

# 基于改进欧氏聚类的散乱工件点云分割

田青华<sup>1</sup>, 白瑞林<sup>1</sup>, 李杜<sup>2</sup>

<sup>1</sup>江南大学轻工过程先进控制教育部重点实验室, 江苏 无锡 214122;

<sup>2</sup>无锡信捷电气股份有限公司, 江苏 无锡 214072

**摘要** 针对机器人随机箱体抓取过程中场景分割困难的问题, 提出一种基于改进欧氏聚类的散乱工件点云分割方法。采用直通滤波法和迭代半径滤波法进行预处理, 得到去除干扰点后的散乱工件点云; 通过基于法线夹角的边缘检测去除点云中的边缘点, 并使相互碰撞的工件在空间上产生分离; 采用改进的搜索半径自适应欧氏聚类进行点云分割, 得到多个工件点云子集, 基于距离约束将去除的边缘点补齐到点云子集之中, 从而完成点云分割。此外, 线下模板点云注册为分割参数的选取提供依据, 从而保证了分割结果的准确性, 提高了分割速度。结果表明: 基于改进欧氏聚类的散乱工件点云分割方法能够准确地分割出感兴趣的工件, 分割时间约为 696 ms, 满足了工业机器人抓取的实时性要求。

**关键词** 机器视觉; 欧氏聚类; 点云分割; 自适应聚类; 随机箱体抓取; 聚类分割

**中图分类号** TP391.41 **文献标识码** A

**doi:** 10.3788/LOP54.121503

## Point Cloud Segmentation of Scattered Workpieces Based on Improved Euclidean Clustering

Tian Qinghua<sup>1</sup>, Bai Ruilin<sup>1</sup>, Li Du<sup>2</sup>

<sup>1</sup>Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education),  
Jiangnan University, Wuxi, Jiangsu 214122, China;

<sup>2</sup>Xinje Electronic Co., Ltd., Wuxi, Jiangsu 214072, China

**Abstract** Aiming at the difficulty of scene segmentation in the process of robotic random bin picking, a point cloud segmentation method based on the improved Euclidean clustering is proposed. The pass-through filter and the iterative radius filtering are used for the pretreatment to obtain the point cloud of scattered workpieces after removing the interference points. The edge points in the point cloud are removed by the edge detection based on normal angle, and the inter-collision workpieces are separated in space. The improved radius adaptive Euclidean clustering is adopted for the point cloud segmentation to obtain the point cloud subsets of many workpieces. The removed edge points will be put into the point cloud subsets based on the distance constraint, and thus the point cloud segmentation is completed. In addition, the offline template point cloud provides reference for the selection of segmentation parameters, which ensures the accuracy of segmentation results and improves the segmentation speed. The experimental results show that the proposed method can accurately segment the interested workpieces, and the segmentation time is about 696 ms. It is satisfied with the real-time requirement of industrial robot picking.

**Key words** machine vision; Euclidean clustering; point cloud segmentation; adaptive clustering; random bin picking; clustering segmentation

**OCIS codes** 150.5758; 150.6910; 150.1135; 120.5800

**收稿日期:** 2017-06-26; **收到修改稿日期:** 2017-07-24

**基金项目:** 江苏高校优势学科建设工程项目(PAPD)、江苏省产学研前瞻性联合研究项目(BY2015019-38)、江苏省科技成果转化专项资金(BA2016075)

**作者简介:** 田青华(1990—), 女, 硕士研究生, 主要从事机器视觉理论与应用方面的研究。

E-mail: 6151905014@vip.jiangnan.edu.cn

**导师简介:** 白瑞林(1955—), 男, 教授, 博士生导师, 主要从事机器视觉与机器人技术方面的研究。

E-mail: bairuilin@hotmail.com

## 1 引言

随着三维数据获取精度的提高和成本的降低,三维激光扫描技术逐渐应用于随机箱体抓取领域<sup>[1]</sup>。使用三维扫描设备获取箱体内散乱工件表面的点云,结合三维点云处理算法得到单个工件位姿,进而引导工业机器人进行抓取,其效率更高、速度更快、智能化程度更好。但是,箱体内工件杂乱,由于工件间的重叠、碰撞等因素的影响,将单个工件从复杂场景中准确地分离出来,仍是随机箱体抓取过程中尚未解决的难题,分割结果的好坏,会直接影响后续位姿计算的准确性<sup>[2]</sup>。

目前常用的分割方法包括聚类分割、基于模型的分割、基于边缘的分割以及区域生长分割。在实际工业应用中,使用最广泛的是聚类分割,它根据对象的某一特征(如距离、法线、曲率等)对点云进行划分,使同一划分内的点云子集拥有相似的属性。聚类分割方法相比基于模型的分割方法<sup>[3]</sup>,其分割效率更高;相比基于边缘的分割方法<sup>[4]</sup>,其分割结果更加准确、稳健性更好;相比区域生长分割算法<sup>[5-7]</sup>,其更具有明显的速度优势。聚类分割算法的关键在于特征的选择,针对不同的对象选用合适的特征,分割结果更加准确。Squizzato<sup>[8]</sup>针对弯曲近90°的钢板工件,使用欧氏聚类分割算法实现了散乱工件点云的分割。Sansoni等<sup>[9]</sup>在欧氏聚类算法的基础上优化了子集选择,根据分割子集的深度值来判断工件的遮挡与被遮挡关系。佐立营<sup>[10]</sup>将工件点云分割分为结合统计距离和空间聚类实现粗分割,及对欠分割子集通过法线差异提取工件平面特征,实现进一步分割。樊丽等<sup>[11]</sup>将融合法线和激光反射强度作为共同判断准则,实现了林下环境的聚类分割。

考虑到生产效率和实时性要求,工业上通常选用易实现、速度快的欧氏聚类分割算法。但欧氏聚类的邻域搜索半径固定,易出现欠分割或过分割的现象,从而影响分割效果。因此,本文提出基于改进欧氏聚类的散乱工件点云分割方法。通过边缘点去除两碰撞物体之间的连接,结合改进的搜索半径自适应的聚类分割方法,克服了上述缺陷,将边缘点补齐到分割结果之中,保留了工件的完整特征,通过线下模板点云注册为分割选取参数提供依据,保证了分割结果的正确性。该方法在兼顾分割效率和速度的同时保留了原始工件的基本特征,为后续工件位姿定位奠定了良好的基础。

## 2 算法原理

基于改进欧氏聚类的散乱工件点云分割方法的流程包括点云预处理、线下模板点云注册和线上目标点云分割,如图1所示。

### 2.1 点云预处理

编码结构光三维测量属于非接触式测量,易受光照变化、外界环境以及相机自身等因素的影响,获取的点云含有大量冗余点、离群点。冗余点多来自箱体底部平面点和箱体壁点,属于非感兴趣部分,离群点是指远离感兴趣目标点,来源于未完全消除的冗余点和测量中的误差点,其点云密度远小于感兴趣区域。二者均会对场景分割以及后续目标位姿估计的准确性造成影响,因此需要进行预处理。预处理步骤为:1)采用 $k$ 维( $k$ -D)树邻近点搜索算法<sup>[12]</sup>建立点云中点与点之间的拓扑关系,实现邻近点的快速搜索;2)在3个坐标维度上分别通过直通滤波器去除来自箱体底部平面、箱体壁的冗余点;3)半径滤波器滤波<sup>[7]</sup>是通过检测目标点邻域半径内的近邻点数目是否小于预设阈值消除离群点的。该方法需要设定邻近点数目阈值 $T_{th}$ ,文献<sup>[10]</sup>指出实际点云中边缘上点的邻近点数量接近搜索区域近邻点数量最大值的50%,因此,通过统计不同邻域下模板点云的最大邻近点数目,在给定邻域时,选取接近其对应最大邻近点数目50%的值作为阈值,可以去除大部分离群点。但是该方法经过一次滤波很难完全去除离群点,因此,本文提出迭代半径滤波方法,在初始滤波的基础上引入迭代思想,进行多次重复滤波,直到滤波前后点云数目差异满足给定值(如10),终止迭代。

为验证预处理方法的有效性,以图2(b)所示散乱塑料工件点云(共56184个点)为例进行实验,图2(c)为去除冗余点后的点云(共13259个点),图2(d)、(e)分别为半径滤波和迭代半径滤波的结果,点云数目分别为11624和11174。迭代半径滤波的离群点去除率比半径滤波提高了3.4%,可见改进后滤波算法能更有效地去除点云中离群点。

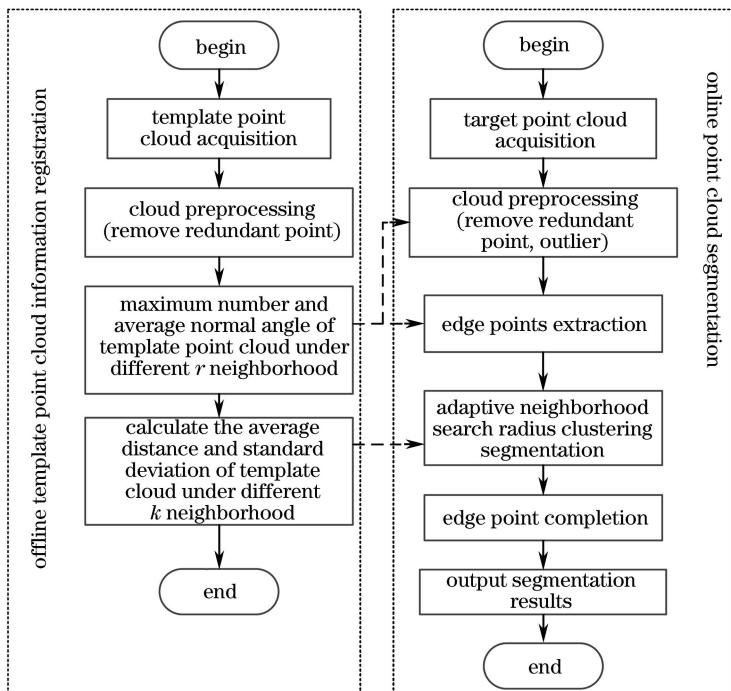


图 1 算法流程图

Fig. 1 Flow chart of the algorithm

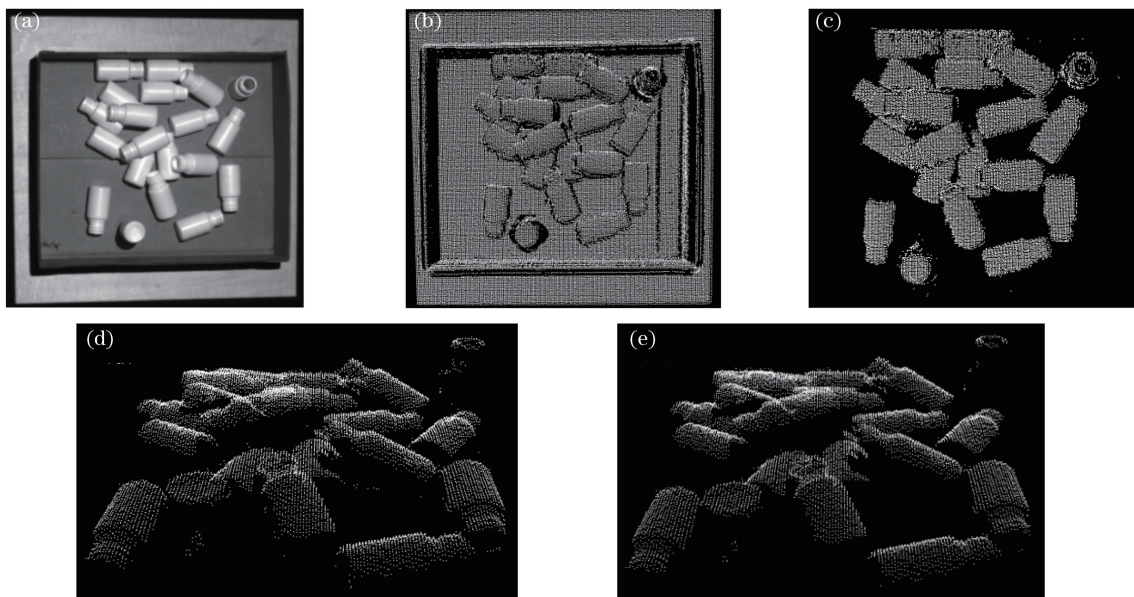


图 2 预处理示意图。(a)原始场景;(b)散落塑料工件;(c)去除冗余后的点云;(d)半径滤波;(e)迭代半径滤波

Fig. 2 Schematic of pretreatment. (a) Original scene; (b) scattered plastic parts; (c) point cloud after removing redundancy;

(d) radius filtering; (e) iterative radius filtering

## 2.2 线下模板点云注册

点云注册通过统计单个对象的某些特征,为后续的杂乱场景分割提供理论依据。当前的分割算法多针对人、树木、建筑物等场景,对象复杂多样,无法单独提取点云进行注册,而随机箱体抓取对象为类别单一的已知物体,易提取单个工件点云。因此,本文提出线下模板点云注册,该方法通过计算模板点云在不同  $k$  邻域下点与邻近点之间的平均距离和标准差、不同  $r$  邻域下最大邻近点数目和点与邻近点之间法线夹角均值,优化后续点云分割的参数选取,有利于提高分割结果的准确性。具体步骤如下:

1) 不同  $k$  邻域下点与邻近点之间的平均距离和标准差

对任意一点  $q_i \in Q$  ( $Q$  为模板点云, 数目为  $N$ ), 搜索距离最近的  $k$  个邻近点, 记作  $\{q_{i1}, q_{i2}, \dots, q_{ik}\}$ , 计算每个邻近点距该点的平均距离:

$$\bar{d}_i = \frac{1}{k} \sum_{j=1}^k \|q_i - q_{ij}\|, \quad (1)$$

遍历点云中所有的点, 得到  $N$  个平均距离值, 求得点云在  $k$  邻域下的平均距离均值  $\bar{d}$  和标准差  $\sigma$ :

$$\begin{cases} \bar{d} = \frac{1}{N} \sum_{i=1}^N d_i, \\ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - \bar{d})^2}. \end{cases} \quad (2)$$

统计不同  $k$  值下点云的平均距离和标准差如表 1 所示。

表 1 不同  $k$  邻域下模板点云的平均距离均值和标准差

Table 1 Average distance and standard deviation of the template point cloud under different  $k$  neighborhood

$k$	1	2	5	10	15	20	25	30
Distance mean /mm	1.0527	1.1660	1.4675	1.8345	2.2154	2.5005	2.7088	2.9438
Standard deviation	0.2441	0.2512	0.3574	0.4777	0.5655	0.6389	0.6557	0.7085

2) 不同  $r$  邻域下最大邻近点数目和点与邻近点之间法线夹角均值

对点云  $Q$  的任意一点, 搜索其  $r$  邻域(小于工件尺寸)内的所有邻近点, 统计其邻近点数目, 遍历点云中的所有点, 得到  $N$  个邻近点数目, 将点云  $r$  邻域下的最大邻近点数目取为最大值。

采用主成分分析(PCA)法<sup>[13]</sup>求取点云中点的法线, 具体步骤是: 对任意一点  $q_i \in Q$ , 搜索其  $r$  邻域半径内所有的邻近点, 设邻近点数目为  $g$ , 记作  $\{q_{i1}, q_{i2}, \dots, q_{ig}\}$ , 利用最小二乘法拟合局部平面:

$$q(n, d) = \arg \min_{(n, d)} \sum_{j=1}^g (n \cdot \bar{q} - d)^2, \quad (3)$$

式中  $n$  为局部平面法线,  $\bar{q} = \frac{1}{n} \sum_{i=1}^n (q_{x_i}, q_{y_i}, q_{z_i})$  为邻近元素的三维质心,  $d$  表示局部平面到坐标原点的距离。

令局部平面过  $\bar{q}$  点,  $n$  为单位法线, 则可将问题转化为求协方差矩阵  $M$  的特征值:

$$|M| = \frac{1}{g} \sum_{j=1}^g (q_{ij} - \bar{q})(q_{ij} - \bar{q})^T, \quad (4)$$

$$|M| \cdot \mathbf{v}_j = \lambda_j \cdot \mathbf{v}_j, j \in \{0, 1, 2\}, \quad (5)$$

式中  $\lambda_j$  和  $\mathbf{v}_j$  分别为协方差矩阵的特征值和特征向量, 且  $\lambda_0 \leq \lambda_1 \leq \lambda_2$ 。具有最小特征值的特征向量即  $q_i$  点的法线, 记作  $\mathbf{v}_i$ 。点  $q_i$  的  $g$  邻近点对应法线分别为  $\{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \dots, \mathbf{v}_{ig}\}$ , 计算点  $q_i$  与  $r$  邻域内邻近点法线夹角的均值、点云的  $r$  邻域下法线夹角均值:

$$\alpha_i = \frac{1}{g} \sum_{j=0}^g \arccos \frac{\mathbf{v}_i \cdot \mathbf{v}_{ij}}{|\mathbf{v}_i| |\mathbf{v}_{ij}|}, \quad (6)$$

$$\partial = \frac{1}{N} \sum_{i=1}^N \alpha_i. \quad (7)$$

统计不同  $r$  邻域下点云的法线夹角均值如表 2 所示。

表 2 不同  $r$  邻域下模板点云的最大邻近点数目和法线夹角均值

Table 2 Maximum number and average normal angle of template point cloud under different  $r$  neighborhood

$r$ /mm	1	2	3	4	5	6	7	8	9	10
Maximum number	4	11	25	45	68	94	125	163	208	245
Average normal angle /( $^\circ$ )	58.08	13.49	5.42	3.42	2.91	2.73	2.89	3.24	3.66	4.19

### 2.3 散乱工件点云分割

基于改进欧氏聚类的散乱工件点云分割的优势在于: 通过去除边缘点消除碰撞物体间的连接; 提出自适应搜索半径的聚类分割, 在工件点云的非边缘区域选取较大的搜索半径, 并在边缘区域选取较小的搜索半

径,从而在保证分割效果的基础上提高算法的速度;基于距离约束方法仅补齐感兴趣工件边缘点,算法易实现、耗时少。具体实现步骤如下。

### 1) 基于法线夹角的边缘点去除

仅去除碰撞区域的点,达到将相邻、碰撞的两个工件在空间上区分开来的目的,从而避免出现聚类过程中多个工件归为一类的现象,对无碰撞、独立的工件则不需要去除边缘点。如图 3 所示,相互碰撞、覆盖的工件点云在碰撞区域(红色区域内)点的法线具有明显差异。考虑到这一因素,采用邻域法线夹角均值约束法提取目标点云边缘点。其难点在于法线夹角阈值的选取,若阈值过大,则不能将工件分离;若阈值过小,则会将工件分离成多个。因此利用 2.2 节线下模板点云注册为阈值选取的依据,经实验验证,将邻域法线夹角均值的最小值设为阈值,可以准确地去除物体接触区域的边缘点,具体实施步骤如下:首先,构建目标点云  $P$  的  $k$ -D 树,根据表 2 确定模板点云邻域法线夹角均值的最小值,设阈值为  $\partial_{th}$ ,读取其对应的邻域半径  $r_{min}$ ;然后,对点  $q_i \in Q$ ,采用 2.2 节步骤 2) 计算该点的  $r_{min}$  邻域下的法线夹角均值为  $\partial_{q_i}$ ,若  $\partial_{q_i} > \partial_{th}$ ,则  $q_i$  点为边缘点,反之,则为非边缘点,遍历点云中所有点,最终得到边缘点云  $P_1$  和非边缘点云  $P_2$ ,且两者满足  $P_1, P_2 \in P, P_1 \cup P_2 = P$ 。

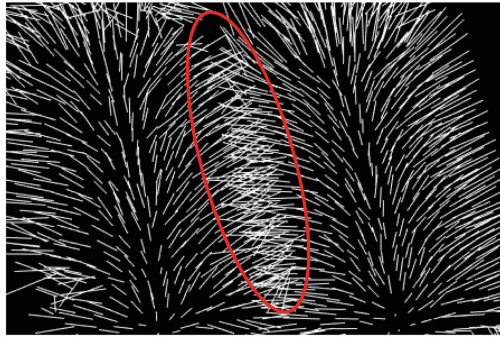


图 3 相互接触工件点云法线示意图

Fig. 3 Normal vector of the point cloud of the contacted workpiece

### 2) 搜索半径自适应的欧氏聚类分割

搜索半径自适应的思想是:在聚类过程中,对位于工件内部区域的点选择较大的搜索半径,对位于工件边缘区域的点选择较小的搜索半径。判断点云中的点是否位于边缘区域需要用到 2.2 节步骤 1) 中模板点云的平均距离和标准差,若点与其  $k$  邻近点之间的距离均值在模板点云的平均距离波动范围内,则为工件边缘点,取其距离均值为搜索半径,反之,则取最远邻近点和该点的距离作为搜索半径。总结搜索半径自适应的欧氏聚类分割算法步骤如下。

① 构建非边缘点云  $P_2$  的  $k$ -D 树,初始化邻近点数目为  $k$ ,由表 1 得到点云  $k$  邻域下的距离均值  $\bar{d}$  和标准差  $\sigma$ ,初始化最小聚类数目  $n_{min}$  和最大聚类数目  $n_{max}$ 。

② 新建一个空的点云索引向量  $s_{cluster\_indices}$  用于存储聚类结果,新建一个整型向量  $q_{queue}$  并初始化为空,用于存储单次聚类结果;新建一个 bool 型向量  $p_{processed}$ ,尺寸和点云  $P_2$  相同,初始化向量为 false,用于表示点云是否被处理过。

③ 初始化  $i=0$ ,对  $p_i \in P_2$ ,执行以下步骤。

a 初始化  $c=0$ ,将  $p_i$  点的索引加入当前序列  $q_{queue}[c]$ ,标记该点为已分类,即  $p_{processed}[i]=true$ 。

b 根据序列  $q_{queue}$  中第  $c$  个元素存储的点索引找到其在点云  $P_2$  中的对应点  $p_c$ ,在  $k$ -D 树中,搜索距离  $p_c$  点最近的  $k$  个邻近点,求  $k$  个邻近点与该点的距离  $\{D_{dist}[j], j=1, 2, \dots, k\}$ ,取均值为  $A_{SinglePoint}$ ,计算固定距离阈值:

$$D = (\bar{d} + \mu\sigma), \quad (8)$$

式中  $\mu$  为比例因子。

若  $A_{SinglePoint} \leq D$ ,则  $p_c$  点为非边缘点,自适应搜索半径  $R_{th}$  表示为

$$R_{th} = \text{Max}\{D_{dist}[j], j=1, 2, \dots, k\}, \quad (9)$$

反之,自适应搜索半径  $R_{th}$  则表示为

$$R_{th} = \frac{1}{k} \sum_{l=1}^k D_{dist} [l]。 \quad (10)$$

c 根据步骤 b 求得的搜索半径  $R_{th}$ ,将位于该半径范围内邻近点的索引加入序列  $q_{queue}$ ,并标记邻近点为已分类, $c=c+1$ 。

d 重复步骤 b、c,直到  $q_{queue}$  中对应的所有的点都被处理过,根据  $n_{min} \leq n_{queue\_size} \leq n_{max}$  判断该聚类结果是否满足数目要求,若满足,则将  $q_{queue}$  中所有的点索引放入点云索引向量  $s_{cluster\_indices}$  中,清空队列  $q_{queue}$ ,反之,则舍弃并清空序列  $q_{queue}$ , $i=i+1$ 。

④ 重复步骤 ③,直到点云  $P_2$  中所有的点都已处理完毕,将聚类结果存放在  $\{s_{cluster\_indices}[0], s_{cluster\_indices}[1], \dots, s_{cluster\_indices}[m-1]\}$  中,将每组点云索引向量中索引在  $P_2$  中对应的点保存成点云输出,记作  $\{C_{cluster\_1}, C_{cluster\_2}, \dots, C_{cluster\_m}\}$ 。

### 3) 边缘点补齐

经过步骤 2) 处理,目标点云  $P_2$  被分割成  $m$  个点云子集,每个子集都代表一个目标工件,但此时所有子集所表示的工件缺乏边缘部分,边缘的缺失会增大位姿估计的误差,因此需要进行边缘点的修补。根据边缘点与其在点云子集中最邻近点的距离是否满足阈值进行边缘点补齐,过程如下:

① 初始化  $j=1$ ,设定最邻近点距离阈值  $T_{thk}$ 。

② 构建子集  $C_{cluster\_j}$  的  $k$ -D 树,对边缘点云  $P_1$  中任意一未处理点  $P_{li}$ ,计算该点到子集  $C_{cluster\_j}$  中最邻近点的欧氏距离  $d_j$ ,比较  $d_j$  和  $T_{thk}$  大小,若  $d_j < T_{thk}$ ,则将  $P_{li}$  点归入子集  $C_{cluster\_j}$ ,并标记该点已处理,反之取下一个边缘点。

③ 遍历点云  $P_1$  中的数据点,检测所有属于子集  $C_{cluster\_j}$  的边缘点,完成边缘点补齐, $j=j+1$ 。

④ 重复步骤 ②、③,直到所有子集边缘点补齐完全,输出补齐边缘点后的点云子集,完成点云分割。

## 3 实验与分析

### 3.1 一组工件点云分割

为验证分割算法的有效性,采用文献[14]的点云数据采集系统获取箱体内部散乱堆放工件点云和单个工件点云进行实验,工件为柱形塑料瓶,瓶身直径为 2.15 cm、长为 4.7 cm。分别采用欧氏聚类分割算法和区域生长分割算法对散乱工件点云进行聚类分割,作为对比实验。实验所用 PC 机安装 64 位的 Windows 7 操作系统,配置为 Intel(R) Core(TM) i3-3220 CPU、3.3 GHz、8.00 GB 内存,使用编程语言为 C++,软件平台为 64 位的 VS2013,涉及库文件为点云库 PCL1.8.0。

模板点云如图 4(a) 所示,共 734 个点,对目标点云进行分割,要求分割出的单个工件点云数目不少于模板点云数目的 75%,方可判定为待抓取对象。对预处理后的散乱工件点云[图 2(e)]按照 2.3 节方法进行分割,其中相关参数阈值如表 3 所示。

表 3 本文算法实验参数

Table 3 Experimental parameters of the proposed method

Parameter	$\partial_{th} / (^\circ)$	$r_{min}$	$k$	$n_{min}$	$n_{max}$	$\mu$	$T_{thk}$
Threshold	2.73	6	5	550	5000	1.5	3

根据模板点云信息注册优化点云分割参数选取标准为:根据表 2 将模板点云法线夹角均值最小值设为法线夹角阈值,即  $\partial_{th}=2.73$ ,此时对应的邻域半径为  $r_{min}=6$  即为提取边缘点所用到的搜索半径;文献[8]指出,聚类分割时,良好的半径阈值已经被证明应稍大于输入点云分辨率(应超过 10%),实际上,该参数对于不同的对象会进行微调。由表 1 可知,所获取到的点云的平均点间距对应  $k=1$  的平均距离,约为 1 mm,因此半径阈值应大于 1.1 mm。根据表 1 选取邻近点数目为 5,满足条件且计算量小;最小聚类数目应大于模板点云数目的 75%,即 550,最大聚类应远大于模板点云数目;权值  $\mu$  用来计算自适应搜索半径,经多次重复实验得到;边缘点补齐时,最近邻点阈值  $T_{thk}$  选取 3 倍点云分辨率。

边缘点提取结果如图 4(b) 所示,其中白色为边缘点云,红色为非边缘点云。对非边缘点云进行搜索半

径自适应的聚类分割,结果如图 4(c)所示,每种颜色代表一个聚类子集,可以看出,每个子集均只包含一个目标工件,没有出现欠分割现象,对各分割子集补齐边缘点后分割结果如图 4(d)所示。

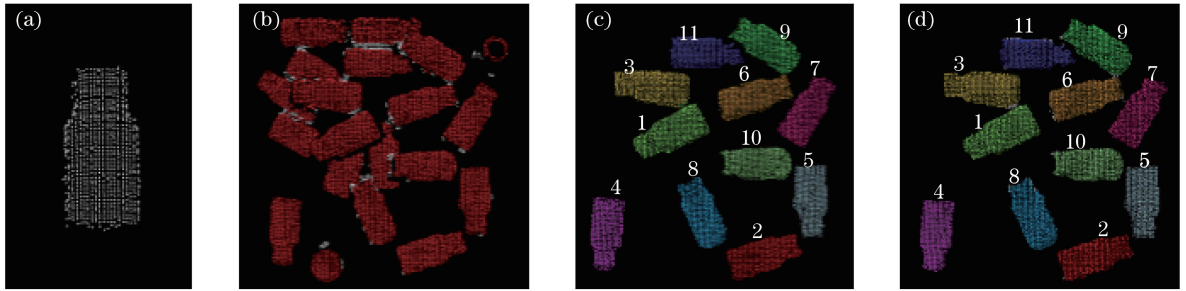


图 4 点云分割实现。(a)模板点云;(b)边缘点提取结果;(c)基于搜索半径自适应聚类;(d)最终分割结果

Fig. 4 Implementation of point cloud segmentation. (a) Template point cloud; (b) edge-point extraction results; (c) adaptive clustering based on search radius; (d) final segmentation result

对比实验 1:对相同预处理后的点云使用欧氏聚类分割算法进行分割实验,取相同参数  $k=5, \mu=1.5$ ,由(8)式计算得到邻域搜索半径为  $r_E=2.01$  mm,如图 5 所示。

对比实验 2:对相同预处理后的点云使用区域生长分割算法进行分割实验,采用邻近点法线夹角差异作为既定准则,参数选取和本文算法一致,如图 6 所示。

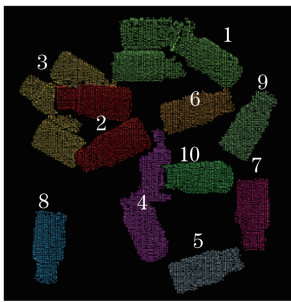


图 5 欧氏聚类分割结果

Fig. 5 Segmentation results of Euclidean clustering

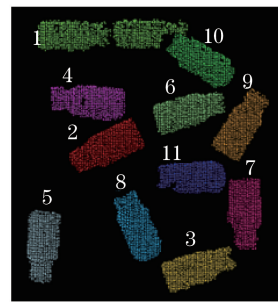


图 6 区域生长分割结果

Fig. 6 Segmentation results of regional growth

为定量分析 3 种方法的分割效果,将各方法分割出的子集数目进行统计,如表 4 所示。其中,本文算法中,before 表示搜索半径自适应欧氏聚类分割算法分割出的各子集的数目,point-covering 表示待补齐到对应子集的边缘点的数目,after 表示各分割子集补齐边缘点后的数目。

表 4 3 种算法分割结果统计

Table 4 Segmentation results of three algorithms

Sub-cloud	1	2	3	4	5	6	7	8	9	10	11	
Euclidean clustering	1819	1405	1358	1120	679	670	660	658	650	634	—	
Region growing	1017	703	693	677	667	661	659	653	652	631	624	
Proposed method	Before	707	695	683	658	657	655	654	646	632	626	572
	Point-covering	15	3	2	0	1	19	0	14	4	9	50
	After	722	698	685	658	658	674	654	660	636	635	622

### 3.2 多组工件点云分割

为进一步验证本文分割算法的有效性,进行了多次重复实验,实验对象包括直径为 2.15 cm、长为 4.7 cm 的柱形塑料瓶和直径为 2.5 cm、长为 6 cm 的柱形塑料瓶及木质柱体。每种对象获取 3 组不同散落场景下的三维点云数据进行实验,统计实验结果如表 5 所示。表 5 中 3 种实验对象分别对应目标 1、2 和 3,其中 A/B/C 形式的的数据记录了 3 种分割方法的结果,A 为欧氏聚类分割结果,B 为区域生长分割结果,C 为本文算法分割结果。在目标数量相同的条件下,分别记录 3 种算法分割出的子集个数、正确分割出的子集个数、分割正确率和算法耗时,用于算法性能分析。

为了直观地观测各算法分割效果,图7给出了3种实验对象在场景2下不同算法的分割结果。

表5 3种算法分割效果对比

Table 5 Comparison of segmentation results of three algorithms

Object	Scene	Number of targets	Number of clusters	Number of correct clusters	Proportion of accuracy /%	Time /ms
1	1	20	10/11/11	6/10/11	60.0/90.9/100	902/2314/635
	2	20	10/11/10	4/9/10	40.0/81.8/90.9	866/1879/481
	3	20	9/15/15	6/15/15	66.7/100/100	1053/2882/707
2	1	20	10/9/12	5/9/12	50.0/100/100	761/2997/641
	2	20	8/10/12	4/10/12	50.0/100/100	932/2968/850
	3	20	9/10/10	6/8/9	66.7/80.0/90.0	818/3282/717
3	1	20	11/11/11	6/11/11	54.5/100/100	1050/4279/688
	2	20	9/9/9	8/9/9	88.9/100/100	931/3742/861
	3	20	10/9/9	6/9/9	60.0/100/100	983/4112/687
Average value		20	9.6/10.6/11	5.7/10/10.9	59.6/94.7/97.9	922/3162/696

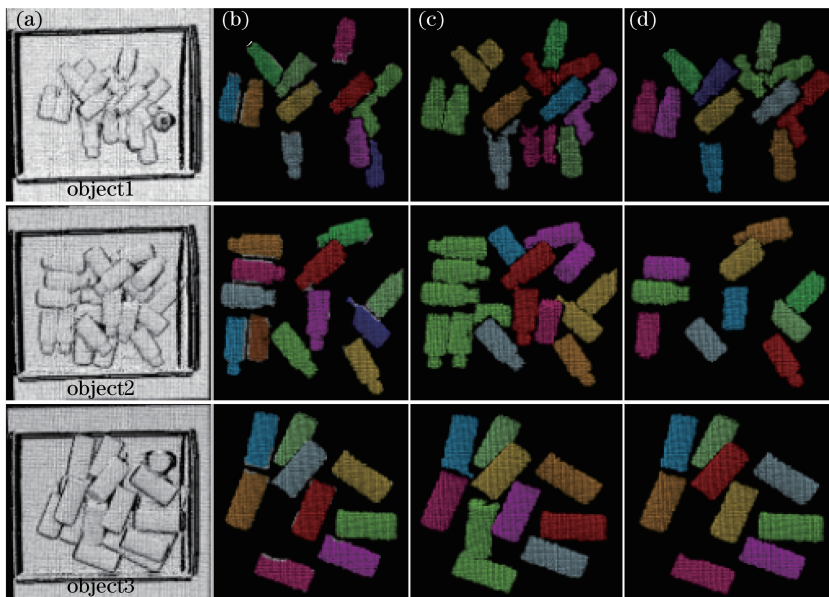


图7 3种算法不同实验对象的分割结果图。(a)实验对象图;(b)本文算法分割结果;

(c)欧氏聚类算法分割结果;(d)区域生长算法分割结果

Fig. 7 Segmentation results of three algorithms with different experimental objects. (a) Experimental objects;

(b) segmentation results of proposed algorithm; (c) segmentation results of Euclidean clustering algorithm;

(d) segmentation results of regional growth algorithm

### 3.3 实验结果分析

1) 在目标工件数目相同的情况下,本文算法能够准确地分割出更多的目标,分割效率更高。表4对一组数据处理结果显示,欧氏聚类算法共分割出10个目标子集,其中子集1、2、3出现了欠分割现象;区域生长算法共分割出11个目标子集,其中子集1出现了欠分割现象;本文算法共分割出11个目标子集,没有出现欠分割现象;由表5多次实验的统计结果也可以看出,本文算法的正确分割率高于另外两种分割算法。

2) 本文算法分割出的工件更完整。对比表4同一工件在3种算法下的分割结果,欧氏聚类分割算法正确分割出的工件为子集5、6、7、8、9、10,分别对应区域生长分割算法分割出的子集3、6、7、5、9、11,本文算法分割出的子集2、6、5、4、7、10,对比对应子集点的数目,结果表明,本文算法分割效果优于欧氏聚类分割算法和区域生长分割算法。

3) 本文算法耗时少,优于另外两种算法。表5中算法耗时是连续执行程序5次的均值,可以看出,本文算法耗时远远少于区域生长分割算法,相较于欧氏聚类分割算法,速度提高了24.5%。



## 4 结 论

提出基于改进欧氏聚类的散乱工件点云分割方法,用于解决随机箱体抓取过程中的场景分割问题,得出如下结论:

1) 提出迭代半径滤波方法去除离群点,在初始半径滤波的基础上引入迭代思想,以滤波前后点云数目满足给定值为约束条件,进行多次重复半径滤波,相较于改进前离群点去除更彻底。

2) 采用先去除边缘点,再聚类分割,最后补齐边缘点的思想,避免了欧氏聚类易出现欠分割或过分割的现象,在聚类分割时,提出基于自适应邻域搜索半径的聚类方法,提高了分割速度,边缘点补齐则保留了工件的表面特征,有利于提高后续位姿定位准确性。

3) 提出线下模板点云注册方法,通过计算模板点云在不同邻域下的邻近点平均距离和标准差、邻近点数目最大值和法线夹角均值优化点云分割的参数选取,保证了分割结果的准确性。

4) 结果表明,本文算法能够准确地分割出感兴趣工件,分割时间约为 696 ms,兼顾了准确性和实时性。相较于欧氏聚类分割算法,分割结果准确性提高了 64.26%,速度提高了 24.5%;相较于区域生长分割算法,速度提高了约 4.5 倍,分割结果准确性提高了 3.2%。

## 参 考 文 献

- [1] Bogue Consultant R. Random bin picking: has its time finally come[J]. *Assembly Automation*, 2014, 34(3): 217-221.
- [2] Nguyen A, Le B. 3D point cloud segmentation: a survey[C]. *IEEE Conference on Robotics, Automation and Mechatronics*, 2013: 225-230.
- [3] Zhu Dehai. Learning tutorials of point cloud library PCL[M]. Beijing: Beihang University Press, 2012: 338-350.  
朱德海. 点云库 PCL 学习教程[M]. 北京: 北京航空航天大学出版社, 2012: 338-350.
- [4] Bao T, Zhao J L, Xu M. Step edge detection method for 3D point clouds based on 2D range images[J]. *Optik-International Journal for Light and Electron Optics*, 2015, 126(20): 2706-2710.
- [5] Vo A V, Truong-Hong L, Laefer D F, *et al.* Octree-based region growing for point cloud segmentation[J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2015, 104: 88-100.
- [6] Wang Guo, Sha Congshu, Wang Jian. Study on segmentation of building facade considering local point cloud density[J]. *Laser & Optoelectronics Progress*, 2015, 52(6): 061001.  
王果, 沙从术, 王健. 考虑局部点云密度的建筑立面自适应分割方法[J]. *激光与光电子学进展*, 2015, 52(6): 061001.
- [7] Lu Weixin, Wan Youchuan, He Peipei, *et al.* Extracting and plane segmenting buildings from large scene point cloud [J]. *Chinese J Lasers*, 2015(9): 0914004.  
卢维欣, 万幼川, 何培培, 等. 大场景内建筑物点云提取及平面分割算法[J]. *中国激光*, 2015, 42(9): 0914004.
- [8] Squizzato S. Robot bin picking: 3D pose retrieval based on point cloud library[D]. Padua: University of Padova, 2012.
- [9] Sansoni G, Bellandi P, Leoni F, *et al.* Optoranger: a 3D pattern matching method for bin picking applications[J]. *Optics and Lasers in Engineering*, 2014, 54: 222-231.
- [10] Zuo Liying. Research on automatic identification and location of scattered parts for robot picking[D]. Harbin: Harbin Institute of Technology, 2015.  
佐立营. 面向机器人抓取的散乱零件自动识别与定位技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2015: 28-36.
- [11] Fan Li, Liu Jinhao, Huang Qingqing. Point cloud segmentation algorithm based on feature fusion used for understory environments[J]. *Journal of Beijing Forestry University*, 2016, 38(5): 133-138.  
樊丽, 刘晋浩, 黄青青. 基于特征融合的林下环境点云分割[J]. *北京林业大学学报*, 2016, 38(5): 133-138.
- [12] Shen J, Liu J, Zhao R, *et al.* A kd-tree-based outlier detection method for airborne LiDAR point clouds[C]. 2011 *IEEE International Symposium on Image and Data Fusion*, 2011: 1-4.
- [13] Nurunnabi A, West G, Belton D. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data[J]. *Pattern Recognition*, 2015, 48(4): 1404-1419.
- [14] Shi Aijun, Bai Ruilin, Tian Qinghua. 3D measurement based on structured light using a combination of gray code and line-shift patterns[J]. *Opto-Electronic Engineering*, 2016, 43(11): 26-32.  
石爱军, 白瑞林, 田清华. Gray 码结合线移的结构光三维测量[J]. *光电工程*, 2016, 43(11): 26-32.