映射,可以对点云数据进行特征提取,并在高维空间中进行数据分析处理。

1.2 SVM 算法

SVM 是分类与回归分析中分析数据的监督学习模型及相关的学习算法。SVM 适用于有监

督的学习训练。

对于二维平面上的一组点云集 $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\}$, 将 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 和 $\mathbf{y} = (y_1, y_2, \dots, y_n)$ 分别作为输入向量和预期输出向量,利用函数 $\varphi(\mathbf{x})$ 对 \mathbf{x} 作映射处理,通过训练可以得到 SVM 的回归方程为:

$$f(\mathbf{x}) = \boldsymbol{\omega}^T \varphi(\mathbf{x}) + b \quad (2)$$

需要优化的目标为:

$$\min \left\{ \frac{1}{2} \|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \right\};$$

$$\text{s. t. } |y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| \leq \varepsilon + \xi_i \quad (3)$$

其中: ξ_i, ξ_i^* 为松弛变量; ε 为宽容系数。对点云集中某个点 (x_i, y_i) ;若 $|y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| < \varepsilon$, 则该点对损失的贡献为 0, 即无损失;若 $|y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| \geq \varepsilon$, 则模型在该点产生的损失为 $|y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| - \varepsilon$ 。

回归模型的损失函数可以表示为:

$$l_{\text{loss}} = \begin{cases} 0, & |y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| < \varepsilon; \\ |y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| - \varepsilon, & |y_i - \boldsymbol{\omega}^T \varphi(\mathbf{x}) - b| \geq \varepsilon \end{cases} \quad (4)$$

在点云集训练时,由于点云是不规则的、无序的模型,因此采用径向基核函数为映射函数,相较于 B 样条曲面更具有保形性和连续性。径向基核函数为:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma}\right), \quad \sigma > 0 \quad (5)$$

2 基于神经网络的点云映射填充算法

2.1 数据预处理

点云孔洞填充算法是基于孔洞边界的特异性来进行曲面拟合,因此孔的边界提取是孔填充的前提准备。对于散乱点云,拟合点云邻域的切平面,将邻域点投影到局部切平面上,通过判断点云的邻域投影点与该点之间的夹角大小来判断该点是否为边界点。

找到孔边界后,计算孔边界点的平均距离 d 。由于孔边界点的信息较少,寻找边界点之间距离小于阈值(阈值设置为 d)的邻域作为扩展边界点,以尽可能完全地保留点云孔洞邻域特征的信息。

对空间点云做降维处理,将点云投影到不同坐标平面上。由于在坐标系上投影时孔的面积不同,根据平面上的孔洞边界建立 Delaunay 三角网,计算三角片的总面积并作为孔洞投影面积。

选择孔洞投影面积最大的坐标系并作为投影坐标系,以最大程度地确保特征的完整性。

由于孔洞的无序性和散乱性,提取出的孔边界会出现特征丢失等症状。为了增强相邻点的相关性,对点云边界采用中值平滑操作,即对边界点排序,记排序之后的点集为 N ,若 N 中相邻两点 $P_i(x_i, y_i)$ 与 $P_j(x_j, y_j)$ 之间的距离大于 d ,则加入平滑点对边界进行平滑处理,平滑点 $P_c(x_c, y_c)$ 的坐标计算公式如下:

$$\begin{aligned} x_c &= \frac{x_i + x_j}{2}, \\ y_c &= \frac{y_i + y_j}{2} \end{aligned} \quad (6)$$

2.2 SVM 孔洞修复

将点投影到平面后,在平面上使用 SVM 对点进行初步修复。在文献[12]方法的基础上进行优化,使用区间划分的思想代替扫描线来进行初步填充,增强平面填充的鲁棒性(下文以投影到 xOy 平面为例),具体算法过程如下:

1) 记孔洞边界点集和孔洞扩展边界点集为 P_b 和 P_{eb} ,找到 P_b 中在 x 方向上的最大值、最小值,分别记为 x_{\max} 、 x_{\min} 。

2) 遍历 P_{eb} 中的点,若点的 x 值没有落在区间 (x_{\min}, x_{\max}) 内,则认为该点对孔洞邻域特征信息没有贡献,将该点删除;若该点落在区间 (x_{\min}, x_{\max}) 上,表示该点为孔洞邻域的特征点,将该点保留。重复操作,直到 P_{eb} 中所有点都在区间 (x_{\min}, x_{\max}) 内。

3) 在 P_b 中沿着 x 方向寻找与 x_{\min} 对应的点 P_{\min} 和与 x_{\max} 对应的点 P_{\max} ,定义间隔步长为:

$$d_x = \frac{x_{\max} - x_{\min}}{n/2} \quad (7)$$

其中, n 为 P_b 中点的总个数。

4) 将 (x_{\min}, x_{\max}) 按照步长划分为不同的区间,即 $(x_{\min}, x_{\min} + d_x), \dots, (x_{\max} - d_x, x_{\max})$,令 $R = P_b \cup P_{eb}$,对于 R 中的点,根据点的 x 值所在的区间不同,将点划分为不同的样本区间。由于点云的密集性,每一组样本点都可认为是一条残缺的曲线,残缺部分即为点云缺失处。

5) 对每一组样本点 $P_i^k, k \geq 0, 0 \leq i \leq n$,其中 k 为样本的序号, n 为样本点的数量。以 y 为输入标签, x 为输出标签,在 SVM 中对样本点进行训练,得到回归模型。之后,对样本点按照 y 值大小进行排序,计算相邻点之间的 y 坐标差值,则差值最大的 2 个点即为残缺曲线端点。选取残

缺曲线端点 $P_m^k(x_m, y_m)$ 和 $P_n^k(x_n, y_n)$ 作为边界点,计算样本点之间的平均距离,记为 d_p ,则填充点的个数为:

$$n = \left\lceil \frac{|y_m - y_n|}{d_p} \right\rceil \quad (8)$$

其中, $\lceil \cdot \rceil$ 表示向上取整。

6) 根据填充点个数,在 P_m^k 与 P_n^k 之间按照 y 值选取一组均匀分布的向量 \mathbf{y} ,将 \mathbf{y} 加入模型,训练得到填充点的 x 。由此可以得到 xOy 平面内的填充点 (x, y) 。

SVM 初步填充效果如图 1 所示。图 1 中: P_{bi} 为孔洞边界点; P_{ebi} 为孔洞拓展边界点。

经过区间划分将 P_b 和 P_{eb} 划分成 3 组不同的训练样本,经过 SVM 训练之后,按照上述步骤可以得到 3 组残缺线的填充点 P_i ,即为经过 SVM 训练的初步孔洞填充点。

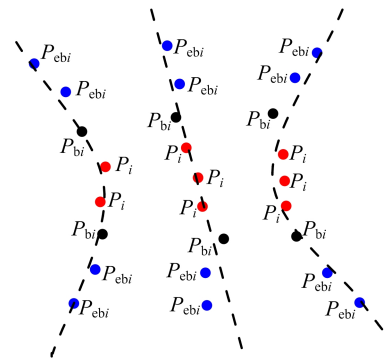


图 1 SVM 初步填充

2.3 基于 RBF 神经网络的孔洞修复优化算法

RBF 神经网络可以使用较少的节点来表示输入空间的复杂非线性关系,适用于非线性回归问题。RBF 神经网络是由输入层、隐藏层、输出层组成的 3 层前馈神经网络。

隐藏层的激活函数采用高斯函数,其表达式如下:

$$\varphi_j(\mathbf{X}) = \exp\left(-\frac{\|\mathbf{X} - \mathbf{C}_j\|^2}{2\sigma_j^2}\right) \quad (9)$$

其中: φ_j 为隐藏层第 j 个单元的输出; \mathbf{X} 为输入矢量; \mathbf{C}_j 为隐藏层第 j 个高斯单元的中心矢量; σ_j 为宽度函数。

RBF 神经网络的输出可表示为:

$$\mathbf{Y} = \mathbf{W}\Phi \quad (10)$$

其中: \mathbf{Y} 为输出矢量; \mathbf{W} 为隐藏层到输出层的权矩阵; Φ 为隐藏层的输出量。

RBF 神经网络的学习算法主要是通过将样

本映射到高维空间,在高维空间中建立超平面模型,从而求解权矩阵 W ,得到输出矢量。

对于三维空间中的孔洞边界点和拓展邻域点集 (x, y, z) ,以 (x, y) 为输入矢量 X , z 为输出矢量 Y ,使用 RBF 神经网络进行训练,得到神经网络训练模型。

将初步填充获得的孔洞点坐标作为输入矢量,在 RBF 神经网络模型中进行训练,可以得到一组向量 z ,由此便可得到孔洞填充点的三维坐标。

3 实验与数据分析

3.1 实验过程

为了评估本文的点云孔洞修复优化算法,基于 MATLAB 对点云数据进行实验,使用经典的点云数据库,并在计算机上运行。

实验采用 AMD R7 4800H, 16 GiB 内存, win10 操作系统, MATLAB R2018b, 在各种数据集上测试本文算法。

选择斯坦福兔子来验证本文算法的可行性,结果如图 2 所示。

从图 2 可以看出,本文算法在视觉上具有良好的填充效果。

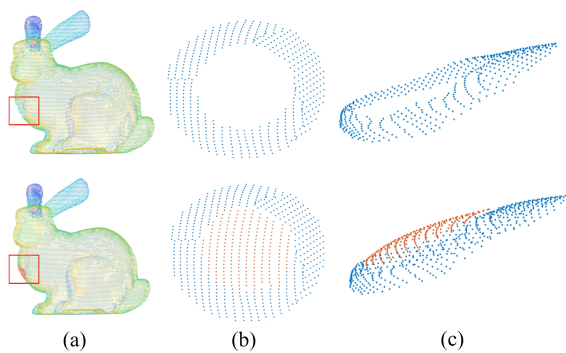


图 2 斯坦福兔子胸部孔填充

3.2 对比分析

为了进一步测试本文算法的优劣性,选取文献[11]、文献[12]、文献[13]和本文的点云孔洞填充算法进行对比实验。首先对兔子背部的孔洞进行实验,得到的填充效果如图 3 所示。

从图 3 可以看出,文献[11]和文献[13]填充的孔洞点具有随机性,填充点不具备拓扑性;而文献[12]填充的算法虽然具有一定的结构性,但由于扫描线过于密集,使得填充效果较差;本文提出的算法具有较强的拓扑性,在恢复激光扫描的有序散乱点云上具有很大的优势。

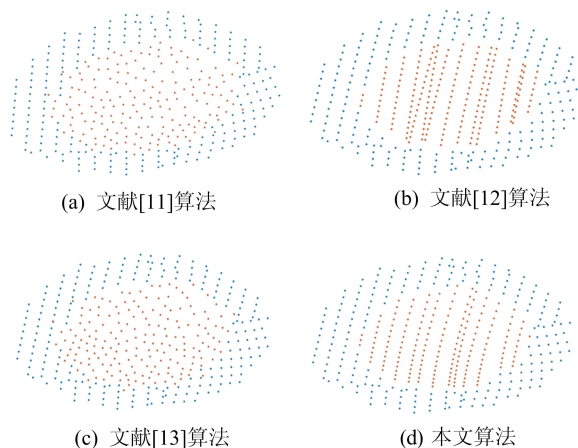


图 3 兔子背部不同算法填充效果对比

选取大象的前腿孔洞作为实验对象,通过文献[11]、文献[12]、文献[13]和本文的点云孔洞填充算法得到的填充效果如图 4 所示。

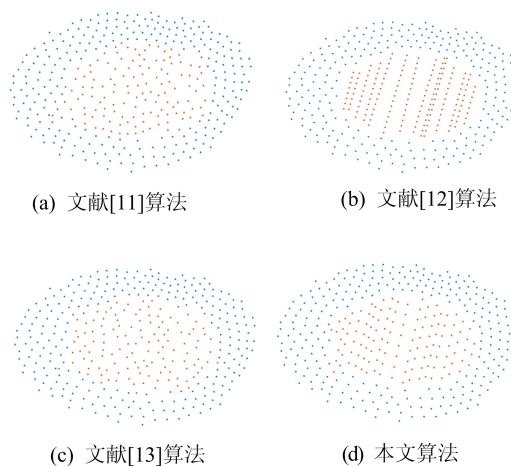


图 4 大象前腿不同算法填充效果对比

从图 4 可以看出,在无序散乱点云的情况下,文献[11]、文献[13]和本文算法在视觉上的填充效果较好,而文献[12]算法填充点过于密集,填充效果较差。

因此,本文提出的算法在散乱点云上均有着不错的视觉效果,填充效果较好。

3.3 数据分析

为了定量分析点云的填充效果,保留了孔的原始点信息,本文定义 2 种误差来比较和分析实验结果。定义双边 hausdorff 距离,设存在 2 个点集 X, Y ,则双边 hausdorff 距离 H 的计算公式如下:

$$H_1 = \sup_{x \in X} \inf_{y \in Y} d(x, y),$$

$$H_2 = \sup_{y \in Y} \inf_{x \in X} d(x, y),$$

$$H = \max\{H_1, H_2\} \quad (11)$$

其中: $d(x, y)$ 表示 x 与 y 两点之间的距离; \sup 和 \inf 分别表示上确界和下确界。

使用双边 hausdorff 距离来判断孔洞边界点集和孔洞填充点的贴合性,令 H_o 、 H_r 分别表示原始点集和填充点集到孔洞边界点的双边 hausdorff 距离,若 H_o 与 H_r 越接近,则表示填充点与孔洞边界点越贴合,填充效果越好。

同时定义填充点与原始点云的拟合程度,使用并改进了文献[14]中提到的点到表面距离的面积加权积分,具体步骤如下:

1) 对原始点云和点云邻域执行 Delaunay 三角剖分,以获得三角剖分的表面积 S_i ($i=1, 2, \dots, n$),其中 n 表示三角片的数量。

2) 遍历填充点集,对填充点集中的每个点,

计算点到最近三角片的距离,取平均值为 d_{avg} 。

3) 计算三角片的平均面积,记为 S_{avg} 。

4) 定义拟合误差,即

$$E_m = d_{avg}^2 / S_{avg} \quad (12)$$

点与三角片之间的距离越近,则认为点的特征与原始点越接近,点云填充的效果越好。

使用上述误差作为评价指标,在不同的模型上进行实验,部分实验效果图如图 5 所示。

文献[11]、文献[12]、文献[13]和本文算法的拟合误差数据见表 1 所列,双边 hausdorff 距离见表 2 所列。

从表 1 可以看出,本文算法的拟合误差比文献[11]、文献[12]、文献[13]算法的更小,即本文算法填充的孔洞与原始点云拟合程度更高,空间结构更相似。

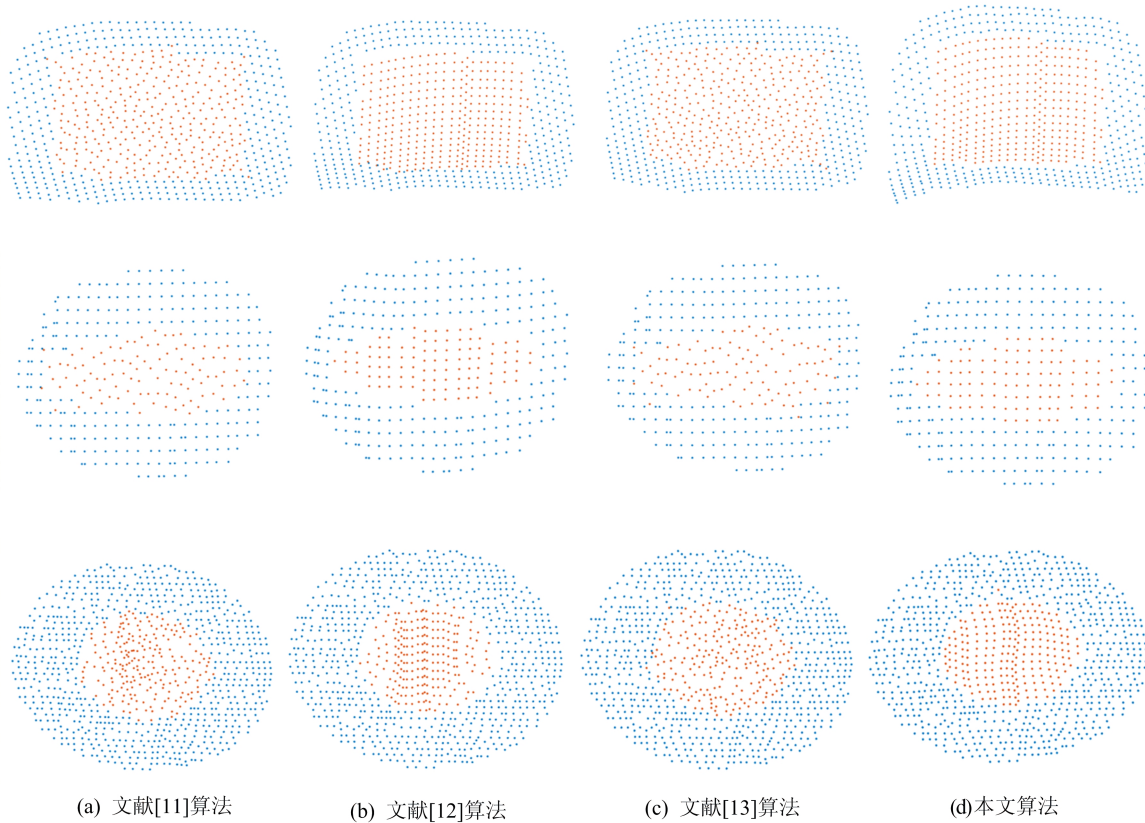


图 5 部分实验效果图

表 1 不同算法的拟合误差

算法	马孔 1	马孔 2	兔子孔 1	兔子孔 2	象孔 1	象孔 2	手孔 1	手孔 2	鹿孔 1	鹿孔 2
文献[11]算法	2.67E-6	4.28E-5	2.26E-1	8.68E-3	9.06E-2	5.58E-5	1.59E-5	2.82E-4	2.76E-4	6.79E-4
文献[12]算法	1.76E-5	1.60E-4	1.03E-1	1.91E-3	2.80E-3	6.14E-6	7.63E-6	6.43E-4	1.84E-4	4.60E-5
文献[13]算法	2.34E-6	7.02E-5	2.11E-2	3.40E-3	1.26E-3	6.98E-6	1.17E-5	3.39E-4	1.53E-4	1.10E-4
本文算法	2.07E-6	2.14E-5	3.39E-3	1.16E-4	8.14E-4	4.85E-6	6.67E-6	2.79E-4	6.90E-5	3.91E-5

表 2 不同算法双边 hausdorff 距离

填充位置	马孔 1	马孔 2	兔子孔 1	兔子孔 2	象孔 1	象孔 2	手孔 1	手孔 2	鹿孔 1	鹿孔 2
原始点	0.007 764	0.003 595	0.012 270	0.009 888	0.061 23	0.047 42	0.110 10	0.091 18	0.039 33	0.037 28
文献[11]算法	0.007 583	0.003 880	0.011 290	0.010 510	0.085 01	0.046 06	0.100 50	0.088 85	0.032 24	0.038 45
文献[12]算法	0.007 667	0.003 909	0.011 840	0.009 823	0.057 55	0.046 99	0.107 30	0.091 46	0.035 71	0.038 11
文献[13]算法	0.007 628	0.003 882	0.011 390	0.010 100	0.059 29	0.046 01	0.106 00	0.092 27	0.032 55	0.035 15
本文算法	0.007 691	0.003 706	0.011 920	0.010 570	0.059 05	0.047 14	0.109 90	0.089 30	0.035 94	0.037 94

表 2 为原始点以及不同算法填充点与孔洞边界的双边 hausdorff 距离。填充点与孔洞边界的双边 hausdorff 距离与原始点越接近,则表示填充点与孔洞边界点衔接较为平滑,填充效果好。由表 2 中数据可知,本文算法相较于其他算法绝大多数情况下效果更好,与孔洞边界衔接较好。

4 结 论

本文研究了基于 RBF 神经网络映射的点云孔洞修复优化算法。将点云投影到平面上,通过 SVM 对孔洞进行初步填充;再采取 RBF 神经网络来优化孔洞修复算法,得到孔洞填充点的三维坐标。实验结果表明,本文算法得到的填充点与原有孔洞边界较为契合,填充效果较好,且保持了较好的拓扑性。同时,将本文所提出的算法与其他点云孔洞填充算法进行对比,结果表明,本文算法填充效果优于其他算法。

[参 考 文 献]

- [1] FORTES M A, GONZALEZ P, PALOMARES A, et al. Filling holes using a mesh of filled curves[J]. *Mathematics and Computers in Simulation*, 2019, 164: 78-93.
- [2] QUINSAT Y, LARTIGUE C. Filling holes in digitized point cloud using a morphing-based approach to preserve volume characteristics[J]. *The International Journal of Advanced Manufacturing Technology*, 2015, 81: 411-421.
- [3] ZHAO W, GAO S, LIN H. A robust hole-filling algorithm for triangular mesh[J]. *The Visual Computer*, 2007, 23: 987-997.
- [4] BRANCH J W, PRIETO F, BOULANGER P. Automatic hole-filling of triangular meshes using local radial basis function[C]//Third International Symposium on 3D Data Processing, Visualization and Transmission, [S. l.]: IEEE, 2006: 97-111.
- [5] WU X M, LI G X, ZHAO W M. Hole filling algorithm in surface reconstruction based on radial basis function neural network[J]. *Key Engineering Materials*, 2009, 392/393/394: 750-754.
- [6] ALTANTSETSEG E, KHORLOO O, MATSUYAMA K, et al. Complex hole-filling algorithm for 3D models[C]//Proceedings of the Computer Graphics International Conference. Yokohama: ACM, 2017: 1-6.
- [7] WANG Y, TANG J, ZHAO Y, et al. Point cloud hole filling based on feature lines extraction[C]//2017 International Conference on Virtual Reality and Visualization (ICVRV). [S. l.]: IEEE, 2017: 61-66.
- [8] NGUYEN V S, TRAN K M, TRAN M H. Filling hole on the surface of 3D point clouds based on reverse computation of bezier curves[C]//Information Systems Design and Intelligent Applications. Singapore: Springer, 2018: 334-345.
- [9] 晏海平, 吴禄慎, 陈华伟. 基于径向基函数的散乱点云孔洞修复算法[J]. *计算机工程与设计*, 2014, 35(4): 1253-1257.
- [10] 陈飞舟, 陈志杨, 丁展, 等. 基于径向基函数的残缺点云数据修复[J]. *计算机辅助设计与图形学学报*, 2006(9): 1414-1419.
- [11] 杨永强, 李淑红. 最小二乘支持向量机的点云数据孔洞修补算法[J]. *吉林大学学报(理学版)*, 2018, 56(3): 692-696.
- [12] 蒋刚. 基于 SVM 和空间投影的点云空洞修补方法[J]. *计算机工程*, 2009, 35(22): 269-271.
- [13] 张艺真, 孙志毅, 柏艳红, 等. 基于 RBF 神经网络的三维点云数据孔洞修补[J]. *太原科技大学学报*, 2022, 43(1): 23-28.
- [14] PAULY M, GROSS M, KOBELT L P. Efficient simplification of point-sampled surfaces[C]//IEEE Visualization. Boston: IEEE, 2002: 163-170.

(责任编辑 胡亚敏)