

基于多标签分类的点云快速批量三维重建

宋婉婷, 江文松*, 罗哉**

中国计量大学计量测试工程学院, 浙江 杭州 310018

摘要 传统的三维重建方法大多侧重于物体表面细节的保留, 在需要快速完成反映物体类别、形状等典型特征点云三维重建工作中, 其效率和自动化程度较低。针对这一问题, 提出一种基于多标签分类的点云批量快速三维重建方法。对三维点云采取先简化再去噪的预处理, 提出一种基于双阈值约束优化的 k 邻域去噪算法, 并将其去噪效果与两种现有方法进行对比评价。利用动态图像卷积神经网络(DGCNN)对预处理后的三维点云进行分类。通过类别匹配一一调用已创建的三维模型库中对应类别的重建结果, 实现点云的批量快速三维重建。若三维模型库中没有对应的类别, 则获取相应类别的三维点云, 加入 DGCNN 中重新训练, 完成三维重建并加入三维模型库, 实现类别的扩充。利用增量 ModelNet40 模型对本文方法进行实际应用验证, 结果表明, 本文方法重建 120 个 10 万数量级的三维点云耗时 11.98 s, 重建 120 个 70 万数量级的三维点云耗时 29.06 s, 重建 120 个 120 万数量级的三维点云耗时 109.98 s, 比传统方法效率提高了近 10 倍。本文方法对于数量级较大、数量较多的三维点云均有明显的效率提升, 可实现点云快速批量三维重建。

关键词 图像处理; 三维重建; 点云分类; 卷积神经网络

中图分类号 TP399

文献标志码 A

doi: 10.3788/LOP202158.1210001

Rapid Batch Three-Dimensional Reconstruction of Point Clouds Based on Multi-Label Classification

Song Wanting, Jiang Wensong*, Luo Zai**

College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou, Zhejiang 310018, China

Abstract Existing traditional three-dimensional (3D) reconstruction methods focus on preserving surface details of 3D objects; however, they cannot realize rapid reconstruction of 3D objects but can only reflect the category, shape, and other typical features of 3D objects. To tackle this problem, a rapid batch method for 3D object reconstruction is proposed. First, point clouds are simplified before they are denoised; an enhanced k -neighbor denoising algorithm with double-threshold constraints is proposed. The denoising performance of this algorithm is compared with that of two traditional denoising methods. Second, the categories of 3D point clouds are obtained using the dynamic graph convolutional neural network(DGCNN). Finally, after matching each category with the categories in a pre-built 3D model library, corresponding 3D models are called sequentially to achieve the batch rapid 3D reconstruction. If there is no corresponding category in the pre-built 3D model library, the 3D point clouds of the corresponding category can be acquired and added to the DGCNN for training and evaluating. The 3D reconstruction result of each category is added to the pre-built 3D model library to increase the number of categories in the library. The effectiveness of our proposed method is verified using the incremental ModelNet40 model. The experimental result shows that the 3D reconstruction method incurs the cost of 11.98 s for 120 point clouds of 100000 points, 29.06 s for 120 point clouds of 700000 points, and 109.98 s for 120 point clouds of 1200000 points, which is approximately 10 times faster than

收稿日期: 2020-08-20; 修回日期: 2020-09-29; 录用日期: 2020-10-14

基金项目: 国家重点研发计划(2019YFB2004900, 2018YFF01012006, 2017YFF0206306)、国家自然科学基金(51675499, 51927811)、浙江省重点研发计划(2020C01096)、浙江省自然科学基金(LQ20E050016)

* E-mail: jwensong@cjlu.edu.cn; ** E-mail: luozai@cjlu.edu.cn

that of the traditional method considered in this study. Overall, this method can significantly improve the efficiency of 3D reconstruction for point clouds with both large order of magnitude and a large number, as well as realize real-time batch rapid 3D reconstruction of point clouds.

Key words image processing; three-dimensional reconstruction; point cloud classification; convolutional neural network

OCIS codes 100.6890; 110.4280; 150.3040

1 引言

三维重建是利用计算机技术对现实物体表面进行三维虚拟建模^[1],它是虚拟现实、逆向工程、几何造型等领域的重要理论基础^[2]。在当前快速发展的数字化车间等涉及机器视觉和跨线规划的应用场景中,机器人对所处环境的三维重建需要快速完成且只需要反映物体的类别、形状等简单特征,以便实现后续的避障等功能。传统的三维重建方法由于更侧重对单一物体表面细节的保留,存在耗时较长、效率不足等问题^[3],不适用于以上工况。因此,怎样提高以上工况中的三维重建效率,是数字化工业推广过程中需主要考虑的问题。

传统的三维重建过程的一般流程是:数据采集、点云去噪、点云简化、三维重建等^[4]。其中,点云去噪的结果是影响三维重建效果的主要因素之一。按照分布规律可以将噪点分为密集噪点、漂移噪点和混杂噪点,对重建效果影响最大的密集噪点和漂移噪点在工程上通常需采用手动删除^[5],效率较低且无法实现自动化操作。对于混杂噪点,文献[6]提出的统计滤波算法与文献[7]提出的半径滤波法均为常用的点云去噪算法。

在提高三维重建的效率时,目前常用的方法一是加速立体匹配的过程^[8-9],二是加速图像语义分割的过程^[10]。但以上两种方法大多针对二维图像,无法很好地适配三维点云的重建。相比于二维图像,三维点云维度更多,对物体的信息描述更加全面与具体^[11]。文献[12-15]通过改进三维重建的方法来提高重建效率,但提升效果有限,且仍不能完成自动

批量重建。目前还没有成熟的、统一的方法来提高三维点云的重建效率。随着神经网络的发展,利用深度学习直接处理三维点云数据的神经网络^[16-18]也越来越多。PointNet^[19]通过独立处理每个点来聚合特征,但是忽视了点与点之间的局部特征的学习;PointNet++^[20]是PointNet的改进算法,但因考虑到置换不变性,损失了部分特征信息;DGCNN^[21]提出一个新的神经网络模块 Edge-Conv 层,用于获取局部信息与全局信息,效果更好。

基于上述信息,针对传统三维重建方法不适用于数字化工业中机器视觉的批量快速三维重建这一问题,本文改进传统方法的重建流程,将卷积神经网络引入三维重建中,对预处理后的三维点云实行快速自动分类,并通过分类结果调用模型,完成三维点云的批量快速三维重建。并且可通过扩充已有三维模型库的操作,解决模型库中没有对应类型的情况。解决了在面对多个待重建三维点云时,需逐个进行三维重建的问题,并有效地提高了重建效率,为包含大量重复物体场景的点云批量快速重建提供了可行的算法支持。

2 三维点云批量快速处理

2.1 点云数据预处理

在点云预处理过程中,针对目前需要人工删除的漂移噪点与密集噪点,提出一种基于双阈值约束优化的 k 邻域去噪算法,并对输入神经网络中的三维点云采取先点云简化再点云去噪的预处理,减少了遍历所需时间,提高了点云预处理的效率。整体预处理流程如图 1 所示。

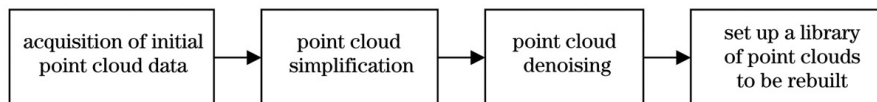


图 1 用于神经网络的三维点云的预处理流程

Fig. 1 Preprocessing flow chart of 3D point clouds for neural network

体素化网格算法^[22]通常用于点云数据的下采样处理。如图 2 所示,该方法是对输入的目标点云创建一个三维体素栅格,在网格划分的每个体素内,用其中的点云数据的重心代替该体素内的全部点,

提取重心点组成精简后的点云数据。该方法在减少点云数量的同时仍能保持物体的形状特征不被破坏^[10]。

在三维点云中,被测物体主体点云相邻点之间

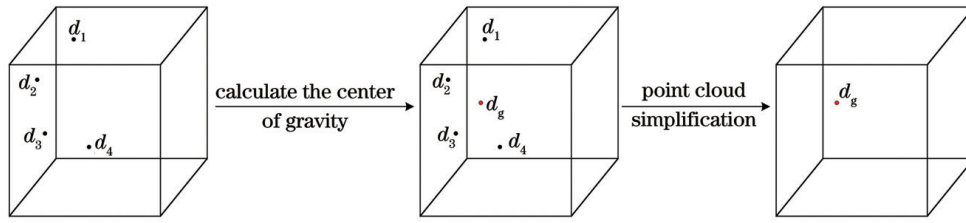


图 2 体素化网格算法精简点云示意图

Fig. 2 Schematic diagram of voxel grid algorithm

分布密集、间距较小；在主体点云周围存在混杂噪声点，其与相邻点之间分布稀疏、间距较大；漂移噪声点和密集噪声点则距离主体点云较远，且漂移噪声点分布稀疏，密集噪声点分布稠密。如图 3 所示，方框圈出的部分为主体点云，椭圆圈出的部分则为三类噪声点云。

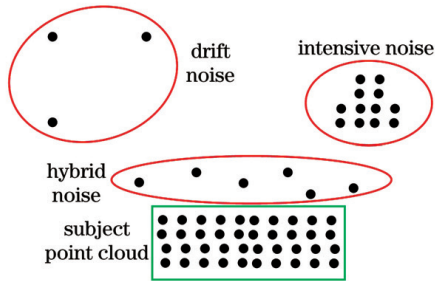


图 3 主体点云与噪声点云

Fig. 3 Subject point cloud and noise point cloud

k 邻域去噪算法^[23]便是基于以上的特点，通过 k_d -tree(空间二分数的一种特殊情况)对散乱点云建立拓扑关系，找到距离每个任意点最近的 k 个点，通过统计分析来移除噪声点。但 k 邻域去噪算法只能剔除混杂噪声点和漂移噪声点，不能剔除密集噪声点，密集噪声点只能依靠手动删除。针对这一问题，本文提出了基于双阈值约束优化的 k 邻域去噪算法，根据主体点云数量级较大、密集噪声点距主体点云存在一定距离，可知主体点云中的任意点与其他所有点的距离的平均值较密集噪声点中的任意点与其他所有点的距离的平均值有明显的差异。因此，通过遍历全部点云得到平均距离，判断其是否在阈值范围内，即可去除 k 邻域去噪算法无法剔除的密集噪声点。具体算法如下：

1) 假设任意点 X_i 的三维坐标为 (x_0, y_0, z_0) , $i=1, 2, \dots, n$, 其 k 邻域中的点为 (x_i, y_i, z_i) , $i=1, 2, \dots, k$, 则 X_i 到邻域中 k 个点的平均距离 D_i 为
$$\sum_{i=1}^k \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} / k$$
, 设 D_i 服从高斯分布, 即 $D_i \sim N(\mu_1, \sigma_1^2)$, 其中 $N(\cdot, \cdot)$ 表示正态分布, μ_1 为均值, σ_1 为标准差。计算得到

的 D_i 如果不在阈值 $T_1 \in (\mu_1 - a \cdot \sigma_1, \mu_1 + a \cdot \sigma_1)$ 范围内, 则对应的点云数据被判断为噪声点, 进行剔除。该步骤可初步剔除漂移噪声点与混杂噪声点, 得到的去噪结果为初步的主体点云。

2) 在当前初步主体点云内, 假设目标点 X'_i 的三维坐标为 (x'_0, y'_0, z'_0) , $i=1, 2, \dots, m$, 其他点的三维坐标为 (x'_i, y'_i, z'_i) , $i=1, 2, \dots, m-1$, 同理可得 X'_i 到其他 $m-1$ 个点的平均距离 D'_i 为

$$D'_i = \frac{\sum_{i=1}^{m-1} \sqrt{(x'_i - x'_0)^2 + (y'_i - y'_0)^2 + (z'_i - z'_0)^2}}{m-1}, \quad (1)$$

其中 m 为初步主体点云总个数, m 个点云数据到其他 $m-1$ 个点的平均距离的平均值为

$$\mu_2 = \frac{\sum_{i=1}^m D'_i}{m}. \quad (2)$$

假设任意点到其他点的平均距离服从正态分布, 即 $D'_i \sim N(\mu_2, \sigma_2^2)$, 将去噪前后的点云数据简化为均值与标准差均不变的系统, 其中, μ_2 为均值, σ_2 为标准差,

$$\sigma_2 = \sqrt{\frac{\sum_{i=1}^m (D'_i - \mu_2)^2}{m}}. \quad (3)$$

计算得到的 D'_i 不在 $T_2 \in (\mu_2 - a \cdot \sigma_2, \mu_2 + a \cdot \sigma_2)$ 范围内时对应的点云数据则被判断为噪声点, 进行剔除, 得到最终的主体点云。

3) 基于先验经验, 通常令 $a=1$, 此区间内的点云数据占全部点云数据的 68.2%, 可满足大部分点云去噪的需要^[24]。

2.2 三维点云分类

将 DGCNN 中提出的神经网络模块 Edge-Conv 层引入 PointNet 中可直接获取三维点云的局部信息与全局信息, 用于三维点云的分类。本文方法中的 DGCNN 的网络结构如图 4 所示。

以 n 个三维点作为输入, 输入表示为 $X = \{x_1,$

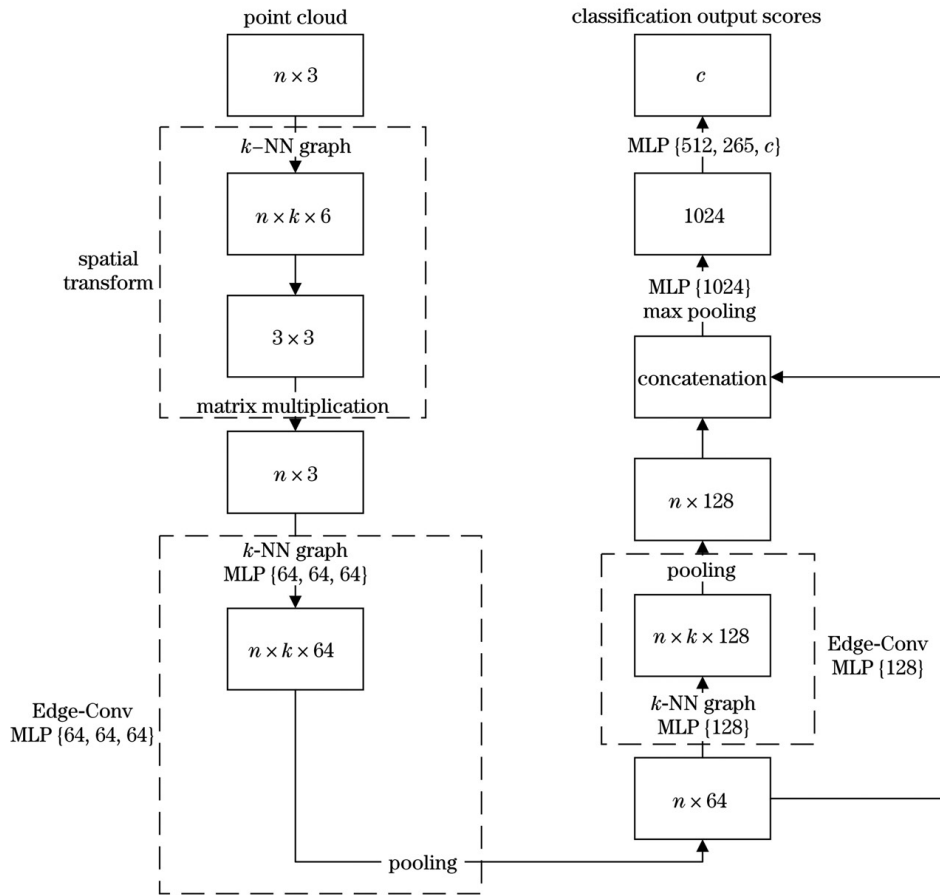


图 4 点云分类的神经网络结构

Fig. 4 Model architecture for point clouds classification

$x_2, \dots, x_n \subseteq \mathbb{R}$, 利用两个 Edge-Conv 层计算得到最终的边缘特征集 ($n \times k \times 128$), 在每个集合中聚合特征, 生成 $n \times 128$ 的张量。然后利用全连接层实现 Edge-Conv 层输出特性的全局聚合。最后通过池化层形成一个全局描述符, 用于生成 c 类的分类分数。在计算边缘特征时, 将全局特征^[25]与局部特征结合, 定义边缘特征 $e_{ij} = h_{\theta}(x_i, x_j)$, 其中 $h_{\theta}: \mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}^F$ 是一个带有一组可学习参数 θ 的非线性函数 (本文 $F=3$), 采用非对称边缘函数来完成 h_{θ} 和其特征点的选择, 即

$$h_{\theta}(x_i, x_j) = h_{\theta}(x_i, x_j - x_i), \quad (4)$$

$$x'_i = \max_{j: (i,j) \in B} h_{\theta}(x_i, x_j), \quad (5)$$

$$e_{ijm} = \theta_m \cdot [x_j + T - (x_i + T)] + \phi_m \cdot (x_i + T) = \theta_m \cdot (x_j - x_i) + \phi_m \cdot (x_i + T), \quad (6)$$

式中: x_i, x_j 为三维点云 X 中的任意两点; $j: (i, j)$ 表示 j 在 (i, j) 的范围内取值; θ_m 与 ϕ_m 为函数 h_{θ} 分别在 m 个不同过滤器下学习得到的参数; B 为局部点云结构的边。最后利用共享多层感知器

(MLP) 实现 $\Theta = (\theta_1, \theta_2, \dots, \theta_m, \phi_1, \phi_2, \dots, \phi_m)$ 时的分类结果。

该算法符合置换不变性与平移不变性的特征, 在翻转与平移的同时, 不会丢失边缘特征点, 有较好的提取效果。

同时, 由于在每一层都有一个不同的有向图 $G^{(l)} = (A^{(l)}, B^{(l)})$, 其中 $A = \{1, 2, \dots, n\}$ 表示点云结构中的点, $B \subseteq A \times A$ 表示点云结构的边, 第 l 层的边缘 $(i, j_{i1}), (i, j_{i2}), \dots, (i, j_{ik_l})$ 取决于 $x_i^{(l)}$ 的 k_l 个邻近点 $x_{j_{i1}}^{(l)}, x_{j_{i2}}^{(l)}, \dots, x_{j_{ik_l}}^{(l)}$, 因此 DGCNN 是在每一层中学习构建不同的 G , 而不是在评估网络之前以固定不变的参数构建, 这使其在深度学习过程中可以兼顾输入数据的局部特征和整体特征, 从而获得比 PointNet 与 PointNet++ 两种算法更合理的分类结果。

2.3 三维重建

本研究利用 Bowyer-Watson 算法^[26]来实现 Delaunay 三角化, 如图 5 所示。首先, 输入待重建物体的散乱点云, 建立一个能容纳所有点的大三角

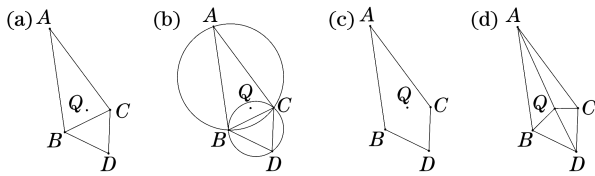


图 5 Delaunay 三角剖分步骤。(a)插入点;(b)空外接圆判断公共内边;(c)删除内边;(d)连接新点完成插入

Fig. 5 Delaunay triangulation. (a) Insert point; (b) judgement of the common inner edge using empty circumcircle; (c) delete the inner edge; (d) insert completed

形;依次将点集中的点插入,利用空外接圆特性判断

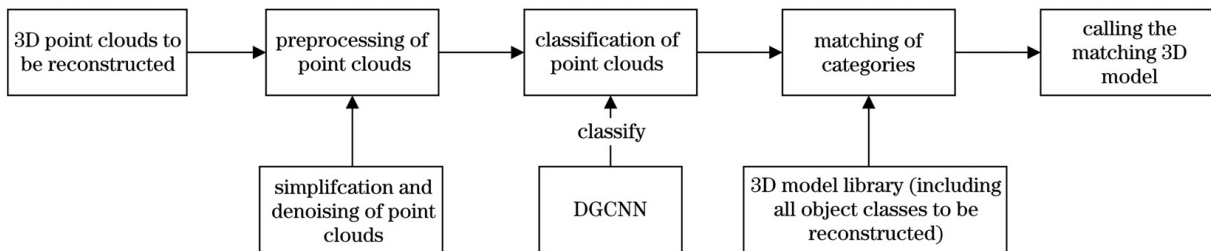


图 6 总体流程图

Fig. 6 Overall flow chart

3 实验设计与结果分析

本实验以六自由度关节臂装配三维非接触激光扫描系统为平台,所使用的 Scanworks 激光扫描系统测量精度为 0.024 mm (2σ), 重复精度为 0.005 mm (2σ), 其中 σ 为标准差。利用该激光扫描系统对待重建的室内场景进行扫描,并通过点云分割得到批量的待重建三维点云。所使用的三维点云类别在 ModelNet40 模型的基础上,增加球体、圆柱体、长方体、贯穿体 4 种。将所有三维点云分别按类别标号并保存。利用激光扫描获得以上 44 种类别物体的三维点云共 8000 组,按 4 : 1 的比例分为训练集与评估集,进行三维点云分类的训练与评估。经多次实验后发现,在训练次数达到 200 次时,分类准确率逐渐趋于平稳,达 90.7%,准确率相对较高,因此设定本模型训练次数为 200 次。此外,事先重建出 44 个类别的标准三维模型,分别保存对应类别,待批量重建时进行调用。

3.1 测试对象的选取

以 44 类中的 3 类简单形状类别(球体、立方体、圆柱体)和 3 类常见复杂类别(贯穿体零件、桌子、椅子)共 6 类作为示例,展示实验效果。6 种类别的标准三维模型如图 7 所示。

通过激光扫描获取 6 种类别的 10 万、70 万、

删除公共内边;然后利用圆准则或球面准则判断插入点的连接方式,完成一个顶点的插入;重复执行此步骤,直至完成所有点的插入,实现 Delaunay 三角化三维重建。

由以上得到本文的总体流程如图 6 所示,对待重建三维点云进行点云简化和去噪的预处理后,保存为待重建点云库,将其输入 DGCNN 完成三维点云的分类,将所得类别与三维模型库进行匹配,一一调用出待重建三维点云的三维模型,完成批量快速的三维重建,从而实现批量快速方法在三维重建中的应用。

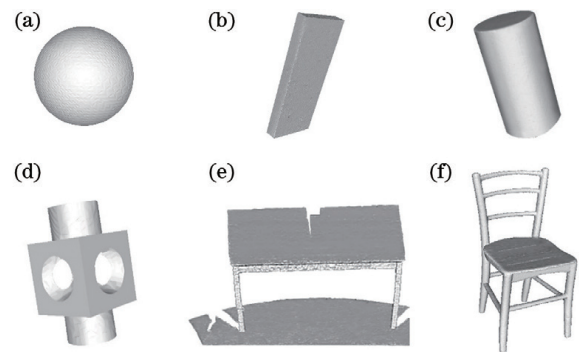


图 7 三维模型示例图。(a)球体;(b)立方体;(c)圆柱体;(d)贯穿体;(e)开源桌子点云集;(f)开源椅子点云集

Fig. 7 Example diagrams of 3D models. (a) Sphere; (b) cube; (c) cylinder; (d) penetrant; (e) open-source table point cloud set; (f) open-source chair point cloud set

120 万数量级的三维点云各 20 组,即共 360 组三维点云作为待重建的测试点云,以验证本文方法的有效性。

3.2 预处理效果的对比

3.2.1 去噪效果的对比

为验证本文算法的有效性,分别扫描标准球、标准量块、标准圆柱 3 个典型类别的标准件,得到初始点云数据,以 3 个标准件作为简单三维点云、开源桌子点云集作为复杂三维点云为例,分别利用 k 邻域去噪算法、半径滤波算法与本文滤波算法进行去噪

并比较。其中,阈值设置的过大或过小可能会导致去噪结果较差或部分主体点云被当作噪点剔除。因此经多次试验并比较,将 k 邻域去噪算法中参数值设置为 $k=50, a=1.0$; 在半径滤波算法中参数值设置为滤波半径 $r=0.02$, 滤波半径内包含点云数量的阈值为 $b=15$; 本文去噪算法的参数设置与 k 邻域去噪算法相同, 为 $k=50, a=1.0$ 。

去噪效果如图 8~11 所示, 去噪后的点云数量如表 1 所示。

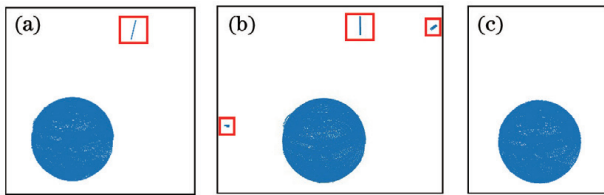


图 8 标准球去噪效果对比。(a) k 邻域去噪算法;
(b)半径滤波算法;(c)本文去噪算法

Fig. 8 Comparison of denoising effect of the standard ball. (a) k -neighbor denoising algorithm; (b) radius filtering algorithm; (c) proposed denoising algorithm

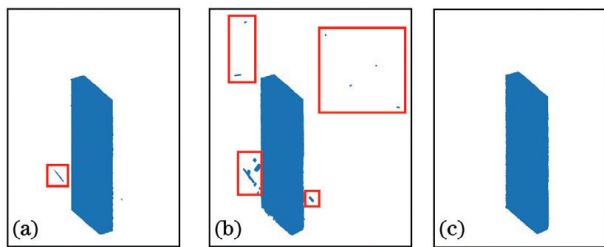


图 9 标准量块去噪效果对比。(a) k 邻域去噪算法;
(b)半径滤波算法;(c)本文去噪算法

Fig. 9 Comparison of denoising effect of the standard gauge block: (a) k -neighbor denoising algorithm; (b) radius filtering algorithm; (c) proposed denoising algorithm

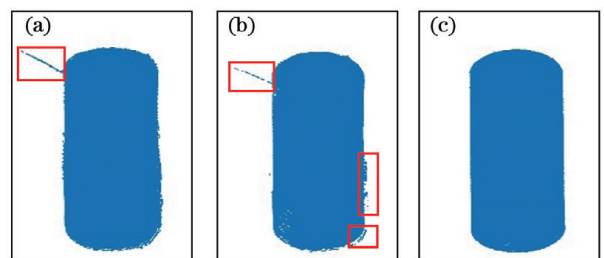


图 10 标准圆柱去噪效果对比。(a) k 邻域去噪算法;
(b)半径滤波算法;(c)本文去噪算法

Fig. 10 Comparison of denoising effect of the standard cylinder: (a) k -neighbor denoising algorithm; (b) radius filtering algorithm; (c) proposed denoising algorithm

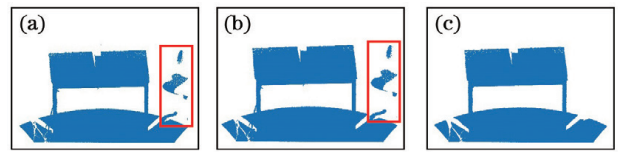


图 11 开源桌子点云去噪效果对比。(a) k 邻域去噪算法;(b)半径滤波算法;(c)本文去噪算法

Fig. 11 Comparison of denoising effect of the open-source table point cloud set: (a) k -neighbor denoising algorithm; (b) radius filtering algorithm; (c) proposed denoising algorithm

表 1 三种方法去噪后的点云数量

Table 1 Number of point clouds after denoising by the three methods

Point cloud set	k -neighbor denoising algorithm	Radius filtering algorithm	Proposed denoising algorithm
Standard ball	103642	108392	88349
Standard gauge block	671400	687593	659876
Standard cylinder	1125163	1149893	1116856
Open-source table point cloud set	451306	452099	429705

图 8 为三种不同的去噪算法对标准球点云数据进行去噪处理后的效果对比。可以看出,光反射等外因造成主体点云外的离群点十分密集,导致 k 邻域去噪算法和半径滤波算法无法根据点云密度剔除此部分噪声点。本文算法可根据 k 邻域的全局信息判断与剔除密集噪点,达到更好的去噪效果。

图 9 为三种不同的去噪算法对标准量块点云数据进行去噪处理后的效果对比。其中,经半径滤波算法去噪后的主体点云外有较多的密集噪点未被剔除,且量块底部边缘保留不完整,去噪效果最差;经 k 邻域去噪算法去噪后的点云周围也有少量密集噪点未被剔除;而本文算法可有效剔除密集噪点,总体去噪效果最好。

图 10 为利用三种去噪算法对标准圆柱去噪后的效果对比。由图可知,采用半径滤波算法进行滤波得到的主体点云周围存在较多的密集噪点,且圆柱边缘处也存在较多的混杂噪点,导致圆柱边缘不清晰,整体去噪效果最差;使用 k 邻域去噪算法得到的点云周围也存在少量密集噪点;本文算法有效地剔除了密集噪点与混杂噪点,且真实地保留了边缘处的点云数据,去噪效果明显优于上述两种去噪算法。

图 11 为三种不同的去噪算法对开源桌子点云

集进行去噪处理后的效果对比。经 k 邻域去噪算法和半径滤波算法的主体点云右侧存在明显的密集噪点,无法根据点云密度将其剔除;而本文算法可根据 k 邻域的全局信息有效剔除密集噪点。

由图 8~11 和表 1 可以看出, k 邻域去噪算法和半径滤波算法由于只考虑了 k 邻域内的点云信息,无法全部去除肉眼即可辨别的明显远离主体点云的密集噪点,去噪效果较差;而本文算法在 k 邻域去噪算法的基础上通过双阈值约束进行优化,引入了基于 k 邻域的全局信息,对远离主体点云的密集噪点也可通过任意点与其他点的平均距离来判断并删除,相比于传统的两种算法,对于密集噪点、漂

移噪点和混杂噪点均有较好的去噪效果,可降低对后续点云分类准确率的影响。

3.2.2 预处理顺序不同时预处理时间和分类效果的对比

为验证本文提出的点云预处理方法的可行性,以 10 万、70 万、120 万三种数量级的三维点云为例,将传统方法中先去噪再简化的预处理时间 t_a 与本文提出的先简化再去噪的预处理时间 t_b 进行对比,得到结果如表 2 所示,两种方式预处理后得到的点云数量如表 3 所示。其中,传统预处理方法中点云去噪算法以 k 邻域去噪算法为例,体素化网格算法的参数选取滤波时创建的体素体积 1.5。

表 2 点云预处理所需时间对比

Table 2 Comparison of the time required for point cloud preprocessing

Number of initial point clouds	Traditional preprocessing method			Proposed preprocessing method		
	Denoising time t_{a1}/s	Simplifying time t_{a2}/s	Total time t_a/s	Simplifying time t_{b1}/s	Denoising time t_{b2}/s	Total time t_b/s
108957	0.973	0.025	0.998	0.053	0.026	0.079
728907	1.880	0.067	1.947	0.094	0.134	0.228
1263084	9.460	0.238	9.698	0.306	0.589	0.895

表 3 预处理后点云数量对比

Table 3 Comparison of point cloud number after preprocessing

Number of initial point clouds	Traditional preprocessing method		Proposed preprocessing method	
	Number of point clouds after denoising	Number of point clouds after simplifying	Number of point clouds after simplifying	Number of point clouds after denoising
108957	103642	1063	1521	1346
728907	671400	4789	6113	5897
1263084	1125163	12040	20963	20738

由表 2 与表 3 中的数据可明显看出,点云简化步骤的处理时间在点云数量级为 10 万与 120 万之间的时间差距较小,整体步骤所需的时间极短,对预处理的时间影响较小;由于根据约束条件判断并剔除噪声点时需遍历全部三维坐标点,点云去噪步骤才是影响预处理时间长短的主要原因。且由表 2 可看出,点云数量的多少与点云去噪时的遍历时间呈正相关,点云数量越大,点云去噪所需的时间越长。因此,进行点云简化后再去噪,可明显缩短点云预处理时所需的遍历时间,提高了数据处理效率。

将 360 组待重建的三维点云按上述两种预处理方式处理后,多次输入 DGCNN 中进行点云分类,得到两种预处理方式下 5 次分类测试结果的准确率,如表 4 所示,其中 a_1 为先去噪再简化方式下的测试准

准确率, b_1 为先简化再去噪方式下的测试准确率。

表 4 准确率对比

Table 4 Comparison of accuracy

Test No.	Test accuracy a_1	Test accuracy b_1
1	0.904	0.907
2	0.901	0.905
3	0.913	0.921
4	0.909	0.904
5	0.917	0.905

由表 4 可知,两种预处理方式对点云分类的准确率没有影响,且使用本文所提出的预处理方法用时更短,可缩短准备工作部分所用时间,提高效率。

同时,实验过程中发现,本文的批量三维重建方

法依赖于前期准备工作所生成的三维模型库,适用于现有三维模型库已包含所有待重建三维点云类别的情况。若在批量重建时发现当前的模型库中没有对应的三维点云类别,可事先通过扫描、网上获取等方式获得未包含类别的大量点云数据,加入 DGCNN 的训练与评估点云库中,重新完成训练与评估这一准备工作;并事先重建出这些类别的三维

模型,加入三维模型库中,完成三维模型库的扩充操作,再进行后续的批量三维重建工作。

3.3 批量快速三维重建的验证

选择点云数量为 10 万左右上述 6 类三维点云 10 个、20 个、40 个、60 个、80 个、100 个、120 个进行批量三维重建实验。其处理流程所得结果如图 12 所示。

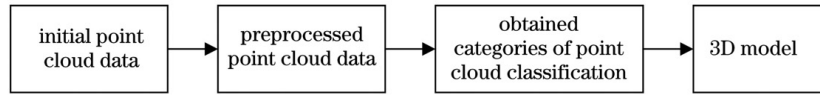


图 12 处理流程

Fig. 12 Processing flow

传统三维重建方法进行批量重建时所需时间为分别进行单个三维点云重建的累计时间 t , 本文方法进行批量三维重建所需时间为点云预处理时间与分类调用所需时间之和 t' , 两种方法的时间对比如表 5 所示。

表 5 点云数量为 10 万级时重建时间对比

Table 5 Comparison of reconstruction time when the number of point clouds is 100000

Number of 3D point clouds	Time required by the traditional method t/s	Time required by the proposed method t'/s
10	10.08	0.87
20	21.78	2.18
40	41.54	3.92
60	62.12	5.98
80	82.46	7.64
100	101.94	9.54
120	122.08	11.98

由表 5 数据可绘制出点云数量为 10 万级时三维点云个数与所需的重建时间的关系,如图 13 所示。

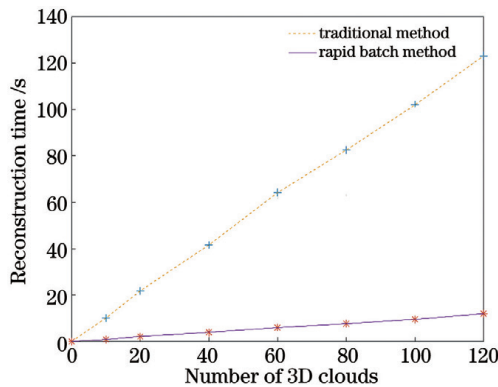


图 13 点云数量为 10 万级时重建个数与重建时间关系

Fig. 13 Relationship between reconstruction number and reconstruction time when the number of point clouds is 100000

选择点云数量为 70 万左右上述 6 类三维点云 10 个、20 个、40 个、60 个、80 个、100 个、120 个进行批量三维重建实验,记录传统方法与本文方法分别所需的时间如表 6 所示。

表 6 点云数量为 70 万级时重建时间对比

Table 6 Comparison of reconstruction time when the number of point clouds is 700000

Number of 3D point clouds	Time required by the traditional method t/s	Time required by the proposed method t'/s
10	20.33	2.36
20	42.08	4.78
40	82.74	9.52
60	123.56	14.34
80	164.14	19.26
100	204.06	23.78
120	245.28	29.06

由表 6 数据可绘制出点云数量为 70 万级时三维点云个数与所需的重建时间的关系,如图 14 所示。

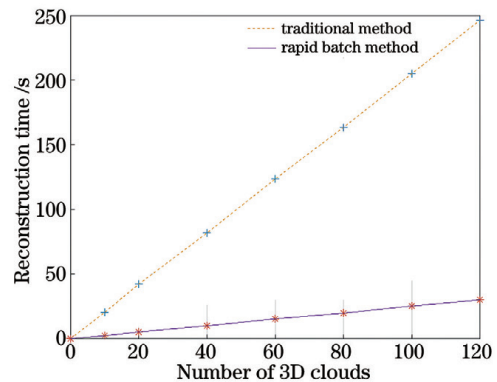


图 14 点云数量为 70 万级时重建个数与重建时间关系

Fig. 14 Relationship between reconstruction number and reconstruction time when the number of point clouds is 700000

选择点云数量为 120 万左右的上述 6 类三维点云 10 个、20 个、40 个、60 个、80 个、100 个、120 个进行批量三维重建实验,记录传统方法与本文方法分别所需的时间如表 7 所示。

表 7 点云数量为 120 万级时重建时间对比

Table 7 Comparison of reconstruction time when the number of point clouds is 1200000

Number of 3D point clouds	Time required by the traditional method t/s	Time required by the proposed method t'/s
10	97.97	9.03
20	194.59	17.94
40	393.72	37.06
60	588.54	56.49
80	780.97	71.96
100	980.56	92.44
120	1178.41	109.98

由表 7 数据可绘制出点云数量为 120 万级时三维点云个数与所需的重建时间的关系,如图 15 所示。

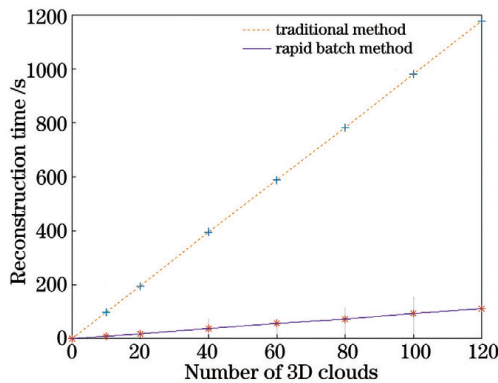


图 15 点云数量为 120 万级时重建个数与重建时间关系
Fig. 15 Relationship between reconstruction number and reconstruction time when the number of point clouds is 1200000

由表 5 可以看出,当点云数量为较少的 10 万级时,传统方法与本文方法重建 10 个三维点云所需的时间为 10.08 s 与 0.87 s,虽然该时间仍在可控范围内,但两者的重建时间相差很大,效率提升 10 倍,此时传统方法在批量重建实时性上的缺陷已经开始显现。随着三维点云个数的增多,两种方法在批量重建上的时间差也逐渐增大,重建 120 个三维点云所需的时间为 122.08 s 和 11.98 s,时间差距较为明显,此时传统方法已经失去实时性,而本文方法仍能保持在可接受范围内。同理,由表 6 和表 7 横向

比较可以同样看出:在点云数量级一定时,在三维点云个数相同的情况下,本文方法批量重建所用时间远小于传统方法所用时间;随着待重建三维点云个数的增多,两种重建方法下三维重建所需时间均增大,但由图 13~15 可知,本文方法的时间增长速度明显小于传统方法,且待重建的三维点云越多,本文方法较传统方法所节省的时间也就越多,因此本文方法在批量重建时效率明显提升,提升约 10 倍。

由表 5~7 纵向比较可以看出:在三维点云个数一定时,传统方法与本文方法均存在点云数量级越大,三维重建所用时间越长的特点;而且随着点云数量级的增大,传统方法三维重建所用时间的增长速度明显大于本文方法的增长速度。因此批量重建点云数量较大的三维点云时,本文方法更具优势。

4 结 论

在对数字化车间等室内场景、或大量建筑物等室外场景进行精度要求不高的基本模型的三维重建时,传统的三维重建方法无法完成批量自动化三维重建,且存在耗时较长、效率不足等问题,针对这一问题,本文提出了一种基于多标签分类的点云批量快速三维重建方法。该方法在进行批量三维重建时,对待重建的批量三维点云采取先简化再去噪的预处理方法,构建出待重建点云库,其中采用基于双阈值约束优化的 k 邻域去噪算法优化点云质量。将待重建点云库输入 DGCNN 中,得到分类结果,通过类别匹配一一调用事先生成的三维模型库中对应类别的三维重建结果,实现三维点云的批量快速三维重建。若三维模型库中缺少对应的类别,可获取相应类别的三维点云,加入 DGCNN 中重新训练;并将三维重建的结果加入三维模型库中,完成类别的扩充。为验证本文改进的去噪算法的有效性,利用标准球、标准圆柱、标准量块以及开源桌子点云集 4 种三维点云进行验证,通过点云可视化可以看出,本文算法较传统的 k 邻域去噪算法和半径滤波算法效果更好,可有效去除主体点云外的密集噪点。为验证本文提出的快速批量处理方法在三维模型重建中的有效性,利用加入了球体、立方体、圆柱体和贯穿体的增量 ModelNet40 模型进行验证。实验结果表明,本文方法重建 120 个 10 万数量级点云数据耗时 11.98 s,重建 120 个 70 万数量级点云数据耗时 29.06 s,重建 120 个 120 万数量级点云数据耗时 109.98 s,相较传统三维重建方法所需的 122.08 s、245.28 s 与 1178.41 s,效率提高了近 10 倍,在点云

数量大与三维点云个数多两方面,效率均有明显的提升,可实现高效的点云批量快速的三维重建。由于常见的室内工况环境往往包含固定类别与形状的物体,从本文所提方法的处理时长可见,该方法可以适用于普通室内工况环境的快速重建。

参 考 文 献

- [1] Shi G, Zheng L J, Wang W, et al. Non-scanning three-dimensional imaging system with a single-pixel detector: simulation and experimental study [J]. *Applied Sciences*, 2020, 10(9): 3100-3112.
- [2] Lhuillier M. Overview of shelling for 2-manifold surface reconstruction based on 3D delaunay triangulation [J]. *Journal of Mathematical Imaging and Vision*, 2017, 59(2): 318-340.
- [3] Escolano S O, Rodriguez J G, Morell V, et al. 3D surface reconstruction of noisy point clouds using growing neural gas: 3D object/scene reconstruction [J]. *Neural Processing Letters*, 2016, 43(2): 401-423.
- [4] Hazzat S, Akkad N, Merras M, et al. Fast 3D reconstruction and modeling method based on the good choice of image pairs for modified match propagation [J]. *Multimedia Tools and Applications*, 2020, 79(11/12): 7159-7173.
- [5] Huang L X. Research on point cloud filtering and plane pitting algorithm based on 3D laser scanning technology [D]. Xi'an: Chang'an University, 2017: 2-3.
黄凌潇. 基于三维激光扫描技术的点云滤波与平面拟合算法研究 [D]. 西安: 长安大学, 2017: 2-3.
- [6] Cheng X J, Jia D F, Cheng X L, et al. The theory and technology of mass-point cloud data processing [M]. Shanghai: Tongji University Press, 2014: 3-20.
程效军, 贾东峰, 程小龙, 等. 海量点云数据处理理论与技术 [M]. 上海: 同济大学出版社, 2014: 3-20.
- [7] Lu D D, Zou J G. Comparative research on denoising algorithms of 3D laser point cloud [J]. *Bulletin of Surveying and Mapping*, 2019(S2): 102-105.
鲁冬冬, 邹进贵. 三维激光点云的降噪算法对比研究 [J]. *测绘通报*, 2019(S2): 102-105.
- [8] He Y, Liang B, Yang J, et al. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features [J]. *Sensors (Basel, Switzerland)*, 2017, 17(8): E1862.
- [9] Wang C. Design and construction of handheld wireless real-time 3D reconstruction system based on RGB-D camera [D]. Harbin: Harbin Institute of Technology, 2019: 7-42.
- 王澈. 基于 RGB-D 的手持无线实时三维重建系统的设计与构建 [D]. 哈尔滨: 哈尔滨工业大学, 2019: 7-42.
- [10] Yang K S. Application and real-time image semantic segmentation based on deep learning in 3D reconstruction system [D]. Chengdu: University of Electronic Science and Technology of China, 2019: 10-44.
杨坤山. 基于深度学习的图像语义分割在三维重建系统中应用与实时化 [D]. 成都: 电子科技大学, 2019: 10-44.
- [11] Ni H, Zhang J X, Lin X G, et al. Edge detection and straight line segment extraction from 3D-point clouds: review and prospect [J]. *Bulletin of Surveying and Mapping*, 2016(7): 1-4.
倪欢, 张继贤, 林祥国, 等. 三维点云边缘检测和直线段提取进展与展望 [J]. *测绘通报*, 2016(7): 1-4.
- [12] Zhou Y, Shen S H, Hu Z Y, et al. Detail preserved surface reconstruction from point cloud [J]. *Sensors*, 2019, 19(6): E1278.
- [13] Chen H Z, Gao J. Point cloud consolidation algorithm for reflective workpieces with point cloud defects [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2019, 31(7): 1258-1266.
陈华臻, 高健. 面向反光工件点云缺陷的点云增强算法 [J]. *计算机辅助设计与图形学学报*, 2019, 31(7): 1258-1266.
- [14] Zhang L, Sun J, Zheng Q, et al. 3D point cloud recognition based on a multi-view convolutional neural network [J]. *Sensors*, 2018, 18(11): E3681.
- [15] Pang Z Y, Zhou Z F, Wang L D, et al. Improved three-dimensional reconstruction algorithm for point cloud data [J]. *Laser & Optoelectronics Progress*, 2020, 57(2): 021102.
庞正雅, 周志峰, 王立端, 等. 改进的点云数据三维重建算法 [J]. *激光与光电子学进展*, 2020, 57(2): 021102.
- [16] Yu T, Yang J. Point cloud model recognition and classification based on K-nearest neighbor convolutional neural network [J]. *Laser & Optoelectronics Progress*, 2020, 57(10): 101510.
于挺, 杨军. 基于 K 近邻卷积神经网络的点云模型识别与分类 [J]. *激光与光电子学进展*, 2020, 57(10): 101510.
- [17] Wang X J, Ma J, Wang N N, et al. Deep learning model for point cloud classification based on graph convolutional network [J]. *Laser & Optoelectronics Progress*, 2019, 56(21): 211004.
王旭娇, 马杰, 王楠楠, 等. 基于图卷积网络的深度学习点云分类模型 [J]. *激光与光电子学进展*, 2019, 56(21): 211004.

- [18] Zhang A W, Liu L L, Zhang X Z, et al. Multi-feature 3D road point cloud semantic segmentation method based on convolutional neural network [J]. Chinese Journal of Lasers, 2020, 47(4): 0410001. 张爱武, 刘路路, 张希珍, 等. 道路三维点云多特征卷积神经网络语义分割方法 [J]. 中国激光, 2020, 47(4): 0410001.
- [19] Charles R Q, Hao S, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 77-85.
- [20] Qi C R, Yi L, Su H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space [C] // 31st Annual Conference on Neural Information Processing Systems (NIPS), December 4, 2017, Red Hook, NY, United States. New York: ACM, Inc., 2017: 5105-5114.
- [21] Wang Y, Sun Y B, Liu Z W, et al. Dynamic graph CNN for learning on point clouds [J]. ACM Transactions on Graphics, 2019, 38(5): 1-12.
- [22] Cabo C, Ordoñez C, García-Cortés S, et al. An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2014, 87: 47-56.
- [23] Fu Y, Zhai J L. Research on scattered points cloud denoising algorithm [C] // 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), September 19-22, 2015, Ningbo, China. New York: IEEE Press, 2015: 1-5.
- [24] Guo Z X, Xie X Y, Liu S, et al. Sparse outlier removal algorithm based on resuming features [J]. Journal of Guizhou Normal University (Natural Sciences), 2016, 34(1): 88-92. 郭子选, 谢晓尧, 刘嵩, 等. 基于特征恢复的离群点移除算法的研究 [J]. 贵州师范大学学报(自然科学版), 2016, 34(1): 88-92.
- [25] Atzmon M, Maron H, Lipman Y, et al. Point convolutional neural networks by extension operators [J]. ACM Transactions on Graphics, 2018, 37(4): 1-12.
- [26] Ray N, Sokolov D, Lefebvre S, et al. Meshless voronoi on the GPU [J]. ACM Transactions on Graphics, 2018, 37(6): 1-19.