

基于改进 FPCC 的实例分割算法^①

冯兴盛, 刘 涌, 唐 磊, 刘文兴

(西南科技大学 计算机科学与技术学院, 绵阳 621010)

通信作者: 刘 涌, E-mail: 190860881@qq.com



摘 要: 3D 点云实例分割是工业自动化中关键的预处理步骤。然而, 在工业抓取场景中存在大量遮挡时, 3D 点云的实例分割网络通常难以区分相似对象。针对这一问题, 本文提出了一种基于 FPCC 的改进算法。本算法有两个分支: 中心点分支, 用于推断实例的中心点, 以及嵌入式特征分支, 用于描述点的特征。分割结果是使用聚类算法获得的。特征增强 (FEH) 模块对中心点预测的准确性起着至关重要的作用。该模块使用特征增强方法来提高预测的准确性, 并且进一步地针对中心点预测而进行了损失函数的修改。实验结果表明, 改进后的算法相比于 FPCC 算法在 Precision 值和 Recall 值上分别提高了 10%、15%。

关键词: 工业抓取; 三维点云; 深度学习; 实例分割; 人工智能; 卷积神经网络

引用格式: 冯兴盛, 刘涌, 唐磊, 刘文兴. 基于改进 FPCC 的实例分割算法. 计算机系统应用, 2024, 33(1): 192-198. <http://www.c-s-a.org.cn/1003-3254/9367.html>

Instance Segmentation Algorithm Based on Improved FPCC

FENG Xing-Sheng, LIU Yong, TANG Lei, LIU Wen-Xing

(School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China)

Abstract: Instance segmentation of 3D point clouds is a critical preprocessing step in industrial automation. However, there are often many occlusions in industrial grasping scenarios, which makes it difficult for instance segmentation networks of 3D point clouds to distinguish between similar objects. To this end, this study proposes an improved algorithm based on FPCC. This algorithm has two branches, including a center point branch for inferring the center points of instances and an embedded feature branch for describing point features. The segmentation results are obtained by clustering algorithms. The feature enhancement (FEH) module plays a crucial role in improving the accuracy of center point prediction. This module employs FEH methods to improve the prediction accuracy and further modifies the loss function for center point prediction. Experimental results show that compared with the FPCC algorithm, the improved algorithm increases the Precision and Recall values by 10% and 15% respectively.

Key words: bin-picking; 3D point cloud; deep learning; instance segmentation; artificial intelligence; convolutional neural network (CNN)

由于 3D 扫描设备的快速发展, 获取三维点云 (3D) 数据已不再困难。三维数据的容易获取, 使得深度学习已经不必限制在二维方面的研究与应用, 深度学习已经逐渐从二维跨向三维。因此如何使用深度学习高效地处理三维点云数据, 已经成为一个盛行的话题。在各种应用中, 比如: 自动驾驶、姿势估计、机器人零件拾

取、医疗影像, 通常都需要将点云分割作为预处理步骤, 再以此为基础进行下一步骤。在机器人领域中, 机器人零件拾取仍然是人们关注的热点。在个体相互叠加、相互遮挡的场景下, 信息混杂, 如何达到很好的分割效果是一大问题。

深度学习在计算机视觉方面的 4 个经典应用: 识

① 收稿时间: 2023-07-20; 修改时间: 2023-08-21; 采用时间: 2023-08-24; csa 在线出版时间: 2023-11-24

CNKI 网络首发时间: 2023-11-28

别、检测和定位、语义分割、实例分割.实例分割是实例分割为属于同一类的对象的单独实例提供了不同的标签.实例分割不光有语义分割的一部分特点,需要不同类别之间进行分类,还有目标检测的一部分特点,需要定位出来不同的个体,即使这些个体属于同一类物品.实例分割大致分成两种方法:自下而上和自上而下.自上而下的实例分割方法先通过目标检测的方法找到个体所在的区域,再在这一部分区域中进行语义分割,分割出来的结果作为输出.自上而下的方法通常实例分割的结果具有较好的对象性.然而,这些方法需要多阶段的训练和删除多余的建议.因此,它们通常是耗时并且计算成本高的^[1-6].自下而上的实例分割方法通常认为实例分割是语义分割之后的后续步骤.特别是,大多数现有的方法都基于同一类别的点应该具有非常相似的特征的假设.因此,这些方法主要侧重于识别特征学习和点分组,通常先进行像素级别的语义分割,在通过聚类、度量学习等手段来区分不同的个体.这些方法处理速度较快,但相对而言并没有特别好的精度^[7-11].两者之间最大的区别就在于是否需要找到个体所在的区域,两种方法各有优劣,但是在存在强烈遮挡的情况下,自上而下的方法在寻找个体所在区域的效率不高,因此性能较差.

1 FPCC 网络原理

现有的拾取场景实例分割方法可分为两类:面向物流的方法和面向工业的方法.前者是多类、多实例

学习.后者倾向于从杂乱的场景中进行一类多实例学习,而不预测语义标签.研究人员已采用主流的 2D 检测或 2D 实例分割网络来定位物流场景中的对象.而在该工业场景下会存在多个同类物体相互堆叠、相互遮挡的问题、三维的分割速度问题以及三维数据标注难度问题.因此,很少有研究专注于工业拾取场景的实例分割.而 Xu 等人^[12]设计了针对此环境的三维实例分割网络 (FPCC).

1.1 FPCC 算法主要流程

FPCC 网络^[12]是直接对点云进行处理的网络.首先,将点云数据的坐标 (x_i, y_i, z_i) 进行如式(1)转换,得到新的坐标 $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$.

$$\begin{cases} \bar{x} = x_i - \min\{x_1, x_2, \dots, x_n\} \\ \bar{y} = y_i - \min\{y_1, y_2, \dots, y_n\} \\ \bar{z} = z_i - \min\{z_1, z_2, \dots, z_n\} \end{cases} \quad (1)$$

FPCC 网络输入的点云数据的格式为 6 维,即 $p_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, n_x, n_y, n_z)$, 其中 (n_x, n_y, n_z) 是每个点云的标准法向量.6 维点云数据 $(N \times 6)$ 输入进特征提取器 (DGCNN 网络^[13]), 得到 256 维的特征向量 $(N \times 256)$.如图 1 所示, FPCC 网络主要有两个分支:中心点预测分支和嵌入式特征分支.点云数据在经过 DGCNN 以及 MLP 后, 得到 256 维的特征向量, 再通过 MLP 将该特征映射为维度相等的两部分, 分别作为中心点分支以及嵌入式特征分支的输入. FPCC 网络采用算法找到每一个物体最可能是物体中心的点, 最终结合点特征根据聚类算法分别将其他点聚类到中心点中, 从而实现同一类别的不同个体的分割.

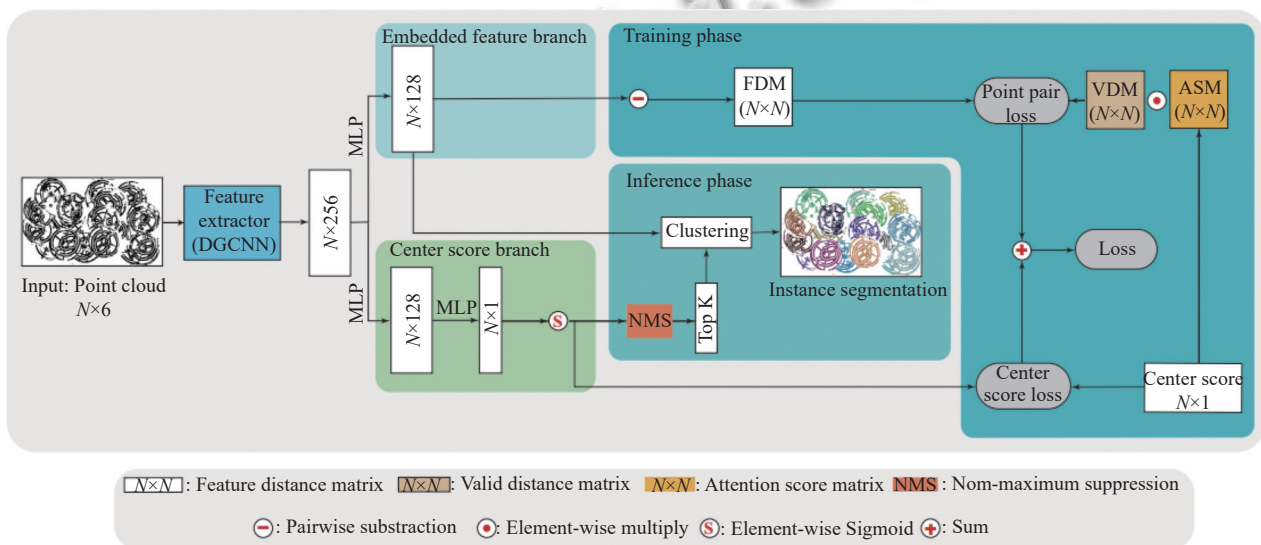


图 1 FPCC 网络框架

1.2 中心点分支

该分支是通过训练得到不同个体的中心点, 该分支得到的中心点的准确度, 直接影响到后续实例分割的结果. 因此, 该分支训练的好坏是至关重要的.

中心点分数是指点与其物体中心点之间的距离. 物体中心的点的得分会比物体边界的点的得分高. 我们采用如式(2)的中心点分数定义:

$$C_{center(i)} = 1 - \left(\frac{\|p_i - c_i\|_2}{d_{\max}} \right)^\beta \quad (2)$$

其中, β , d_{\max} 是常数, FPCC 中已经证明了 $\beta = 2$ 时, 使中心点分数在 0-1 区间的得分更加均匀, 故取 $\beta = 2$, d_{\max} 则是根据零件的大小进行调整. c_i 是 p_i 所属物体的几何中心坐标. $C_{center(i)}$ 的取值在 [0, 1] 区间. 由以上公式可以看出, 离 c_i 越近, 则 $C_{center(i)}$ 越接近于 1, 离 c_i 越远, 则 $C_{center(i)}$ 越接近于 0. 在 FPCC 中证明了边界区域上的点大多得分为 0, 靠近中心的点大约得分为 1, 这更加符合了本文的改进方法.

1.3 嵌入式特征分支

1.3.1 特征矩阵

在学习到的特征空间中, 同一实例的特征点在特征空间中应该非常接近, 而不同实例的特征点应该会被区分开, 因此我们采用距离公式来表示点与点之间的特征距离, 即:

$$d(a, b) = \|a - b\|_2 \quad (3)$$

其中, $d(a, b)$ 表示 a 和 b 之间的欧几里得距离. 而特征矩阵 $D_F \in R^{N \times N}$ 中第 (i, j) 个元素表示为:

$$d_{F(i, j)} = \|e_F^{(i)} - e_F^{(j)}\|_2 \quad (4)$$

1.3.2 有效特征矩阵

有效特征矩阵 $D_V \in R^{N \times N}$ 是一个二进制矩阵, 其中每个元素是 0 或 1. 该矩阵的主要作用是判断特定的欧几里得距离内的点是否是属于同一个实例. 在训练阶段, 同时根据点的特征距离和欧几里得距离对点进行聚类, 如果两点的欧几里得距离超过了最大距离 d_{\max} 的两倍, 那断定这两点不属于同一实例, 这样会直接忽略距离太远的点, 太远的点也就不会关注损失. 而特征矩阵 $D_V \in R^{N \times N}$ 中第 (i, j) 个元素表示为:

$$d_{V(i, j)} = \begin{cases} 1, & \text{if } \|p_i - p_j\|_2 < 2d_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

式(5)表示判断 p_i, p_j 两点之间的欧几里得距离是否在合理范围内.

1.3.3 注意力权重矩阵

中心点被用作聚类的参考点, 所以更靠近中心位置的点对应该具有更高的权重. 因此引入注意力权重矩阵 $S_A \in R^{N \times N}$ 来增加重要点的权重. 而特征矩阵 $S_A \in R^{N \times N}$ 中第 (i, j) 个元素表示为:

$$S_{A(i, j)} = \min(1, S_{center(i)} + S_{center(j)}) \quad (6)$$

1.3.4 嵌入式特征损失

一对点 (p_i, p_j) 可能有以下两种关系: (1) (p_i, p_j) 属于同一个实例. (2) (p_i, p_j) 不属于同一个实例. 因此将嵌入式特征损失 L_{EF} 定义为:

$$L_{EF} = \sum_i \sum_j w_{(i, j)} k_{(i, j)} \quad (7)$$

其中, $w_{(i, j)}$ 是 $W \in R^{N \times N}$ 中的第 (i, j) 个元素, 而 W 则是 D_V 和 S_A 的逐元素乘积, 即:

$$w_{(i, j)} = d_{V(i, j)} \times S_{A(i, j)} \quad (8)$$

而 L_{EF} 中 $k_{(i, j)}$ 是点与点之间的损失, 表示为:

$$k_{(i, j)} = \begin{cases} \max(0, d_{F(i, j)} - \varepsilon_1), & \text{if } p_i \text{ and } p_j \text{ 是一个实例} \\ \max(0, \varepsilon_2 - d_{F(i, j)}), & \text{otherwise} \end{cases} \quad (9)$$

其中, ε_1 、 ε_2 是常数, 并设置为满足条件 $0 < \varepsilon_1 < \varepsilon_2$, 因为不同实例中的点对的特征距离应大于属于同一实例的点对^[8]. 我们不需要使相同实例中的点对的特征距离接近零, 但小于阈值 ε_1 , 这有助于学习^[14].

2 FPCC 算法改进

对 FPCC 网络算法网络进行阅读理解以及实验验证, 发现模型中对于中心点不在物体本身的物体, 会存在中心点预测不准的问题, 这一问题也在该论文中提到过. 为了提升模型对于数据集的分割效果, 本文对 FPCC 算法改进主要为以下几个方面.

(1) 改变了原网络模型结构. 对于中心点分支进行改进, 本文提出了新的中心点预测框架 FEH, 该框架可以增强中心点预测的准确性. 该框架通过对中心点的预测, 再将预测结果放入网络与其余特征一起学习, 加强了对中心点这一特征的关注, 从而提高预测的中心点的正确率.

(2) 改进了 FPCC 网络的损失函数. 针对中心点预

测不准的问题,要使得网络对中心点更加关注,因此要修改网络的学习目标(损失函数),修改了网络的损失函数,添加了更多与中心点相关的值,加强对中心点的监控,从而提高中心点预测的准确度.

2.1 网络结构改进

FEH网络是本文设计用于中心点的预测,如图2所示,该网络的输入为经过多层感知机(MLP)得到的128维特征向量($N \times 128$)和点云数据($N \times 6$),将128维特征向量($N \times 128$)通过多层感知机(MLP)得到预测的

中心点得分($N \times 1$),再将该预测的中心点得分分别加入正在学习的特征空间中,使得中心点在特征空间与其余点分离,网络更容易学习中心点的特征.其中,本文设计的特征加强方法是FEH网络的重点.而训练过程中的特征学习是使用DGCNN网络中的EdgeConv模块,该模块因为有 k -nn图表,使得点云不仅仅有坐标点信息,还将该点周围最近的 k 个点与该点联系到了一起,从而加强了点与点之间的联系,突显了局部信息,达到更好的学习效果.

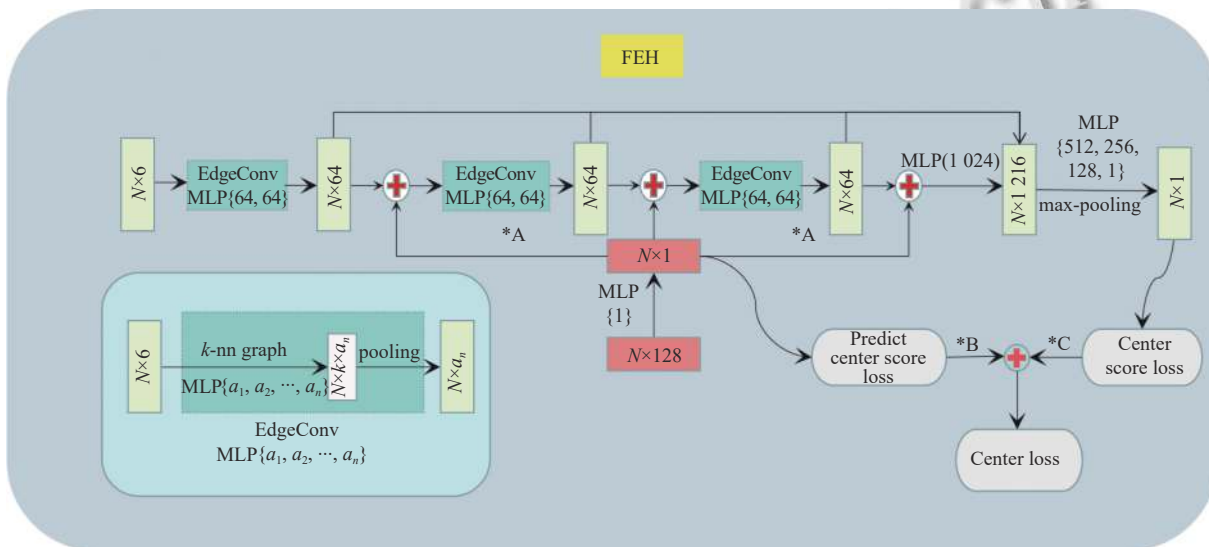


图2 FEH网络框架

特征加强方法是如图3所示,该方法是将得到的预测的中心点置信度通过加权的方法按照一定比例加入到点特征中,从而突显中心点的特征,达到更好的学习效果.

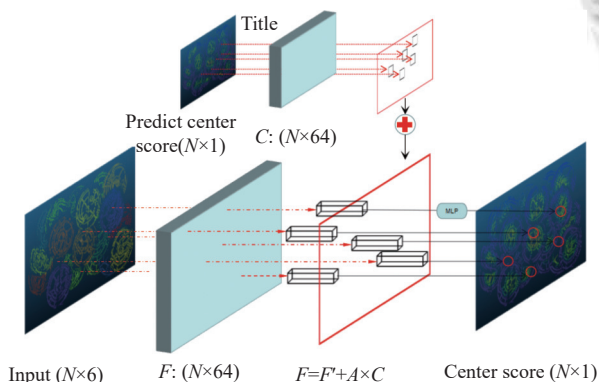


图3 中心点特征增强方法

设点云数据为 $P = \{p_0, p_1, \dots, p_n\}$,点云个数为 N ,第 i 个点 $p_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, n_x, n_y, n_z)$.中心点置信度为 $C = \{c_0, c_1, \dots, c_n\}$,第 i 个点的置信度为 $c_i (0 \leq c_i \leq 1)$.设特征向量为

$F = \{f_0, f_1, \dots, f_n\}$,第 i 个点的特征向量为 $f_i(1 \times 64)$.

中心点置信度是指该点是中心点的概率大小,假设中心点置信度是两个极端,只存在很大和很小的两种可能性,即 $C = \{C_s, C_b\}$.集合 C_s 是中心点置信度很大的点集合, $C_s = \{c_{m0}, c_{m1}, \dots, c_{ms}\} (0.5 \leq c_{mi} \leq 1.0)$,集合 C_b 是中心点置信度不是很大的点集合, $C_b = \{c_{n0}, c_{n1}, \dots, c_{nb}\} (0 \leq c_{ni} \leq 0.5)$.

新的特征向量如式(10)得出 $F' = F + A \times C$,其中 A 是一个常数, $F' = \{f'_0, f'_1, \dots, f'_n\}$.

$$\begin{cases} f'_0 = f_0 + A \times c_{m0} \\ f'_1 = f_1 + A \times c_{n0} \\ \vdots \\ f'_{n-1} = f_{n-1} + A \times c_{ms} \\ f'_n = f_n + A \times c_{nb} \end{cases} \quad (10)$$

经过特征加强之后, C_s 中的点会在原有的特征空间中变动大一点,而 C_b 中的点,在原有的特征空间中会

有稍微变动,相比之下,中心点和不是中心点在特征空间中会区分开,从而达到容易学习的目的。

2.2 中心点损失函数

由于 $L1$ 损失函数的鲁棒性,平滑 $L1$ 损失如式 (11) 所示被用作中心得分分支的损失函数^[15]。

$$L1(x) = \begin{cases} 0.5|x|^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (11)$$

预测的中心点得分损失 $L_{\text{pre_center}}$ 、中心点得分损失 L_{center} 和中心损失 L_{CL} 如式 (12) 所示:

$$\begin{cases} L_{\text{pre_center}} = \frac{1}{N} \sum_i^N L1(S_{\text{pre_center}(i)} - \hat{S}_{\text{pre_center}(i)}) \\ L_{\text{center}} = \frac{1}{N} \sum_j^N L1(S_{\text{center}(j)} - \hat{S}_{\text{center}(j)}) \\ L_{\text{CL}} = B \times L_{\text{pre_center}} + C \times L_{\text{center}} \end{cases} \quad (12)$$

其中, $S_{\text{pre_center}(i)}$ 是预测的中心点得分, $S_{\text{center}(i)}$ 为中心点得分。 B 和 C 是每个损失的比例大小,表示该损失在总损失中的重要性。较小的 L_{CL} 意味着预测的中心点更接近实际的中心点。

3 实验分析

3.1 实验数据集

本文实验所采用的数据集是开源的数据集 IPA 数据集^[16]。IPA 数据集:这是第一个用于 6D 物体姿态估计和实例分割的公共数据集,用于箱子拾取,其中包含足够的注释数据,可用于基于学习 (learning-based) 的方法。数据集由合成场景和真实场景组成。合成场景和真实场景中提供了深度图像、3D 点云、每个对象的 6D 姿势注释、可见性分数和每个对象的分割遮罩。数据集包含 10 个不同的对象。所有对象的训练场景都是合成的,只有 Ring-Screw 和 Gear-Shaft 的测试场景才是真实的数据。

3.2 实验参数设置

本文使用 TensorFlow 1.1 来构建我们的网络模型 RFCC,并使用 Adam^[17] 优化器进行培训,初始学习率为 0.0001,批量大小为 2,动量为 0.9。所有训练和验证都在 NVIDIA GeForce RTX 3080 GPU 上进行。在训练阶段,设置 $\epsilon_1 = 0.5$, $\epsilon_2 = 1$, $\alpha = 3$, $\beta = 2$, $A = 0.3$, $B = 3$ 和 $C = 6$ 。其中, ϵ_1 , ϵ_2 , α , β 等参数是参考 FPCC 网络的参数设置,在训练过程中,为了避免参数不同而导致的比较误差,因此采用的是与 FPCC 网络相同的参数设置。

而 A , B 和 C 的值是通过进行大量消融实验而得到的参数设置。在训练过程中的每个批次中,输入点 ($N = 4096$) 从每个场景中随机采样,每个点只能采样一次。每个点都转换为 6D 矢量 ($\bar{x}_i, \bar{y}_i, \bar{z}_i, n_x, n_y, n_z$) 用于输入 RFCC 网络。重复采样,直到场景的剩余点小于 N 。训练 RFCC 模型需要的时间大约需要 15 h。

3.3 实验结果及分析

3.3.1 召回率、准确率和 $F1$ 分数

本文采用召回率 (recall, R) 和准确率 (precision, P) 以及两者的权重结合 $F1$ 分数来评价网络模型的性能。精确率是针对我们预测结果而言的,它表示的是预测为正的样本中有多少是真正的正样本。那么预测为正就有两种可能了,一种就是把正类预测为正类 (TP) 另一种把负类预测为正类 (FP),也就是 $P = TP / (TP + FP)$ 。召回率是针对我们原来的样本而言的,它表示的是样本中的正例有多少被预测正确了。那也有两种可能,一种是把原来的正类预测成正类 (TP),另一种就是把原来的正类预测为负类 (FN),也就是 $R = TP / (TP + FN)$ 。这两个衡量标准往往存在冲突;当精度较高时,召回率往往较低,反之亦然。为了平衡 precision 和 recall 之间的权衡,通常使用 F 测度,即 precision 与 recall 的加权谐波均值,定义为 $F1 = 2 \times P \times R / (P + R)$ 。表 1 是实例分割在 IPA 数据集上的数据结果。在圆环这种中心点不在物体本身上的物体 (Ring-Screw),本文的改进方法很好地解决了这一问题,并在精度上提升了 12%,并且本文方法并没有降低其他物体分割的效果。

表 1 实例分割的对比结果 (%)

方法	Ring-Screw			Gear-Shaft		
	P	R	$F1$	P	R	$F1$
ASIS ^[18]	15.51	11.56	13.26	20.14	9.46	9.46
3D-BoNet ^[1]	27.88	19.8	23.16	26.57	20.11	22.89
Point Group ^[19]	52.4	41.22	46.14	58.79	36.8	45.27
FPCC ^[12]	58.43	48.74	53.15	54.29	69.53	60.97
改进FPCC	70.4	63.09	66.54	56.74	64.63	60.43

3.3.2 消融实验

表 2 给出了通过使用参数 A 的不同值进行特征加权而获得的分割结果。当 A 的值设置得太大时,中心点特征变得过于突出,这会阻碍其他特征的学习,并导致较差的分割性能。相反,当 A 的值设置得太小时,中心点特征的突出效果不够明显,这会使网络难以学习中心点特征,并导致不好的学习效果。因此,选择合适的 A 值对于获得良好的分割性能是重要的。

表2 A值的消融实验

Data	Ring-Screw (%)		Gear-Shaft (%)	
	Precision	Recall	Precision	Recall
0.1	67.9	61.76	57.67	67.18
0.3	70.4	63.09	56.74	64.63
0.5	70.16	63.34	54.33	62.92

参数 B 和 C 表示分配给两个中心点损失函数的权重, 它们的相对比例会显著影响网络的学习性能. 为了评估这些参数的有效性, 本文进行了4次消融实验, 如表3所示. 结果表明, 为这两种损失设置适当的权重对于改进网络的分割性能至关重要. 因此, 可以根据具体应用场景调整 B 和 C 值, 以优化网络的分割性能.

表3 B值的消融实验 ($A=0.3$)(%)

Data	Ring-Screw		Gear-Shaft	
	Precision	Recall	Precision	Recall
$B:C$				
1:1	64.22	57.54	52.3	61.16
1:2	70.4	63.09	56.74	64.63
1:5	62.32	56.08	51.39	59.93

3.3.3 检测结果可视化

从检测结果可以清晰地看到, 改进后的算法较改进前的分割效果更好. 如图4所示, 可以清晰地看出, 改进后的中心点的预测更加集中. 从可视化可以看出, 本文提出的中心点增强的方法是有作用的.

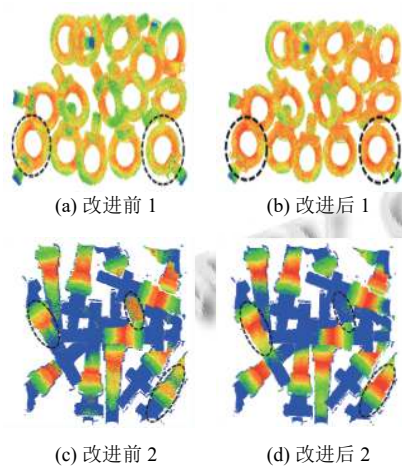


图4 改进后中心点效果对比图

对于中心点不在物体上的物体, 如: Ring-Screw, 该物体的预测一直都在实际的中心点上, 因此会降低分割的效果, FPCC 网络不适合中心点不在物体本身上的对象. 但是, 如图5所示, 改进后的网络相比于原网络, 分割效果更好.

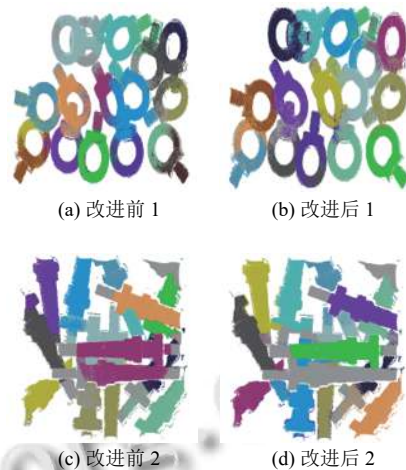


图5 改进后分割效果对比图

4 结论与展望

本文根据 FPCC 网络改进出一个新的网络, 该网络增加了一个新的中心点预测模块, 可以更加准确地预测中心点. 本文也继承了 FPCC 网络的便捷性, 可以直接对点云数据进行处理. 在单一物体并且有大量遮挡以及堆叠的场景下, 本文取得了较好的实例分割结果.

本文还有一定的局限性, 会在之后的工作中尝试解决. (1) 因为增加了 FEH, 该网络的参数会增加, 导致计算量也会相对增加. (2) 物体中心点预测得更加准确, 导致现在的聚类算法不能起到更好的聚类效果. 因此设计一种新的聚类算法是我们未来的工作.

参考文献

- 1 Yang B, Wang JA, Clark R, *et al.* Learning object bounding boxes for 3D instance segmentation on point clouds. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 605.
- 2 Hou J, Dai A, Nießner M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4416–4425.
- 3 Yi L, Zhao W, Wang H, *et al.* GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 3942–3951.
- 4 Narita G, Seno T, Ishikawa T, *et al.* PanopticFusion: Online volumetric semantic mapping at the level of stuff and things.

- Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems. Macao: IEEE, 2019. 4205–4212.
- 5 Zhang FH, Guan CY, Fang J, *et al.* Instance segmentation of LiDAR point clouds. Proceedings of the 2020 IEEE International Conference on Robotics and Automation. Paris: IEEE, 2020. 9448–9455.
- 6 Engelmann F, Bokeloh M, Fathi A, *et al.* 3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 9028–9037.
- 7 Wang WY, Yu R, Huang QG, *et al.* SGPN: Similarity group proposal network for 3D point cloud instance segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 2569–2578. [doi: [10.1109/cvpr.2018.00272](https://doi.org/10.1109/cvpr.2018.00272)]
- 8 Charles RQ, Su H, Kaichun M, *et al.* PointNet: Deep learning on point sets for 3D classification and segmentation. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 77–85. [doi: [10.1109/cvpr.2017.16](https://doi.org/10.1109/cvpr.2017.16)]
- 9 Liu C, Furukawa Y. MASC: Multi-scale affinity with sparse convolution for 3D instance segmentation. arXiv:1902.04478, 2019.
- 10 Mo KC, Zhu SL, Chang AX, *et al.* PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 909–918.
- 11 Liang ZD, Yang M, Wang CX. 3D graph embedding learning with a structure-aware loss function for point cloud semantic instance segmentation. arXiv:1902.05247, 2019.
- 12 Xu YJ, Arai S, Liu DY, *et al.* FPCC: Fast point cloud clustering-based instance segmentation for industrial bin-picking. Neurocomputing, 2022, 494: 255–268. [doi: [10.1016/j.neucom.2022.04.023](https://doi.org/10.1016/j.neucom.2022.04.023)]
- 13 Wang Y, Sun YB, Liu ZW, *et al.* Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics, 2019, 38(5): 146. [doi: [10.1145/3326362](https://doi.org/10.1145/3326362)]
- 14 Li RK, Zhang YM, Niu DM, *et al.* PointVGG: Graph convolutional network with progressive aggregating features on point clouds. Neurocomputing, 2021, 429: 187–198. [doi: [10.1016/j.neucom.2020.10.086](https://doi.org/10.1016/j.neucom.2020.10.086)]
- 15 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2015. 1440–1448. [doi: [10.1109/ICCV.2015.169](https://doi.org/10.1109/ICCV.2015.169)]
- 16 Kleeberger K, Landgraf C, Huber MF. Large-scale 6D object pose estimation dataset for industrial bin-picking. Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Macao: IEEE, 2019. 2573–2578. [doi: [10.1109/iros40897.2019.8967594](https://doi.org/10.1109/iros40897.2019.8967594)]
- 17 He KM, Gkioxari G, Dollár P, *et al.* Mask R-CNN. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 2980–2988.
- 18 Wang XL, Liu S, Shen XY, *et al.* Associatively segmenting instances and semantics in point clouds. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2019. 4091–4100. [doi: [10.1109/CVPR.2019.00422](https://doi.org/10.1109/CVPR.2019.00422)]
- 19 Jiang L, Zhao HS, Shi SS, *et al.* PointGroup: Dual-set point grouping for 3D instance segmentation. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020. 4866–4875. [doi: [10.1109/CVPR42600.2020.00492](https://doi.org/10.1109/CVPR42600.2020.00492)]

(校对责编:牛欣悦)