

文章编号:1001-5078(2024)11-1702-07

· 激光应用技术 ·

# 基于改进 LCCP 的堆叠目标点云分割算法

高显棕, 金建辉

(昆明理工大学 信息工程与自动化学院, 云南 昆明 650500)

**摘要:**点云分割作为无序分拣任务中的一个重要处理步骤,其分割精度直接影响后续的目标识别与姿态估计准确率。针对传统 LCCP 算法在物体复杂堆叠场景下分割效果不佳的问题,本文提出了一种基于改进 LCCP 的点云分割算法,首先使用改进的 VCCS 算法将点云划分为超体素,通过融入高斯曲率信息,进一步改善超体素容易跨越物体边界的问题,然后判定邻接超体素的凹凸连接关系,为了进一步减小噪声的影响,对于所有体积小于给定阈值的超体素,判定其与所有邻接超体素间的连接关系,合并所有凸连接的超体素,得到最终分割结果。实验结果表明,本文方法相比于 LCCP 和 CPC 算法在分割精确率上提升了 3.1% ~ 22%,且算法整体性能有明显提升。

**关键词:**点云分割;高斯曲率;超体素;凹凸连接

**中图分类号:**TN249;TP391 **文献标识码:**A **DOI:**10.3969/j.issn.1001-5078.2024.11.008

## Stacked target point cloud segmentation algorithm based on improved LCCP

GAO Xian-zong, JIN Jian-hui

(School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

**Abstract:**As an essential processing step in unordered picking tasks, point cloud segmentation directly impacts the subsequent accuracy of object recognition and pose estimation. To address the problem of inadequate segmentation performance of the traditional LCCP algorithm in complex object stacking scenarios, an improved LCCP point cloud segmentation algorithm that incorporates Gaussian curvature information is proposed in this paper. Initially, an enhanced VCCS algorithm is employed to partition the point cloud into super-voxel, and by integrating Gaussian curvature information, the issue of super-voxel easily crossing object boundaries is further addressed. Subsequently, concave-convex connectivity among adjacent super-voxel blocks is determined, followed by the merging of all convexly connected super-voxel to form the final segmentation results. The experimental results demonstrate that the method improves segmentation precision by 3.1% to 22% compared to LCCP and CPC, with a noticeable enhancement in overall algorithm performance.

**Keywords:**point cloud segmentation; Gaussian curvature; super-voxel; concave-convex connectivity

### 1 引言

随着机器人分拣技术的迅速发展,越来越多的

新问题需要解决,无序分拣便是其中备受关注的的一个。无序分拣也叫“拣箱问题”,其包含了多个子问

**基金项目:**国家自然科学基金项目(No. 61364008);云南省基础研究计划重点项目(No. 202101AS070016)资助。

**作者简介:**高显棕(1996-),男,硕士研究生,研究方向为点云数据处理,无序分拣等。E-mail:1013295985@qq.com

**通讯作者:**金建辉(1974-),男,硕士,高级实验师,硕士生导师,主要研究方向为电子科学与技术,图像处理等。

E-mail:2438102125@qq.com

收稿日期:2024-02-27

题,如目标识别,目标分割,姿态估计,运动规划等<sup>[1]</sup>。顾名思义,“拣箱问题”要解决的即是如何分拣箱体内杂乱摆放的物体,通常的解决方法是使用深度相机或雷达等传感器获取目标的点云数据,然后对数据进行处理,最后完成定位和抓取操作。

点云分割不仅能够提高识别和定位的准确性,还大大降低了后续计算的复杂度。近年来基于深度学习的点云分割方法发展迅速,并在多个公开数据集上取得了很好的效果<sup>[2-4]</sup>,但这种方法存在着数据集获取困难,训练时间长,泛用性差等问题。而非学习的方法更适合解决某一类场景的分割问题,其中基于区域的分割方法对堆叠点云这类复杂场景的分割效果最好。基于区域的点云分割方法是一种利用点云数据中的邻域信息来实现分割的技术。该方法通过将具有相似属性的附近点聚集成不同的区域,实现对点云数据的分割和分类。Fan 等<sup>[5]</sup>提出了一种自适应点云分割算法,能够在没有人干预的情况下自行选取种子点,并使用改进的区域生长算法对不同点云模型进行了分割实验。针对相互紧贴的目标,Zhao 等<sup>[6]</sup>提出了一种基于边缘限制区域生长的点云分割算法,该方法选取曲率最小的点作为初始种子,通过双生长准则控制种子精确生长,实现了对相邻物体的精确分割。杨琳等<sup>[7]</sup>结合超体素和区域生长算法对植物点云进行分割,解决了传统区域生长算法分割精度低的问题。

上述方法主要利用点云的颜色、空间距离、法向量和曲率等信息聚类具有相似属性的点,而 Stein 等<sup>[8]</sup>提出局部凸性连接生长算法(Locally Convex Connected Patches, LCCP),利用超体素邻接块的凹凸性进行点云分割,在多物体堆叠的场景下取得了不错的效果。该算法先使用体素云连通性分割算法<sup>[9]</sup>(Voxel Cloud Connectivity Segmentation, VCCS)将点云划分为超体素,记录超体素邻接块,判定邻接超体素间的连接关系,最后将所有凸连接超体素合并形成最终分割结果。但 VCCS 算法生成的超体素容易跨过物体边界,且邻接超体素块间的凹凸关系判定容易受到噪声体素影响,导致出现过分割或欠分割。为了解决上述缺陷,本文对传统 LCCP 算法进行了改进,通过在超体素聚类中融入高斯曲率信息,避免超体素向物体边界外生长,改善欠分割和过分割问题。

## 2 点云的超体素聚类

超体素聚类是一种将具有相似特征的体素组合到一起的方法,与其他分割手段不同,超体素聚类的目的并不是分割出某种特定物体,而是对点云进行过分割,将场景点云划分成许多小块,并研究每个小块之间的关系。在进行语义分割任务前加入超体素聚类能有效降低噪声的影响,并减少后续算法的复杂度,提高算法可靠性。

### 2.1 VCCS 算法

点云不同于图像,其不存在类似像素一样的邻接关系,所以 VCCS 算法在进行聚类之前,首先利用八叉树结构将无序的点云体素化,建立体素间的邻接关系。体素对应于二维图像中的像素,在三维空间中表现为一个个立方体,立方体的分辨率为  $R_{\text{voxel}}$ 。

VCCS 算法是一种典型的超体素聚类算法,它首先将体素化后的场景点云划分为分辨率为  $R_{\text{seed}}$  的三维空间网格,选取与网格中心相距最近的体素作为种子体素,为了去除因噪声产生的错误种子体素,VCCS 算法还以每一个种子体素为中心建立半径为  $R_{\text{search}}$  的搜索区域,搜索区域内的体素个数,若数量小于设定阈值,则将该种子体素去除。

确定好所有的初始种子体素后,需要定义聚类规则,VCCS 算法定义了相似性距离,综合考虑体素间的颜色、空间距离、几何特征的相似性。聚类过程如图 1 所示,首先从所有聚类中心开始向外流动,搜索全部邻接体素,计算每个邻接体素与聚类中心的相似性距离,对距离最小的体素进行标记,并将其邻接体素加入到搜索队列中,通过这种方式不断向外搜索,直到遍历所有体素,然后将所有被标记体素加入种子体素中,形成新的聚类中心,重复上述步骤直到聚类中心稳定。

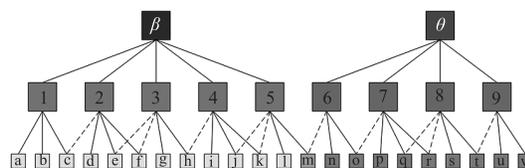


图 1 超体素聚类过程示意图

Fig. 1 Schematic diagram of super-voxel clustering process

### 2.2 融合高斯曲率信息的改进 VCCS 算法

在多目标堆叠场景下,由于物体粘连处形成了一个共同的边界面,导致传统的 VCCS 算法在形成

超体素时容易跨越物体边界导致后续出现错误分割,为了解决这一缺陷本文通过引入曲率信息增强算法对物体边界的识别。高斯曲率是描述曲面弯曲程度的一种度量,表示曲面上某一点附近曲面的几何特征和凹凸性质。当高斯曲率  $K > 0$  时,表示曲面在该点附近呈凸面状,如球面;当  $K < 0$  时,表示曲面在该点附近呈凹面状,如马鞍面;当  $K = 0$  时,则表示曲面在该点附近成某种“平坦”状,即曲面能够展开为平面,如圆柱面。在物体相互粘连处,由于两个物体的表面形成了一个共同的边界,导致曲面的几何形状发生了明显的改变,所以高斯曲率会发生突变。

2.2.1 点云高斯曲率计算

首先求出主曲率值  $k_1$  (最大曲率) 和  $k_2$  (最小曲率),然后通过如下公式计算点云的高斯曲率  $K$  :

$$K = k_1 \cdot k_2 \tag{1}$$

常见的计算点云主曲率的方法有两种:基于二次曲面拟合的曲率计算方法和基于相邻点法向量的曲率计算方法<sup>[10]</sup>,由于点云表面上特定点的所有相邻点决定了点云的局部形状,如果仅通过曲面拟合而不考虑法向量的影响,计算曲率时可能会产生较大误差,故本文选用基于相邻点法向量的曲率计算方法来求取特定点的主曲率。

1) 首先对法向曲率进行局部拟合,对于点云中的每个点  $p$ ,假设  $q_i$  为点  $p$  的第  $i$  个近邻点,建立一个通过点  $p$  和  $q_i$  的密切圆来估计点  $p$  的法曲率  $k_n^i$  :

$$k_n^i = -\frac{\sin\beta}{|pq_i| \sin\alpha} \tag{2}$$

2) 然后对欧拉方程进行最小二乘拟合,建立一个过点  $p$  的切平面  $S$ ,其法向量为  $\vec{N}$ , $\vec{e}_1$  和  $\vec{e}_2$  为点  $p$  处的主方向,对应的主曲率为  $k_1$  和  $k_2$ , $\theta$  为  $\vec{e}_1$  和  $\vec{e}_2$  间的夹角, $\theta_i$  为  $\vec{X}$  和  $\vec{pQ}_i$  间的夹角,其中  $\vec{pQ}_i$  是  $\vec{pq}_i$  在平面  $S$  上的投影。根据欧拉公式,法向曲率和主曲率有如下关系:

$$k_n^i = k_1 \cos^2(\theta_i + \theta) + k_2 \sin^2(\theta_i + \theta), (i = 1, 2, \dots, m) \tag{3}$$

将公式(3)转化为一个优化问题的求解:

$$\min_{k_1, k_2, \theta} \sum_{i=0}^m [k_1 \cos^2(\theta_i + \theta) + k_2 \sin^2(\theta_i + \theta) - k_n^i]^2 \tag{4}$$

其中:

$$\begin{aligned} &k_1 \cos^2(\theta_i + \theta) + k_2 \sin^2(\theta_i + \theta) \\ &= \cos^2\theta_i (k_1 \cos^2\theta + k_2 \sin^2\theta) + 2\cos\theta_i \sin\theta_i \cdot \\ &(k_2 \cos\theta \sin\theta - k_1 \cos\theta \sin\theta) + \sin^2\theta_i (k_1 \sin^2\theta + \\ &k_2 \cos^2\theta) \end{aligned} \tag{4}$$

公式(4)可以变为如下矩阵形式:

$$\min_{\mu} \|M\mu - R\|^2 \tag{5}$$

式中:

$$M_{m \times 3} = \begin{bmatrix} \cos^2\theta_1 & 2\cos\theta_1 \sin\theta_1 & \sin^2\theta_1 \\ \vdots & \vdots & \vdots \\ \cos^2\theta_i & 2\cos\theta_i \sin\theta_i & \sin^2\theta_i \\ \vdots & \vdots & \vdots \\ \cos^2\theta_m & 2\cos\theta_m \sin\theta_m & \sin^2\theta_m \end{bmatrix}$$

$$R_{m \times 1} = \begin{bmatrix} k_n^1 \\ \vdots \\ k_n^i \\ \vdots \\ k_n^m \end{bmatrix}$$

$$\mu = [A, B, C]^T$$

$$A = k_1 \cos^2\theta + k_2 \sin^2\theta$$

$$B = (k_2 - k_1) \cos\theta \sin\theta$$

$$C = k_1 \sin^2\theta + k_2 \cos^2\theta$$

对公式(5)进行最小二乘拟合后,可以得到  $\mu$  的估计值,进一步可以得出:

$$\begin{aligned} \begin{bmatrix} A & B \\ B & C \end{bmatrix} &= \begin{bmatrix} k_1 \cos^2\theta + k_2 \sin^2\theta & (k_2 - k_1) \cos\theta \sin\theta \\ (k_2 - k_1) \cos\theta \sin\theta & k_1 \sin^2\theta + k_2 \cos^2\theta \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \end{aligned} \tag{6}$$

所以主曲率  $k_1$  和  $k_2$  为矩阵  $W = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$  的特征值。

上述变量的几何关系如图2所示。

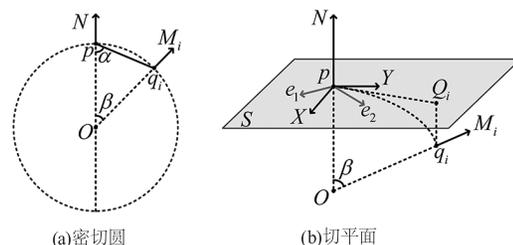


图2 密切圆及切平面上的变量几何关系  
Fig.2 Geometric relationships of variables on closely cut circle and tangent plane

### 2.2.2 基于改进 VCCS 算法的超体素聚类

通过融入高斯曲率信息对 VCCS 算法进行改进,提高算法对物体边界的识别,保证后续分割的效果。算法流程如图 3 所示,首先将点云数据进行体素化,建立相邻点之间的邻接关系并计算体素云的高斯曲率,接着选取初始种子体素,通过加入高斯曲率信息并按照 VCCS 算法的聚类规则进行超体素聚类,最后生成若干个超体素,具体的相似性距离计算公式如下:

$$D = \sqrt{w_c D_c^2 + \frac{w_s D_s^2}{3R_{seed}^2} + w_n D_n^2 + w_k D_k^2} \quad (7)$$

式中,  $D$  表示种子体素与邻接体素的相似性距离,  $D$  值越小,表明体素间的特征属性越相近;  $D_c$  表示颜色上的差异;  $D_n$  表示法向量上的差异;  $D_s$  代表距离上的差异;  $D_k$  表示高斯曲率上的差异;  $w$  表示一系列权重。

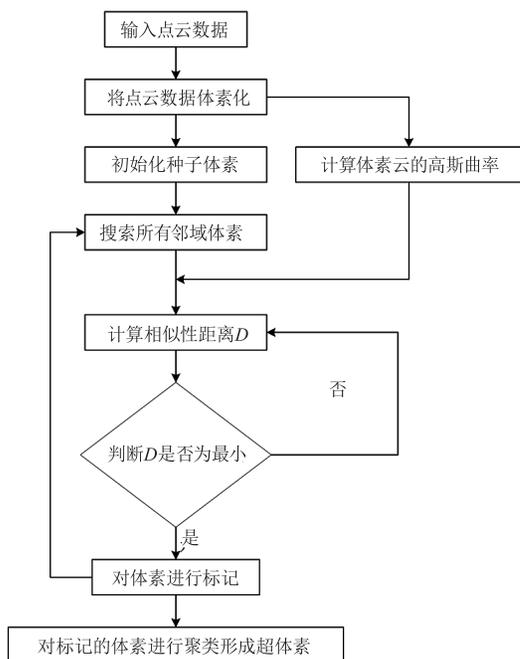


图3 改进 VCCS 算法流程图

Fig. 3 Flowchart of improved VCCS

### 3 基于凹凸关系的区域生长

超体素聚类完成之后,再判断相邻两超体素间连接  $(\vec{p}_i, \vec{p}_j)$  是凸性还是凹性。凹凸关系通过 CC (Extented Convexity Criterion) 和 SC (Sanity Criterion) 准则进行判别,其中 CC 准则利用相邻两超体素形心连线向量和法向量间的夹角来判断凹凸性。

如图 4 所示,法向量  $\vec{n}_1, \vec{n}_2$  与形心连线向量  $\vec{d} = \frac{\vec{x}_1 - \vec{x}_2}{\|\vec{x}_1 - \vec{x}_2\|}$  间的夹角分别为  $\alpha_1, \alpha_2$ , 当  $\alpha_1 < \alpha_2$  时,两超

体素为凸连接,同时两凸连接的超体素存在如下关系:

$$\alpha_1 < \alpha_2 \Rightarrow \cos(\alpha_1) - \cos(\alpha_2) > 0 \Leftrightarrow \vec{n}_1 \cdot \hat{d} - \vec{n}_2 \cdot \hat{d} > 0 \quad (8)$$

式中,  $\hat{d} = \frac{\vec{x}_1 - \vec{x}_2}{\|\vec{x}_1 - \vec{x}_2\|}$ , 同理对于凹连接可以得到如下关系:

$$\alpha_1 > \alpha_2 \Leftrightarrow \vec{n}_1 \cdot \hat{d} - \vec{n}_2 \cdot \hat{d} < 0 \quad (9)$$

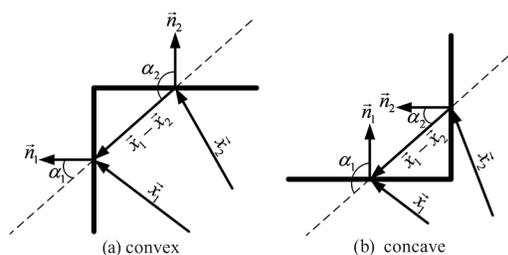


图4 CC 准则判定方式说明

Fig. 4 Explanation of extented convexity criterion determination method

在图像噪声影响下,可能会出现将同一平面上的超体素判定为凹连接的情况,故引入了阈值  $\beta_{Thresh}$  来识别法向量夹角较小的情况,即当满足以下条件时认为两超体素位于同一平面上,应将其合并。

$$\begin{aligned} \beta &= \angle(\vec{n}_1, \vec{n}_2) \\ &= |\alpha_1 - \alpha_2| \\ &= \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2) < \beta_{Thresh} \end{aligned} \quad (10)$$

$$CC_b(\vec{p}_i, \vec{p}_j) = \begin{cases} \text{true}, & (\vec{n}_1 \cdot \vec{n}_2) \cdot \hat{d} > 0 \cup (\beta < \beta_{Thresh}) \\ \text{false}, & \text{其他} \end{cases} \quad (11)$$

定义了基本的凸性准则后,为了进一步减小噪声超体素对分割结果产生的影响,在凹凸性分类时考虑较小超体素的邻域信息。首先选取体积阈值  $V_{Thresh}$ , 对于所有体积小于阈值的超体素  $\vec{p}_k$ , 判断其与所有邻接超体素  $\vec{p}_i, \vec{p}_j$  的凹凸性连接关系,若所有连接都为凸性,则将其合并,否则视为噪声并将其剔除,则额外的凸性准则可以定义为:

$$CC_e(\vec{p}_k) = CC_b(\vec{p}_i, \vec{p}_k) \cap CC_b(\vec{p}_j, \vec{p}_k) \quad (12)$$

完整的 CC 准则定义为:

$$CC(\vec{p}_i, \vec{p}_j) = CC_b(\vec{p}_i, \vec{p}_j) \cap CC_e(\vec{p}_k) \quad (13)$$

仅仅依靠 CC 准则并不能完全判断出两超体素间的凹凸关系,因为两超体素之间除了凸连接和凹连接外,还存在奇异连接,即两超体素仅通过单个点连接,这时候仅使用 CC 准则可能会将其错误识别

为凸连接,于是引入 SC 准则来解决该问题。

如图 5 所示,  $\vec{d} = \vec{x}_1 - \vec{x}_2$  为两超体素形心连线向量,  $\vec{s} = \vec{n}_1 \times \vec{n}_2$  为两法向量的向量积,通过测量  $\vec{d}$  和  $\vec{s}$  间的夹角  $\theta$  就能判定两超体素是否为奇异连接。

首先定义:

$$\begin{aligned} \theta(\vec{p}_i, \vec{p}_j) &= \min(\angle(\vec{d}, \vec{s}), \angle(\vec{d}, -\vec{s})) \\ &= \min(\angle(\vec{d}, \vec{s}), 180^\circ - \angle(\vec{d}, \vec{s})) \end{aligned}$$

从图 5 中可以看出,奇异连接发生在  $\theta$  较小时,因此可以引入阈值  $\theta_{\text{Thresh}}$ , 当  $\theta < \theta_{\text{Thresh}}$  时为奇异连接。

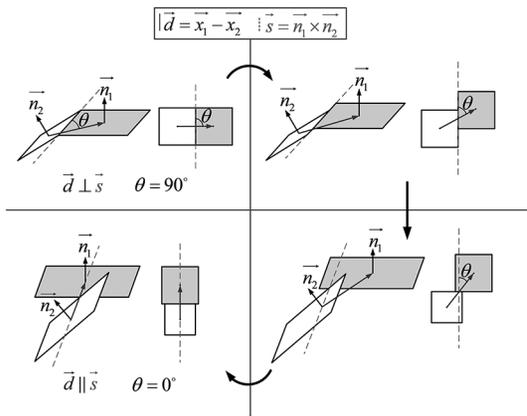


图 5 SC 准则判定方式说明

Fig. 5 Explanation of sanity criterion determination method

与 CC 准则类似,考虑到图像噪声的影响,应当放宽阈值条件,可将阈值设定为两超体素法线夹角的 S 型函数:

$$\theta_{\text{Thresh}}(\beta) = \theta_{\text{Thresh}}^{\max} \cdot (1 + \exp[-a \cdot (\beta - \beta_{\text{off}})])^{-1} \quad (14)$$

式中,  $\beta = \angle(\vec{n}_1, \vec{n}_2)$ ,  $\theta_{\text{Thresh}}^{\max}$ ,  $\beta_{\text{off}}$  和  $a$  为预设值。

所以 SC 准则可表示为:

$$SC(\vec{p}_i, \vec{p}_j) = \begin{cases} \text{true}, & \theta(\vec{p}_i, \vec{p}_j) > \theta_{\text{Thresh}}(\beta(\vec{n}_1, \vec{n}_2)) \\ \text{false}, & \text{其他} \end{cases} \quad (15)$$

最终可得凸连接的判定公式为:

$$\text{conv}(\vec{p}_i, \vec{p}_j) = CC(\vec{p}_i, \vec{p}_j) \cap SC(\vec{p}_i, \vec{p}_j) \quad (16)$$

## 4 实验结果与分析

### 4.1 实验环境与数据集

本实验在 Intel Core i5-9400 CPU、16GB 内存的计算机上进行,选用 C++ 编程语言,并安装点云处理库 PCL 1.10.1。

实验在两个不同环境和物体的数据集上进行检测。数据集 1 为 OSD (Object Segmentation Database) 数据集<sup>[11]</sup>,共包含 111 组 RGB-D 图像和点云,以及每一个场景所出现物体的掩膜图,图像中的物体包括书、纸盒、杯子、塑料瓶等。本实验选择物体数较多的 23 张点云图作为检测对象,并使用 LCCP 算法、CPC (Constrained Plane Cut) 算法和 RG (Region Growing) 算法进行对比实验。

数据集 2 为 BOP 数据集集中的 IC-BIN 图集<sup>[12]</sup>,该数据集包含 114 组 RGB-D 图像以及每一张图像所出现物体的掩膜图,检测对象为堆叠咖啡杯和牛奶盒。从侧面拍摄的图像由于箱体的遮挡对分割结果影响较大,故本文筛选出所有从顶部俯拍的 54 组 RGB-D 图像作为检测图像,并将所有 RGB-D 图像转化为点云数据,同样使用 RG 算法、CPC 算法和 LCCP 算法进行对比实验。

### 4.2 实验结果与分析

图 6 和图 7 分别为两个数据集的彩色图像以及各算法的分割结果,从图 6 结果中可以看出, RG 算法分割效果最差,出现了大量的过分割和欠分割; LCCP 算法和 CPC 算法出现了部分欠分割和过分割的情况;本文方法明显优于 LCCP 算法和 CPC 算法,在前两种方法出现欠分割的区域,本文方法基本都能正确分割,仅出现了一次欠分割的情况,同样过分割的情况也得到了改善。

在图 7 所示分割结果中,由于物体堆叠更为密集, LCCP 算法和 CPC 算法均出现了大量欠分割,其中 LCCP 算法出现了 6 次明显的欠分割,最多有 2 个物体被划分为同一区域; CPC 算法出现了 7 次明显的欠分割,最多有 5 个物体被划分为同一区域;本文方法仅出现了 2 次明显的欠分割,最多有 2 个物体被划分为同一区域。

为了更客观地评价本文方法的分割结果,本文使用精确率 P、召回率 R、 $F_1$  评分作为指标对分割结果进行评价,指标计算方式如下:

$$P = \frac{TP}{TP + FP} \times 100\% \quad (17)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (18)$$

$$F_1 = 2 \frac{P \cdot R}{P + R} \times 100\% \quad (19)$$

其中, TP 为物体上正确分割点的数目; FP 为误分割

点,即所有欠分割和过分割区域点的数目, FN 为漏分割点,即未被完整识别的单个物体上的错误分割点的数目。本实验不考虑部分位于底部被严重遮挡的物体,通过计算最终分割效果对比如表 1 所示。

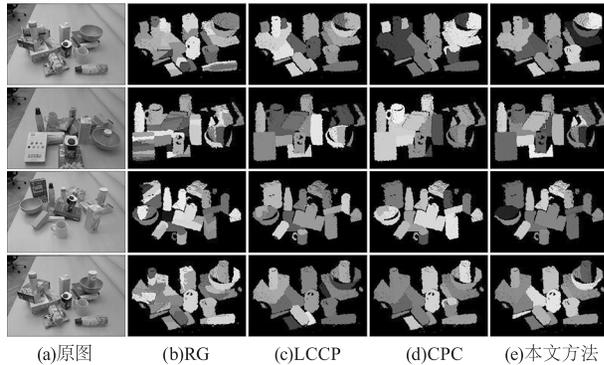


图 6 数据集 1 分割结果示例

Fig. 6 Example of segmentation results for dataset 1

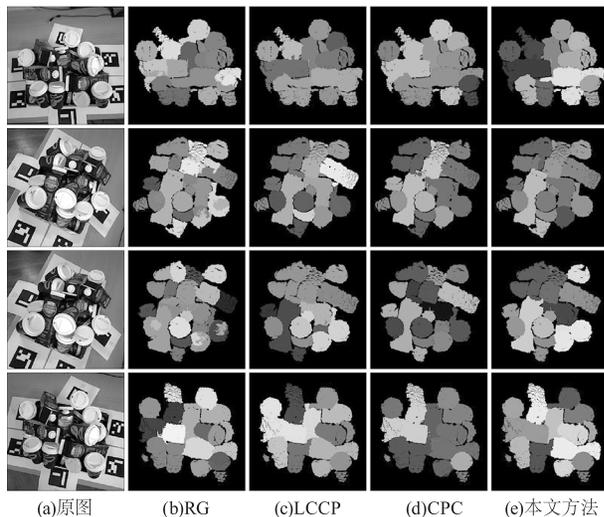


图 7 数据集 2 分割结果示例

Fig. 7 Example of segmentation results for dataset 2

表 1 分割结果对比

Tab. 1 Comparison of segmentation results

数据集	方法	P/%	R/%	$F_1$ /%
1	RG	44.3	48.8	46.4
	LCCP	90.4	<b>94.6</b>	92.5
	CPC	87.2	93.3	90.1
	本文方法	<b>93.5</b>	94.4	<b>93.9</b>
2	RG	36.5	53.7	43.5
	LCCP	63.7	91.2	75.0
	CPC	54.1	<b>92.3</b>	68.2
	本文方法	<b>76.1</b>	91.1	<b>82.9</b>

区域生长算法分割效果很差,显然不适用于复杂环境下的物体分割。与 LCCP 和 CPC 算法相比,

本文方法在精确率上有明显的提升,在数据集 1 中较另两种方法分别提升了 3.1 % 和 6.3 %,在数据集 2 中由于物体摆放更加紧密,精确率提升更为明显,分别提升了 12.4 % 和 22 %。由于 3 种方法都没有出现明显的过分割,所以召回率均达到了 90 % 以上。 $F_1$  评分为精确率和召回率的调和平均值,作为一个综合评价指标,本文方法同样也有明显的提升。

表 2 所示为本文方法与区域生长算法和 LCCP、CPC 算法分割所用平均时长的对比。

表 2 平均分割时间对比

Tab. 2 Comparison of average segmentation time

方法	超像素聚类/s	区域生长/s	平均总时间/s
RG	/	7.42	7.42
LCCP	7.97	4.77	12.74
CPC	7.99	14.66	22.65
本文方法	9.43	5.25	14.68

区域生长算法所用时间最短,但其分割结果较差,不满足复杂场景的应用要求;本文方法虽然用时较 LCCP 算法更广,但仅比 LCCP 算法多出 15 %,同样能满足应用要求。

## 5 结论

针对 LCCP 算法在复杂的物体堆叠场景下点云分割效果不佳的问题,本文提出了一种改进 LCCP 点云分割方法,通过融合高斯曲率信息来提高算法对物体边界的识别,并引入了额外的凸性准则,进一步减小噪声的影响,改善了 LCCP 算法容易造成欠分割的问题。实验表明,所提出的改进算法相比于传统的几种点云分割算法在分割精度上有明显的提升,虽然分割效率有所降低,但仍然能满足应用需求。

## 参考文献:

- [1] Zhai Jingmei, Huang Le. Review of unordered picking technology for robots[J]. Packaging Engineering, 2022, 43(8): 66-75. (in Chinese)  
翟敬梅, 黄乐. 机器人无序分拣技术研究[J]. 包装工程, 2022, 43(8): 66-75.
- [2] Hu X, Li D. Research on a single-tree point cloud segmentation method based on UAV tilt photography and deep learning algorithm[J]. IEEE Journal of Selected

- Topics in Applied Earth Observations and Remote Sensing, 2020, 13: 4111 – 4120.
- [3] Song Y, Sun Z, Li Q, et al. Learning indoor point cloud semantic segmentation from image-level labels[J]. The Visual Computer, 2022, 38(9 – 10): 3253 – 3265.
- [4] Jana A, Subhash H M, Metaxas D. Automatic tooth segmentation from 3D dental model using deep learning: a quantitative analysis of what can be learnt from a single 3D dental model [C]//18th International Symposium on Medical Information Processing and Analysis. SPIE, 2023, 12567: 42 – 51.
- [5] Fan Y, Wang M, Geng N, et al. A self-adaptive segmentation method for a point cloud[J]. The Visual Computer, 2018, 34: 659 – 673.
- [6] Zhao X, Li Q, Zhang M, et al. Closely adjacent object segmentation based on periphery-restrictive region growing [J]. Measurement Science and Technology, 2022, 33(12): 125901.
- [7] Yang Lin, Zhai Ruifang, Yang Xu, et al. Segmentation of plant organs point clouds through super voxel-based region growing methodology[J]. Computer Engineering and Applications, 2019, 55(16): 197 – 203. (in Chinese)
- 杨琳, 翟瑞芳, 阳旭, 等. 结合超体素和区域增长的植物器官点云分割[J]. 计算机工程与应用, 2019, 55(16): 197 – 203.
- [8] Christoph Stein S, Schoeler M, Papon J, et al. Object partitioning using local convexity [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 304 – 311.
- [9] Papon J, Abramov A, Schoeler M, et al. Voxel cloud connectivity segmentation-supervoxels for point clouds [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2027 – 2034.
- [10] Zhang X, Li H, Cheng Z. Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures[J]. Proc. Asia Graph, 2008, (1): 72 – 79.
- [11] Richtsfeld A, Mörwald T, Prankl J, et al. Segmentation of unknown objects in indoor environments [C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012: 4791 – 4796.
- [12] Dumanoglou A, Kouskouridas R, Malassiotis S, et al. Recovering 6D object pose and predicting next-best-view in the crowd [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 3583 – 3592.