

6-5-2020

Denoising Algorithm Based on Neighborhood Expansion Clustering

Xinggong Li

School of Information, Yunnan Normal University, Kunming 650500, China;

Yaping Zhang

School of Information, Yunnan Normal University, Kunming 650500, China;

Yuwei Yang

School of Information, Yunnan Normal University, Kunming 650500, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Denoising Algorithm Based on Neighborhood Expansion Clustering

Abstract

Abstract: *Aimed at the three dimensional point clouds model with outliers and noises which is reconstructed based on 2D images, a new denoising algorithm based on neighborhood expansion clustering is proposed. The search for other neighboring points of each data point is conducted by using the Euclidean distance between data points and the transitive property of the neighborhood location relation. All points are processed for cluster partition, which detects and filters the outliers in the point clouds model. The concept of neighborhood expansion clustering and the fast search algorithm based on dynamic grids division are discussed. It solves the problem of detecting and filtering the outliers distributed in isolation or densely around the point clouds model, which improves the efficiency of traditional k-nearest neighbor algorithm to denoise the point clouds data. Simulation results show that the proposed algorithm can effectively filter the outliers distributed in isolation or densely around the point clouds model.*

Keywords

neighborhood expansion, outliers, denoising, k-nearest neighbor, Euclidean distance

Recommended Citation

Li Xinggang, Zhang Yaping, Yang Yuwei. Denoising Algorithm Based on Neighborhood Expansion Clustering[J]. Journal of System Simulation, 2017, 29(11): 2663-2670.

一种基于邻域扩展聚类的去噪算法

李幸刚, 张亚萍, 杨雨薇

(云南师范大学信息学院, 云南 昆明 650500)

摘要: 针对基于二维图像重建出的带有离群点和噪声的三维点云模型, 提出了一种基于邻域扩展聚类的去噪算法。通过数据点之间的欧氏距离以及相邻位置关系的可传递性, 搜索每个数据点的邻域, 然后对所有点进行聚类划分, 从而检测和滤除点云模型中的离群点。重点讨论了点云邻域扩展聚类的概念和方法、如何利用基于动态网格划分法快速搜索点的邻域。解决了点云模型周围孤立及密集分布的离群点检测和滤除问题, 提高了传统k-近邻等算法对于点云数据的去噪效率。仿真实验结果表明, 该算法可有效滤除点云模型中孤立和密集分布的离群点。

关键词: 邻域扩展; 离群点; 去噪; k-近邻; 欧氏距离

中图分类号: TP391.72

文献标识码: A

文章编号: 1004-731X (2017) 11-2663-08

DOI: 10.16182/j.issn1004731x.joss.201711010

Denoising Algorithm Based on Neighborhood Expansion Clustering

Li Xinggang, Zhang Yaping, Yang Yuwei

(School of Information, Yunnan Normal University, Kunming 650500, China)

Abstract: Aimed at the three dimensional point clouds model with outliers and noises which is reconstructed based on 2D images, a new denoising algorithm based on neighborhood expansion clustering is proposed. The search for other neighboring points of each data point is conducted by using the Euclidean distance between data points and the transitive property of the neighborhood location relation. All points are processed for cluster partition, which detects and filters the outliers in the point clouds model. The concept of neighborhood expansion clustering and the fast search algorithm based on dynamic grids division are discussed. It solves the problem of detecting and filtering the outliers distributed in isolation or densely around the point clouds model, which improves the efficiency of traditional k-nearest neighbor algorithm to denoise the point clouds data. Simulation results show that the proposed algorithm can effectively filter the outliers distributed in isolation or densely around the point clouds model.

Keywords: neighborhood expansion; outliers; denoising; k-nearest neighbor; Euclidean distance

引言

基于二维图像的建模方法是当前计算机视觉

以及计算机图形学界极为活跃的研究领域。由于图像包含了丰富的几何场景信息, 所以可以根据一组无序图像来恢复三维几何结构, 从而快速生成三维模型。拍摄时, 由于模型所处的复杂周围环境以及光线的强弱影响等因素, 使用这些图像重建出的三维点云模型会包含许多噪点和离群点。如果直接进行表面重建, 会导致最后的模型表面粗糙甚至变



收稿日期: 2016-04-26 修回日期: 2016-09-02;
基金项目: 国家自然科学基金(61262070, 61462097);
作者简介: 李幸刚(1992-), 男, 河南平顶山, 硕士生, 研究方向为计算机图形学; 张亚萍(1979-), 女, 云南凤庆, 博士, 副教授, 研究方向为计算机图形学, 并行计算。

<http://www.china-simulation.com>

• 2663 •

形。因此,为了获得更精确的三维模型,需要在表面重建之前对点云数据做去噪处理。常见的去噪算法有很多种,对于有序或部分有序的点云,可以用最小二乘滤波^[1]、维纳滤波^[2]、平滑滤波^[3]等。对于散乱点云,可以用平均曲率流^[4]、移动最小二乘法^[5]等方法。根据离群点的分布情况,离群点检测方法可分为基于密度的检测方法、基于聚类的检测方法、基于距离的检测方法等^[6]。根据一组图像重建出的三维点云模型,数据点主要密集分布在模型表面,而离群点则是孤立存在或是密集分布在模型周围。当前许多算法对于点云中孤立存在的离群点具有较好的滤除效果,例如基于密度的离群噪声点检测算法^[7]、基于欧氏距离的检测算法^[8]、基于 k-近邻树的离群检测算法^[9]等。但是对于密集分布在一起且数量较大的离群点集合,以上算法则不能达到较好的滤除效果。本文针对离群点的这一分布特点,提出一种邻域扩展聚类的去噪算法,通过动态网格划分法构建点云拓扑关系来快速所有点进行聚类划分处理,从而有效检测和滤除离群点。

1 基于图像的三维重建

基于图像的三维重建主要是通过多幅二维图像中物体的颜色、位置、形状等信息,来恢复它在空间中的三维信息。其基本步骤为:

(1) 图像的采集。针对想要重建的物体,在不同水平及垂直高度上进行拍摄,以便获得整个物体的真实三维信息。

(2) 特征点检测与匹配。首先,使用 SIFT (Scale-invariant feature transform)^{[10][11]}算法分别找出每张图像中的特征点,然后根据这些特征点将所有图像进行两两匹配。

(3) 相机标定与场景恢复。利用摄影定理可计算照相机与图像的空间位置关系等三维结构场景信息,并进行投影重建。这一步称为从运动中恢复结构(SFM, Structure from Motion)^[12-14]。

(4) 三维点云重建。根据由 Y. Furukawa 提出的 PMVS(patch-based multi-view stereo)算法^[15]过

程,首先检测每幅图像中的特征点,然后匹配这些特征点并根据三角测量原理恢复这些点的深度信息,从而得到物体表面特征点的稀疏三维点云,然后用面片表示这些空间点,称之为种子面片。最后从种子面片开始进行扩散,直到得到的稠密面片能够将物体或场景的表面全部覆盖为止。

根据上述第(4)步 PMVS 算法得到的三维重建模型,所有数据点都是通过矩形面片的形式来表示。数据点为面片的中心,数据点的方向即为面片的法向,数据点所带的颜色值即为该面片的颜色。为后续描述方便,本文出现的离群点的滤除方法是将离群点以及离群点所在面片同时从点云模型中删除。

2 欧氏距离

欧氏距离(Euclidean distance)又称欧几里得距离,是一个通常采用的距离定义,它是在 m 维空间中两点之间的真实距离。假设三维空间中的两点 $A(x_1, y_1, z_1)$ 、 $B(x_2, y_2, z_2)$, 那么 A 与 B 之间的欧氏距离计算公式为:

$$D(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (1)$$

本文中所述两数据点的距离均指这两点之间的三维欧氏距离。

3 邻域扩展聚类

根据二维图像重建得到的三维点云模型,数据点基本都分布在模型表面,而且相邻数据点之间距离很小,这些密集分布的点能够较好的反应出物体真实表面特性。如果点云中两个点 p, p' 之间的空间距离满足下面不等式,则认为这两个点是相邻的。

$$D(p, p') = \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2} < D \quad (2)$$

假设相邻的点之间存在一条路,若点云中每个数据点与其他点之间都存在可到达的路,那么我们可以认为该点云是一个整体连续的数据点集合。根据点云的位置相邻连续性,我们可以找到点云模型中最大的聚类集合,并认为该集合即为重建点云模型表面数据点组成的集合。其他小的聚类集合则认

为是离群点被滤除。利用该方法能够很好的检测并滤除重建过程中出现的离群点、噪点, 从而提高重建模型的精度。

判断两个数据点是否满足位置相邻的阈值 D 的大小, 对整个算法的去噪效果起着决定性作用。如果 D 值设置过大, 则不能有效识别距离点云模型表面较近的离群点。如果 D 值设置过小, 则会滤除一些应当保留的数据点, 造成点云模型不够完整。为保证整个三维重建过程的自动化和完整性, 我们采用抽样平均法自动生成 D 值。首先从点云数据集中随机抽取 α 个数据点作为样点, 找到每个样点的最近邻数据点, 并记录与最近邻点之间的距离。然后将这 α 个距离值按照从小到大的顺序进行排列, 将排序后结果记为 $d_i (i=1, 2, \dots, \alpha)$, 为排除个别较大或较小距离的影响, 删除这些距离值中最大以及最小的 β 个。最后, 将剩下的 $\alpha-2\beta$ 个距离值做平均运算, 即可得到抽样平均距离 \bar{d} , 如公式(3)所示:

$$\begin{cases} \bar{d} = \sum_{i=\beta}^{\alpha-\beta} d_i / (\alpha - 2\beta) \\ \alpha = N / k_1 \\ \beta = \alpha / k_2 \end{cases} \quad (3)$$

其中: N 为点云中数据点总个数, k_1 、 k_2 为比例因子。根据已经得到的抽样平均距离, 令 $D = \delta \bar{d}$ (δ 为比例因子), 将 D 作为判断两数据点是否相邻的参考阈值。

4 邻域搜索

在三维点云中, 搜索与某个点距离较近的其他点常用的方法有: 划分空间单元格法^[16]、八叉树法^[17]、k-d 树法^[18]和遍历该点与所有数据点的距离的方法。前两种方法都是将点云所在包围盒进行划分, k-d 树法通常用于查找距离最近的两点, 而遍历所有点则是时间复杂度最高的方法。基于图像的三维重建, 为了获得高精度、高逼真的点云模型, 重建出的点云模型往往包含数十万乃至上百万个数据点。如果使用上述几种方法来搜索所有数据点的邻域, 会消耗大量的计算时间。除此之外, 由于数据点分布没有特定的状态, 特别是离群点的分布

较为散乱、不规律。所以为了实现相邻点的快速搜索, 需要先建立数据点之间的空间拓扑关系, 然后再搜索每个点的相邻点, 进而构造最大聚类集合。

本文采用文献[19]中提出的基于动态网格划分的散乱点 k-近邻快速搜索算法。其算法思想为: 假设点云模型中共包含 n 个数据点, 可将这些数据点表示为 $P_i(x_i, y_i, z_i)$, $i=1, 2, \dots, n$, 根据它们的 x , y , z 坐标值大小分别进行排序。首先比较所有数据点在 x 轴方向上的大小, x 值小的数据点排在前面, x 值较大的排在后面。如果两点在 x 方向上坐标值大小相同, 则比较它们的 y 方向坐标值, 如果这两点在 y 方向的坐标值也相同, 则继续比较它们的 z 方向坐标值, 同样按坐标值从小到大排序。依据此方法, 分别得到所有数据点在 x , y , z 三个方向上的坐标值排序。如果任意一点 p 的三维排序坐标为 (m, s, t) , 则表示该点在 x 轴方向排序后的序列号为 m , 在 y 轴方向排序后的序列号为 s , 在 z 轴方向排序后的序列号为 t 。以 p 点为中心, 在 x 轴的正反两个方向各搜索 l 个点, 从而得到在 x 轴方向上距离 p 点最近的 $2l$ 个点。然后采用同样方法分别得到在 y 轴和 z 轴方向上距离 p 点最近的 $2l$ 个点。如果 p 点在三维排序的某个坐标轴上, 位于它之前或之后的数据点少于 l 个, 则只取这少于 l 个点作为该方向相邻点搜索结果。最后, 求出 p 点在 x , y , z 三个方向上搜索到的所有数据点的交集, 设该交集为 Q 。以二维排序为例, 在图 1(a)中, 灰色区域表示 p 点在 x 轴和 y 轴两个方向上搜索得到的所有数据点。图 1(b)中灰色区域表示 p 点在两个方向上搜索得到的所有数据点的交集 Q 。

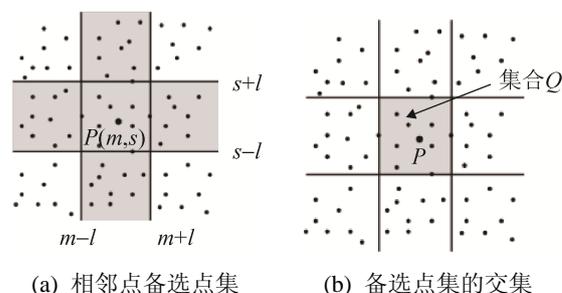


图 1 邻域搜索
Fig.1 Domain search

将点 p 跟点集 Q 中所有数据点做距离计算, 如果两者之间的距离满足 $D(p,p') < D$, 则认为点 p' 是 p 的相邻点。根据此方法可快速搜索点云中每个数据点的邻域。

5 基于邻域扩展聚类的去噪算法

5.1 算法描述

从点云中任意一点 p 出发, 找到该点邻域内所有点, 构成点集 F 。然后依次找出 F 中每个点的邻域, 如果这些相邻点不属于集合 F 则将其存入 F 中。重复该过程, 直到没有新的相邻点存入为止。判断此时 F 中数据点数量是否足够多, 如果数量不够, 则认为这些相邻的点只是点云模型周围的离群点集合, 将 F 中所有点从点云中剔除, 然后从另一点出发重新搜索数据点邻域聚类集合。如果数据点数量足够多, 则认为集合 F 即为所求最大聚类集合, 该集合以外的其他点被视为离群点加以滤除, 从而完成点云模型去噪工作。算法具体过程如下:

(1) 根据重建出的三维点云, 利用基于动态网格划分的邻域快速搜索算法构建数据点之间拓扑关系, 得到所有数据点在 x, y, z 三个方向上的坐标值排序。

(2) 采用抽样平均法得到抽样平均距离 \bar{d} , 令 $D = \delta \bar{d}$ (δ 为比例因子), 将 D 作为判断数据点是否相邻的参考阈值。

(3) 从点云中任取一点 p , 根据其三维排序坐标, 分别在 x, y, z 三个方向上, 以 p 点为中心沿正反两个方向各搜索 l 个点, 得到点 p 的邻域备选点集, 并求得它们的交集 Q 。

此时, 集合 Q 中的元素应满足:

$$\forall Q(m_i, s_i, t_i) = \begin{cases} m-l \leq m_i \leq m+l \\ s-l \leq s_i \leq s+l \\ t-l \leq t_i \leq t+l \end{cases} \quad (4)$$

其中: m, s, t 为 p 点三维排序坐标, m_i, s_i, t_i 分别为集合 Q 中第 i 个元素的三维排序坐标。

(4) 将集合 Q 中的数据点分别与点 p 做距离计

算, 如果两者之间的距离满足 $D(p,p') < D$, 则认为点 p, p' 是相邻的。将数据点 p 以及 Q 中所有与 p 满足相邻关系的点 p' 添加到新的集合 F 中。

(5) 从集合 F 中除 p 点之外的其它点出发, 重复(3)到(4)过程, 将所有相邻的数据点添加到 F 中, 如果某个点在 F 中已经存在, 则不再存入该点。

(6) 重复上述(5)过程, 直到 F 中没有新的点存入为止。

(7) 假设此时集合 F 中数据点个数为 M , 原始点云包含的数据点个数为 N , 如果 $M \geq \lambda * N$ (λ 为比例因子, 本文 λ 取 0.5), 则认为集合 F 中的所有点即为所求去噪后的点云模型数据。如果 $M < \lambda * N$, 则认为 F 中的所有点均为离群点, 于是从原始点云中不属于集合 F 的其它数据点开始重新执行(2)至(6)过程。直至找到满足要求的集合 F 或遍历完所有数据点为止。

5.2 算法实例与分析

将基于邻域扩展聚类的去噪算法在主频 3.4GHz、内存为 4.0GB 的 Windows7、64 位操作系统戴尔 PC 机上, 用 VS2010 进行了仿真实验。在本次实验中, 原始图像格式为 1067×800 像素, 然后使用 VisualSFM 建模软件对每组图像进行建模。第一组为玩具 ET 的 8 张各角度图像, 如图 2 所示; 第二组为石质花盆的 20 张各角度图像; 第三组为为佛陀雕塑的 30 张各角度图像。在加入去噪算法前, 根据三组图像重建出的三维点云模型, 在正视、侧视及俯视角度的显示效果如图 3 所示。



图 2 不同角度 ET 图像

Fig.2 ET images with different angles

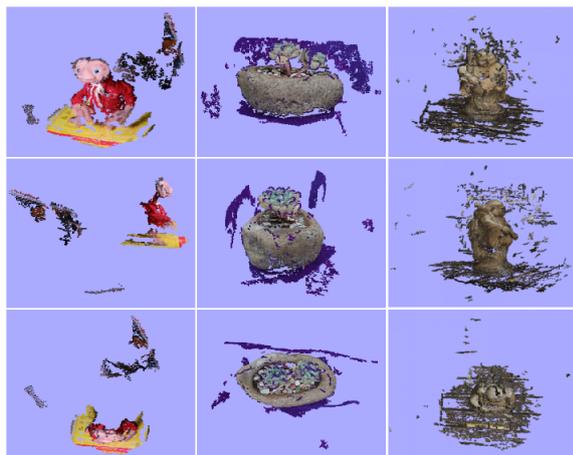


图 3 多角度原始点云模型效果
Fig.3 Effect of multi-angle original point cloud model

从上图 3 可以看出, 根据图像重建出的点云模型由于背景的干扰、光线强弱等因素, 点云中包含了許多孤立存在或密集分布在一起的离群点。如 ET 点云模型中离群点主要分布在模型后面; 花盆模型中离群点分布在模型表面轮廓附近; 佛陀雕塑模型中离群点则是围绕在模型周围。针对以上三组点云, 我们加入了基于邻域扩展聚类的去噪算法。仿真实验 1 中, 在保证每组点云原始数据点个数不变的前提下, 测得本文算法在 δ 取值不同时, 即判断相邻的阈值 D 的大小取值对去噪效果的影响, 各实验数据如表 1 所示。其中 Num 表示模型号, N 表示原始点云中数据点个数, M 表示去噪后数据点个数, R 表示去噪率, Y/N 表示点云模型是否完整。仿真实验 2 中, 在保证 δ 值不变的前提下, 对四种算法的去噪时间进行对比, 单位为秒(s)。其中 a 为传统 k-近邻算法^[20], 该算法基本思想是: 点 p 与点云中距离它第 k 近的点之间的距离, 记为 $D^k(p)$, 依照此方法求得所有数据点的第 k 近距离。然后将所有点中 k-近邻距离值较大的点视为离群点加以滤除, 在本实验中 k 值取 50。b 为不进行坐标值排序而是直接搜索数据点邻域进行扩展聚类的算法。c 为文献[13]提出的基于动态网格划分的散乱点 k-近邻快速搜索算法, k 值同样取 50。d 为本文算法。由于算法 a、b 与搜索长度 l 值无关, 因此单独列出。得到的各实验数据如表 2 所示。

表 1 不同 δ 值下点云去噪效果
Tab.1 Denoising effect of point cloud under different δ values

Num	δ	D	N	M	R	Y/N
1	10	0.5912	8847	0	—	否
1	15	0.7883	8847	4491	—	否
1	20	0.9854	8847	7583	14.3	是
1	25	1.1825	8847	7734	12.6	是
1	30	1.5767	8847	7823	11.6	是
2	20	0.3167	78541	37491	—	否
2	30	0.4653	78541	53642	—	否
2	50	0.7755	78541	68933	12.2	是
2	100	1.5510	78541	70617	10.0	是
2	150	2.3266	78541	71465	9.01	是
3	5	0.1562	163409	0	—	否
3	10	0.3124	163409	108645	—	否
3	20	0.6249	163409	142531	12.8	是
3	30	0.9373	163409	154677	5.34	是
3	50	1.5623	163409	158954	2.73	是

表 2 不同算法去噪时间对比
Tab.2 Comparison of denoising time between different algorithms

Num	δ	a	b	l	c	d
1	20	54.43	45.22	100	72.93	69.21
1	20	47.64	37.07	200	67.19	52.80
1	20	48.71	46.06	300	55.46	63.02
1	20	52.14	50.45	400	62.61	73.99
1	20	50.65	41.43	500	94.01	78.82
2	50	676.4	569.3	100	960.0	860.8
2	50	657.8	505.3	300	732.2	762.5
2	50	675.4	515.8	500	623.5	720.6
2	50	656.1	531.4	800	689.1	659.7
2	50	692.6	528.4	1000	771.5	686.4
3	15	5779	3708	300	4927	4548
3	15	5687	3271	500	4034	4478
3	15	5633	3538	1000	3501	3431
3	15	5617	3626	1500	4065	2814
3	15	5726	3936	2000	4372	3634

实验 1: 本文算法在不同 δ 值下的去噪效果。由表 1 中实验数据可以看出, 当 δ 取值过小时, 会影响重建模型的完整性, 造成错误的滤除效果。如果 δ 取值过大, 去噪率则会降低, 不能达到理想的去噪效果。由于第一组图像较少, 且模型所处背景比较复杂, 所以造成重建点云中包含较多的离群

点, 数据点不能完全覆盖模型表面, 部分区域数据点之间距离较大, 所以 δ 取值较大。第二组模型中花瓣与花瓣之间数据点数量较少、分布比较稀疏, 如果 δ 值不够大, 则使重建模型不够完整。第三组图像由于模型与其背景颜色相似, 所以重建模型周围环绕着大量块状离群点集合, 但是由于用于重建的图像数量足够多, 所以数据点聚类集合之间距离较小。当采用本文算法对三组点云去噪, δ 值分别取 20、50、15 时, 去噪后效果如图 4 所示。可以看到加入基于邻域扩展聚类的去噪算法后, 能够有效检测和滤除点云模型周围的离群点。



图 4 本文算法去噪后点云模型
Fig.4 Point cloud model after denoising in this paper

实验 2: 不同算法去噪时间对比。根据表 2 中实验数据可以看出:

(1) 当数据点数量较多时, 算法 a 去噪时间最长。这是由于每个数据点不仅要和其他所有点做距离运算, 还要计算这些距离中的第 k 小距离, 所以时间复杂度最高。

(2) 算法 b 同样需要将每一个数据点同其他所有点进行距离运算, 随着数据点数量的增加, 去噪时间成几何倍数增长。但是算法 b 不需要将计算每个点的第 k 近距离, 所有比算法 a 去噪时间稍短。

(3) 算法 c 是针对算法 a 的改进, 当数据点较多时, 可有效加快 k-近邻搜索速度。但是如果 l 取值过小, 则不能一次性找到足够多的 k-近邻备选点, 点集 Q 中的元素个数小于 k , 此时需要增大 l 然后重新进行搜索。而如果 l 取值过大, 备选点集中元素过多, 同样会降低搜索速度。

(4) 算法 d 是针对算法 b 的改进, 即对点云数据进行预先处理, 当数据点较多时, 选取适当的 l 值能够拥有比 b 更快的搜索速度。由于该算法既不

需要计算每个点的第 k 近距离, 同时也不存在因为找不到足够多备选点而要增大 l 值重新进行搜索的问题, 所以, 算法 d 去噪时间比算法 a、c 都要短。如果 b、d 算法最开始搜索的点就属于最大聚类集合上的点, 此时这两种算法去噪时间最短。

(5) 由表 1 中数据可看出, 当采用本文算法, 对三个模型分别滤除 1 200、1 万、2 万个数据点后, 点云模型仍然能够保持较好的完整性。如果采用 a 算法对这些模型进行去噪, 当设定离群点个数同样为 1 200、1 万、2 万时, 去噪后效果如图 5 所示。由于算法 a、c 都属于 k-近邻算法, 而算法 b、d 同为基于邻域扩展聚类的算法, 所以不再给出 b、c 算法的去噪效果图。从图 5 可以看出, k-近邻算法对于稀疏分布的噪点具有较好的检测效果, 但是对于大量密集分布在一起的噪点不能起到很好的检测效果。所以, 对于由图像重建出的三维点云模型, 基于邻域扩展聚类的算法不仅比 k-近邻算法具有更快的去噪速度, 而且具有更好的去噪效果。

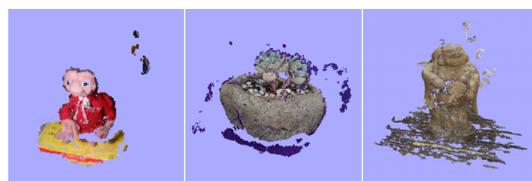


图 5 k-近邻算法去噪后点云模型
Fig.5 Point cloud model after denoising by k- nearest neighbor algorithm

6 结论

本文针对由图像重建出的三维点云模型, 结合基于动态网格划分的散乱点 k-近邻快速搜索算法, 加入点云邻域扩展聚类的概念, 通过构造数据点的最大聚类集合, 从而快速检测和滤除点云模型中的离群点。实验结果证明, 该算法能够有效地检测出点云模型周围孤立或密集分布的离群点, 大大提高了重建模型的精度。基于邻域扩展聚类的去噪方法局限在于不能有效检测出点云模型轮廓上颜色不一致的噪点, 如图 5 花盆模型轮廓上的黑色噪点。除此之外, 如果点云模型由多个部分组成, 那么当

各部分连接处数据点较少且分布稀疏时, 数据点个数较少的部分会被当做离群点滤除, 例如当 δ 取值 25 时, 会错误地滤除花盆模型中某些花瓣部分, 造成重建点云模型的不完整。针对这一类的模型, 可以通过保留所有聚类集合中数据点个数大于 $\lambda * N$ (λ 取 0.1、0.2..., N 为点云中数据点总个数) 的集合, 然后将数据点较少的其他集合视为离群点从而避免类似错误。除此之外, 本文算法虽然比 k -近邻算法时间复杂度更低, 但去噪时间依然很长。所以针对位于点云模型轮廓上颜色不一致的噪点的检测方法、对结构较为复杂的模型的去噪方法以及如何提高去噪的效率, 将是今后学习和研究的重点。

参考文献:

- [1] Jaehoon J, Sungchul H, Sanghyun Y, et al. Automated 3D Wireframe Modeling of Indoor Structures from Point Clouds Using Constrained Least-Squares Adjustment for As-Built BIM[J/OL]. December, 23, 2015. Journal of Computing in Civil Engineering(S1943-5487), doi: 10.1061/(ASCE)CP.1943-5487.0000556.
- [2] Diwakar M, Kumar M. CT Image Noise Reduction Based on Adaptive Wiener Filtering with Wavelet Packet Thresholding[C]// International Conference on Parallel, Distributed and Grid Computing. IEEE, Jaypee university of Information Technology, Solan, India, 2014.
- [3] Mazhari M, Hasanzadeh RPR. Suppression of Noise in SEM Images Using Weighted Local Hysteresis Smoothing Filter[J/OL]. FEB, 18, 2016. Scanning (S1932-8745). DOI: 10.1002/sca.21311.
- [4] Colding T, Minicozzi W I, Pedersen E. Mean Curvature Flow[J]. Bulletin of the American Mathematical Society(S0273-0979), 2015, 52(9): 4551-4587.
- [5] Rodriguez-Cuenca B, Garcia-Cortes S, A study of the roughness and curvature in 3D point clouds to extract vertical and horizontal surfaces[C]// 2015 IEEE International Geoscience and Remote Sensing Symposium (S2153-6996). IEEE. Milan, 2015: 4602-4605.
- [6] Angiulli F, Basta S, Lodi S, et al. GPU Strategies for Distance-based Outlier Detection[J]. IEEE Transactions on Parallel and Distributed Systems (S1045-9219). 2016, (99): 1.
- [7] Plessis MCD, Shiino H, Sugiyama M. Online direct density-ratio estimation applied to inlier-based outlier detection[J]. Neural Computation(S0899-7667), 2015, 27(9): 1899-1914.
- [8] Ramaswamy S, Rastogi R, Shim K. Efficient Algorithms for Mining Outliers from Large Data Sets[J]. Acm Sigmod Record(S0163-5808), 2000, 29(2): 427-438.
- [9] Petrovskiy M, Convolution Kernels for Outliers Detection in Relational Data[J]. Intelligent Data Engineering and Automated Learning(S0302-9743). 2003, 2690: 661-668.
- [10] Liu L, Peng F Y, Zhao K, et al. Simplified SIFT Algorithm for Fast Image Matching[J]. Infrared & Laser Engineering(S1007-2276), 2008, 37(1): 181-184.
- [11] Pysillos A P, Kayafas E. Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme[J]. IEEE Transactions on Intelligent Transportation Systems (S1524-9050), 2010, 11(2): 322-328.
- [12] Crandall D J, Owens A, Snavely N, et al. SfM with MRFs: Discrete-Continuous Optimization for Large-Scale Structure from Motion[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence (S0162-8828), 2013, 35(12): 2841-2853.
- [13] Ullman S. The Interpretation of Structure from Motion[M]//The Interpretation of Visual Motion. MIT Press, 1979: 133-175.
- [14] Mandelbrot B B, Passoja D E, Paul lay A J. The Fractal Character of Fracture Surface of Metals[J]. Nature (S0028-0836), 1984, 308(19): 721-722.
- [15] Schnabel R, Wahl R, Klein R. Efficient RANSAC for Point-Cloud Shape Detection[C]// Computer Graphics Forum, 2007: 214-226.
- [16] Schnabel R, Wahl R, Klein R. Efficient RANSAC for Point-Cloud Shape Detection[C]// Computer Graphics Forum, 2007: 214-226.
- [17] 张毅, 刘旭敏, 隋颖, 等. 基于 K -近邻点云去噪算法的研究与改进[J]. 计算机应用, 2009, 29(4): 1011-1014. Zhang Yi, Liu Xumin, Sui Ying, et al. Research and improvement of denoising method based on K -neighbors[J]. Journal of Computer Applications, 2009, 29(4): 1011-1014.
- [18] 李明磊, 李广云, 宗文鹏. 激光扫描点云准确快速去噪方法[J]. 测绘通报, 2015(12): 27-29. Li Minglei, Li Guangyun, Zong Wenpeng. Accurate and Fast Denoising Method of Laser-Scanned Point Clouds[J]. Bulletin of Surveying and Mapping, 2015(12): 27-29.

(下转第 2677 页)

<http://www.china-simulation.com>