计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(8): 2076−2094, 2026

3Deus: 基于点云探测的复杂场景物品识别方法

邵智超[1] 邱杰凡[1] 戴光麟[1,3] 李乔宏[1] 陈翰墨[1] 肖克江[2]
[1](浙江工业大学计算机科学与技术学院 杭州 310023)
[2](华中师范大学人工智能教育学部 武汉 430079)
[3](浙江机电职业技术大学 杭州 310053)
(211123030114@zjut.edu.cn)

3Deus: Object Recognition Method Based on Point Cloud Detection in Complex Scenario

Shao Zhichao[1], Qiu Jiefan[1], Dai Guanglin[1,3], Li Qiaohong[1], Chen Hanmo[1], and Xiao Kejiang[2]
[1] (*College of Computer Science and Technology*, *Zhejiang University of Technology*, *Hangzhou* 310023)
[2] (*Faculty of Artificial Intelligence in Education*, *Central China Normal University*, *Wuhan* 430079)
[3] (*Zhejiang Polytechnic University of Mechanical and Electrical Engineering*, *Hangzhou* 310053)

Abstract With the development of artificial intelligence technology, vision-based object recognition, as an important component of the Internet of things perception layer, can be applied in various cargo sorting scenarios when combined with sorting robots. However, in complex environments where objects are disorganized, relying solely on object recognition algorithms may not be sufficient to distinguish target objects from interfering ones, as the difference between them may be the spatial distance. This can negatively impact the robot's normal grasping actions. To address this issue, we propose a lightweight object recognition method for complex environments—3Deus. This method combines deep learning with 3D point cloud data processing. It uses a depth camera to synchronously capture both RGB images and 3D point cloud data in the recognition scene. The method eliminates the plane where interfering objects are located using the 3D point cloud, and distinguishes the plane containing the target object from the stacked interfering objects with an improved RANSAC algorithm. It also employs a plane expansion mechanism to ensure the integrity of the plane point cloud where the target object is located. Finally, the object recognition algorithm is used to obtain the recognition bounding box of the target object and construct its mapping relationship with the specific point cloud cluster. We deploy 3Deus on edge computing devices and validate its effectiveness and robustness by constructing different experimental scenarios. Experimental results show that compared with the traditional random sampling consensus algorithm, 3Deus can effectively distinguish the target objects from interference objects, achieving an accuracy of over 95%. Moreover, by utilizing the plane expansion mechanism, the recognition accuracy of the target object is improved by 30.48%.

收稿日期: 2025-09-21; **修回日期**: 2026-04-24
基金项目: 国家自然科学基金项目(62173158, 62377023); 浙江工业大学校企联合项目(KYY-HX-20240328); 湖北省自然科学基金联合基金重点项目(2025AFD195); 中央高校基本科研业务费专项资金(XJ2026001901, XJ2026001501); 数字教育湖北省重点实验室开放研究基金(F2026K03)
This work was supported by the National Natural Science Foundation of China (62173158, 62377023), the ZJUT School-enterprise Joint Project (KYY-HX-20240328), the Key Program of the Joint Fund of Hubei Provincial Natural Science Foundation (2025AFD195), the Fundamental Research Funds for the Central Universities (XJ2026001901, XJ2026001501), and the Open Fund of Hubei Key Laboratory of Digital Education (F2026K03).
通信作者: 戴光麟(dgl@zjut.edu.cn)

Key words object recognition; 3D point cloud detection; machine vision; depth camera; plane detection

摘 要 随着人工智能技术的发展，基于视觉的物品识别作为构建物联网感知层的重要组成部分，与分拣机器人相结合后可应用在各类货物分拣的具体场景中。然而，在混乱摆放物品的复杂场景中，由于目标物品与干扰物品可能存在空间距离差，仅依赖物品识别算法并不能很好地区分目标物品和干扰物品，势必对机器人的正常抓取动作产生负面影响。为了解决该问题，提出了一种轻量化的复杂环境下的物品识别方法——3Deus。该方法将深度学习和3D点云数据处理相结合，利用3D点云对干扰物品所处的平面做整体消除，并基于改进的随机采样一致性算法区分目标物品所处平面与堆叠干扰物；运用平面扩张机制，保证目标物品所处平面点云的完整性；最后，借助物品识别算法获取目标的识别框并构建其与具体点云簇的映射关系。将3Deus部署在边缘计算设备上，并通过构建不同的实验场景验证了3Deus的有效性和鲁棒性。实验结果表明，与传统的随机采样一致性算法相比，3Deus能够有效地对目标物品和干扰物品进行区分，其准确率达到了95%以上，同时3Deus利用平面扩张机制将目标物品的识别准确率提高了30.48个百分点。

关键词 物品识别；3D点云探测；机器视觉；深度摄像头；平面检测

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202550696 **CSTR:** 32373.14.issn1000-1239.202550696

物品识别作为物联网感知层的一项重要任务，是物联网实现环境信息采集与智能处理的核心技术之一。近年来，随着基于视觉的物品识别算法的快速发展，其被广泛地运用在如违禁物品识别[1]、工业生产[2-3]、交通运输[4]等多个领域。目前，物品识别算法主要分为基于传统视觉模型与基于深度学习模型2类。基于传统视觉模型的物品识别算法主要包括Viola-Jones算法[5]、基于Gabor和LDA结合方法[6]、基于SIFT特征和RANSAC匹配算法[7]等。传统模型往往基于手工特征提取，使用场景有限。相反，深度学习模型利用自身的特征提取器，使其具有较高的鲁棒性。常用的深度学习模型包括SAM2[8]，YOLOv11[9]，DN-FPN[10]等。

然而，随着工业物联网应用场景中（需要结合空间信息的场景）对物品识别复杂度的不断增加，单一的物品识别算法也逐渐表现出了其不足之处。物品识别算法一般用于处理2D图像，而2D图像只能呈现物品在平面上的投影，因此目前的物品识别算法在2D图上很难利用空间距离信息进一步区分处于不同平面的同类物品。以分拣机器人为例，基于单一物品识别算法的分拣机器人[11-13]工作场景的要求较为严苛。如图1所示，其为典型的机械臂分拣作业场景：待分拣的各类物品被无规则地放置在料架上，目标物品识别的难度非常大。由于单一的物品识别算法无法根据场景的空间特征区分目标物品与干扰物，若干扰物堆叠在目标物品周围且出现在分拣机器人摄像头的视野范围内，则会导致物品识别算法错误地将干扰物认定为目标物品，产生错误分拣。



Fig. 1 Sorting robot sorting scenario

图1 分拣机器人分拣场景

而随着深度摄像机相关技术的成熟与普及，实时获取场景的空间信息已成为可能，深度摄像机也因此被逐渐运用在医学成像[14]、农业采摘[15]、无人驾驶[16]等领域。目前，主流的深度摄像机按技术实现可分为3类：基于结构光技术[17]（如Orbbec Gemini 2 L和Orbbec Astra 2）、基于双目视觉技术[18-19]（如Intel RealSense D455系列）以及基于飞行时间测量技术[20-21]（如微软Azure Kinect）。其中，基于结构光技术的深度摄像头需要发射专门的结构光线（如红外光线）并接收其反射光，形成主动探测。虽然结构光的强度将直接影响测量距离，且这种光线容易受到环境光线以及物品表面对结构光线吸收作用的影响，进一步降低测量精度，但是这种类型的深度摄像机具有低成本与低功耗的优势，能够提供实时且低分辨率的3D点云信息，因此通用性更强。3D点云由深度摄像机捕获的用于表达场景空间立体特征的数据以及由众多包含深度信息的点构成。利用3D点云中

的3D空间特征信息弥补单一基于2D图像的物品识别算法在工业分拣场景中的不足，是当前研究的热点。

针对以上问题，当前已有研究者提出将深度学习应用于对3D点云的处理。PointStaClu[22]采用神经网络进行特征提取，并通过稳定聚类判别优化点云聚类任务。还有一些方法采用点云语义分割[23-25]的方式，将具有相似特征的点聚类成同质化的区域。然而，3D点云往往是由上万个点组成，即使在数据预处理阶段，可以以下采样的方式降低数据量，但其规模仍然十分庞大，导致模型在训练和推理阶段造成大量的内存占用与时间消耗。因此基于深度学习的3D点云处理模型很难直接部署到资源受限的嵌入式设备上。

为此我们尝试将深度摄像机与物品识别算法相结合，提出一种基于3D点云探测的轻量化物品识别方法3Deus。3Deus的核心是利用3D点云数据构建一个精确的目标物品及其所处平面的点云模型(简称为目标点云模型)。利用串行聚类集合改进传统的随机样本一致性(random sample consensus, RANSAC)算法，实现目标点云模型的粗略构建；利用平面扩张机制解决了平面点云中平面破碎、边缘损坏以及点云误删的问题，构建更为精细且完整的目标点云模型。3Deus在物品识别层面则通过3D与2D之间的映射，实现3D立体空间信息与2D像素信息之间的一致性绑定；使用通用物品识别算法YOLOv11完成对场景中所有物品的识别，并结合构建的目标点云模型，从所有的识别结果中去除干扰物品，实现对目标物品的充分识别。

由于3Deus在点云探测过程中使用了低复杂度的聚类算法与其他算法实现3D空间特征的提取，且在物品识别算法上可以自由选择算法模型的规模，因此相较于点云深度学习的方法，3Deus更为轻量化，不需要桌面级高性能显卡的支持，硬件成本更低。在实验中，我们也将3Deus分别部署在嵌入式边缘设备Jetson Nano和Jetson NX上，验证了算法实时性，实验结果表明，其更适合物联网边缘设备。

本文的主要贡献包括4个方面：

1) 提出一种基于串行聚类集合改进的RANSAC算法，以解决传统的RANSAC算法在拟合平面过程中无法区分目标物品所在的平面与堆叠物的问题。

2) 为了解决在干扰点云去除过程中出现的目标物品所处平面点云误删的情况，本文提出了平面扩张机制。在删除干扰点云的过程中，进一步判断被删点云所处空间位置，以保留原本属于目标物品所处平面的点云，从而提高平面的完整性。

3) 提出了一种相较于点云深度学习更加轻量化的点云探测与分析方法，具有更低的计算量、更低的硬件成本以及更高的可嵌入性。

4) 为了验证3Deus在区分目标物品与干扰物上的准确性以及在嵌入式设备上的计算效率，我们设计了多种实验场景，分别记录3Deus在不同场景下的测试结果，并且在Jetson Nano与Jetson NX上分别测试算法实时性，3Deus处理时间相比于其他算法，达到了SOTA水平。

1 相关工作

1.1 面向3D点云的深度学习

随着深度学习的发展，越来越多的研究者将深度学习运用在3D点云处理[26-28]之中。Zeng等人[29]提出GNN-PRFNet，针对现有点云配准拼接未端到端且忽略局部特征的问题，构建含特征提取、配准及拼接融合模块的端到端网络，解决了传统方法局部特征利用不足的问题。Eggen等人[30]提出AdaptNet分类网络和加权激活池化(WAP)模块，设计可动态调整特征权重的池化操作，显著降低了3D点云特征聚合的计算复杂度。Zheng等人[31]提出MATNet网络，针对Transformer忽略点云局部区域及多尺度特征的局限，设计自适应融合多尺度邻域特征的序贯网络，改善了点云拓扑稀疏中多特征聚合不足的问题。Li等人[32]提出深度层次化学习(DHL)，针对3D数据缺乏层次化语义解析的问题，构建含层次正则化项、嵌入融合模块的通用框架，结合视觉语言模型自动构建类层次结构，解决了3D任务的语义层次化表征难题。Wu等人[33]针对SNN在3D点云处理的高功耗局限，设计空间感知脉冲神经元(关联点位置与脉冲生成)及3D残差块，实现了在低计算成本之下的高分类精度。

虽然以上神经网络均能实现理想的点云处理效果，但它们需要经过多阶段的训练以及庞大的数据集支撑。且以上神经网络大多采用相邻搜索机制来获取点云间的相邻信息，而这些算法会占用大量的内存资源。因此可以得出，点云深度学习的方式存在一个较为明显的问题，即训练和推理时会消耗大量的时间和计算资源，不适合普及应用，特别是在计算能力受限的边缘设备中。

1.2 传统目标物所处平面识别方法

平面拟合是平面识别中的一个重要处理步骤。

为了提高点云平面拟合的精度，已有研究者们提出了多种方法。Holz 等人[34]以深度图为基础，通过邻域法计算平面法向量以及平面距离空间坐标原点的距离，进而使用聚类算法得到平面。Poppinga 等人[35]在平面拟合中，应用协方差矩阵与均方误差（mean square error, MSE）有效提高了计算效率。Jin 等人[36]提出了一种深度平面检测算法，该算法由 2 部分组成：基于区域增长平面检测和 2 阶段细化。在第 1 阶段，通过滑窗法与线性最小二乘法，在深度图中拟合出所有平面，其次计算每个平面的平面度，并按照平面度大小进行排序。在第 2 阶段，算法根据临近点到平面的距离进行区域面积增长，从而得到最终的平面。EI-Sayed 等人[37]首先利用八叉树（oct-tree）将点云划分为多个包含有限点的小立方体，其次根据点云局部密度对各个立方体进行下采样，最终借助主成分分析（PCA）的方式在点云集合中寻找平面。以上算法均能够实现一定的平面拟合效果，但在复杂场景下其鲁棒性较低，难以区分干扰物所处平面与目标物品所处平面，也无法准确地区分目标物品与干扰。

1.3 点云与 2D 视觉融合的物品识别相关工作

工业场景中，单一 2D 视觉无法利用空间信息，单一 3D 点云缺乏纹理特征，因此 RGB-D 融合识别成为研究热点。但现有方法仍存在“干扰平面区分差、边缘部署难”的问题。基于 Faster R-CNN 的改进框架[38]通过将点云深度特征融入 2D 卷积层，采用融合深度信息的改进 RoI Pooling 模块提升目标定位精度，在分拣场景中识别准确率较高，但模型参数量大，无法在边缘设备高效部署。YOLOv8 的 RGB-D 改进版本[39]通过对称双分支主干网络提取特征，引入多模态特征融合模块实现 RGB 与深度特征在通道和空间位置的融合，推理时间相对低，但未设计专门的干扰平面去除机制，当地面存在同类物品时，误识别率较高，无法满足分拣机器人的精准抓取需求。ABB 机器人的 PickMaster5 系统[40]结合 RGB-D 数据与传统模板匹配技术，虽降低了对复杂模型的依赖，但该系统主要适用于规则摆放物品的 palletizing 场景，在随机堆叠场景中识别率显著下降。

综合上述相关工作的局限性，现有方法无法同时满足轻量化部署、复杂场景抗干扰、平面完整性保障三大核心需求。点云深度学习方法存在计算复杂度高、难部署边缘设备的问题，3Deus 通过摒弃复杂神经网络，采用“RANSAC 改进+轻量级聚类”，将计算量大大降低；传统 RANSAC 无法区分堆叠干扰且平面易破碎，3Deus 通过串行聚类集合区分堆叠物，并利用平面扩张机制修复破碎平面；RGB-D 融合方法存在干扰平面区分差、误识别率高的缺陷，3Deus 先通过 3D 点云去除地面/堆叠干扰平面，再与 2D 识别框映射，误识别率大大降低；针对边缘部署中推理时间长、资源占用高的问题，3Deus 整体算法无大规模参数，满足实时需求。

2 3Deus 的设计与实现

本节将着重介绍 3Deus 中点云探测部分，该部分主要由干扰物品所处平面点云去除算法、基于聚类集合改进的 RANSAC 算法、基于平面扩张机制的改进识别算法组成。该部分的主要目的是去除干扰物所处平面与目标物品所处平面（以下简称为目标台）旁的堆叠物，实现目标台以及目标物品的点云分割。RANSAC 平面拟合是点云探测的核心，本节也将分析 RANSAC 平面拟合的执行流程与不足之处，并给出一套解决方案。

2.1 干扰物品所处平面点云去除算法

一般场景的 3D 点云会包含多个平面，传统的平面拟合算法并不能区分期望的平面和干扰的平面。另外，干扰物可能会位于多个平台，但一般情况下干扰物都位于地面上。在本文中，我们只讨论干扰物处于地面以及在目标台旁堆叠的情况。

以图 2 为例，由可视化的 3D 点云图观察出，场景中包含了多个平面，如目标台平面、桌腿平面以及地面，其中，桌腿因其点云面积相对较小，在该场景中可以忽略。地面的点云面积最大，因而对 RANSAC 平面拟合效果的影响也是最大的。然而，干扰物品所处平面（即地面）与目标台在 3D 空间中是平行的，这就意味着在 3D 空间平面方程 $Ax+By+Cz+D=0$ 中，干扰物所处平面与目标台的平面方程在前 3 个系数



Fig. 2 Initial point cloud

图 2 初始点云

上是相同的。另外,目标台点云与干扰物品所处平面点云的数量远远大于3D点云数据中的任意一个平面点云的数量,因此RANSAC算法会优先拟合两者。

根据以上分析,我们首先采用RANSAC算法拟合3D点云中最大平面。在此过程中,我们不需要关注RANSAC算法拟合到了干扰物品所处平面还是目标台,只需关注其获取的平面方程。紧接着,我们提取其平面方程中的A, B, C这3个系数,并设置系数D为0,以此来另外生成1个虚拟平面。假若点云数量为N,由于空间中1个点可以用x, y, z坐标来描述,因此点云集合可以用1个矩阵来存储,这个矩阵大小为N×3,用符号$\boldsymbol{CP}$表示。记平面系数矩阵$\boldsymbol{PC}$=(A, B, C),计算每个点距离虚拟平面的投影长度:

$$\boldsymbol{D}_i = \boldsymbol{CP} \times \boldsymbol{PC}^{\mathrm{T}}, \tag{1}$$

$\boldsymbol{D}_i$表示点到平面的投影,大小为N×1。在3D空间中,投影长度越小,则说明该点在空间中的位置就越靠近底部。因此我们可采用硬阈值的方式实现底端点云去除。$Prec_{\mathrm{d}}$是根据实际场景确定的阈值系数,$thres_{\mathrm{d}}$是计算得出的硬距离阈值。

$$thres_{\mathrm{d}} = (\max \boldsymbol{D}_i - \min \boldsymbol{D}_i) \cdot Prec_{\mathrm{d}} + \min \boldsymbol{D}_i。 \tag{2}$$

最后将大于$thres_{\mathrm{d}}$的点云保留下来,将小于$thres_{\mathrm{d}}$的点云去除,即可将位于目标台下方的干扰物所处平面点云去除。本节算法的流程见算法1,其中省略了RANSAC算法的实现步骤。

算法1. 干扰物品所处平面点云去除算法。

输入:3D点云矩阵$\boldsymbol{CP}$,迭代次数N,去除点云的百分比$Prec_{\mathrm{d}}$;

输出:剩余点云矩阵$\boldsymbol{CP}$。

① A, B, C, D=*RANSAC*($\boldsymbol{CP}$);

② $\boldsymbol{D}_i$=*Cal_DistanceAll*($A, B, C, \boldsymbol{CP}$); /*计算所有点云距虚拟平面的距离*/

③ $Max\boldsymbol{D}_i = \max(\boldsymbol{D}_i)$;

④ $Min\boldsymbol{D}_i = \min(\boldsymbol{D}_i)$;

⑤ 根据式(2)和$Prec_{\mathrm{d}}$计算阈值$thres_{\mathrm{d}}$;

⑥ $\boldsymbol{CP}$ = *Remove*($\boldsymbol{CP}, \boldsymbol{D}_i, thres_{\mathrm{d}}$); /*从$\boldsymbol{CP}$中删除小于$thres_{\mathrm{d}}$的点*/

⑦ return $\boldsymbol{CP}$。

2.2 基于串行聚类集合的改进RANSAC算法

目前,RANSAC算法在拟合平面中具有较高的准确率,但在复杂环境下仍具有不足之处。根据RANSAC算法的执行流程可知,RANSAC算法在随机选取空间中的3个点后即可计算得到一个平面的方程。但在判断其他点是否位于这个平面的依据则是点到平面的距离,若遇到干扰物与目标台高度平齐的情况,则会使得RANSAC算法无法区分干扰物与目标台,因此RANSAC算法在本质上是计算空间中包含点云数量最多的一个扁平空间。

以图3为例,平面A为传统RANSAC算法拟合出来的桌面平面。理想情况下仅需要获取平面A,但由于干扰物堆叠过高且与平面A相平齐,导致RANSAC算法错误地认为堆叠物与平面A相平齐的那部分点云(记为平面B)也属于平面的一部分。然而实际上,平面A与平面B并不属于同一平面,2个平面近乎垂直。另外,由于受到目标台上物品的遮挡,拟合得到的平面必然是破碎的,在点云图中表现为数个孤立的点云簇。因此如何判断孤立的点云簇属于目标台还是干扰物是本节重点解决的问题。

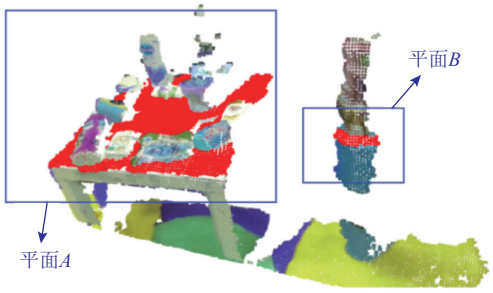


Fig. 3 Incorrect plane fitting

图3 错误的平面拟合

我们提出一种利用串行聚类集合算法对RANSAC算法进行改进。其中,聚类集合包括了基于密度的聚类DBSCAN(density-based spatial clustering of applications with noise)、K-维树(K-dimensional tree, KD-Tree)、K-均值(K-Means)三种聚类算法。将以上3种聚类算法按照一定的顺序执行,对传统RANSAC算法进行改进,实现干扰点云的滤除与目标点云模型的粗略构建。

首先,利用RANSAC算法拟合剩余点云中的目标台平面,得到平面方程$A_{\mathrm{T}}x + B_{\mathrm{T}}y + C_{\mathrm{T}}z + D_{\mathrm{T}} = 0$记平面系数$\boldsymbol{PA}$=($A_{\mathrm{T}}$, B_{T}, C_{T})。其次,借助DBSCAN密度聚类,将拟合出来的平面按照密度划分为多个点云簇,并分别标记。利用KD-Tree计算每一个点云簇内的平均法向量。在代码实现方面,使用Open3d函数库中的函数*estimate_normals*。用$\boldsymbol{PN}_{\mathrm{a}}$记录所有点云簇的平均法向量,若点云簇数量为$M_{\mathrm{a}}$,则$\boldsymbol{PN}_{\mathrm{a}}$的大小为$M_{\mathrm{a}}$×3。计算每个点云簇的平均法向量$\boldsymbol{PN}_{\mathrm{a}}$和目标台平面法向量$\boldsymbol{PA}$之间的夹角,计算式如式(3)所示。其中|·|运算符表示对向量取模。

$$\boldsymbol{Ag} = \arccos\left(\frac{\boldsymbol{PN}_{\mathrm{a}} \times \boldsymbol{PA}^{\mathrm{T}}}{|\boldsymbol{PN}_{\mathrm{a}}| \cdot |\boldsymbol{PA}|}\right)。 \tag{3}$$

通过转换将角度限制在 [0, 90] 之间，得到一个大小为 M_a×1 的向量 ***Ag***。最后，借助 *K*-Means 对向量 ***Ag*** 进行 1D 聚类，使其分为正常与异常 2 类数据，数值较小的为正常类，较大的为异常类，删除异常点云簇后即可获得最终粗略构建的目标点云模型。本节的具体算法流程见算法 2。

算法 2. 串行聚类集合改进的 RANSAC 算法。

输入：点云矩阵 ***CP***;

输出：正常平面点云簇 ***PNor***。

① A_T, B_T, C_T, D_T, ***Plane***=*RANSAC*(***CP***); /****Plane*** 为拟合平面的点云矩阵*/

② ***clusterList***=*dbscan*(***Plane***); /****clusterList*** 表示所有的点云簇*/

③ $\boldsymbol{PN}_a$=*estimate_normals*(***clusterList***); /*计算所有点云簇的平均法向量*/

④ 根据式(3)计算 ***Ag***，并将 ***Ag*** 限制在 [0, 90] 的范围内；

⑤ ***Nor***, ***abNor***=*Kmeans*(***Ag***); /*借助 *Kmeans* 算法将 ***Ag*** 分为异常和正常角度，***Nor*** 一类为正常，***abNor*** 一类为异常*/

⑥ 将所有正常角度对应的点云簇记为 ***PNor***;

⑦ return ***PNor***。

2.3 基于平面扩张机制的改进识别算法

凸包能从庞大的点云数据中迅速找到具有一定特征的目标点云，因此凸包计算在 3D 点云处理的过程中具有非常重要的意义。在 3D 点云处理的过程中，凸包提取具有一定特征的目标点云时，能够高效确定庞大点云数据的外部边界。通过计算凸包，我们能够得到点云数据的最外层几何形状，这对于后续的目标识别、路径规划、空间优化等应用至关重要。

在 3Deus 中，2 次凸包计算分别基于 3D 点云和 2D 图像，前者能够将分离的点云簇包裹成一个统一的平面整体，后者能够根据平面整体进一步获取最外围凸包的点。

首先，针对 2.2 节中获取的目标台点云计算凸包，将所有分离的点云簇合并为一个平面整体，并记录下所有的 3D 凸包点。根据坐标转换公式将所有 3D 凸包点映射到 2D 图上。转换公式如下：

$$\boldsymbol{M}_{\text{camera}}=\begin{pmatrix} f_x & 0 & pp_x & 0 \\ 0 & f_y & pp_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{4}$$

其中 $\boldsymbol{M}_{\text{camera}}$ 表示摄像机的内部参数矩阵，f_x 表示像素宽度方向上的焦距，f_y 表示像素高度方向上的焦距。pp_x 与 pp_y 表示主点在像素坐标系下的坐标，这里主点通常表示像素的中心点。不同的深度摄像头具有不同的内部参数矩阵，其内部参数矩阵皆可从指定深度摄像头的相关说明文档中获取。

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}=\boldsymbol{M}_{\text{camera}}\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}, \tag{5}$$

$$\begin{cases} su=x_c\cdot f_x+pp_x\cdot z_c, \\ sv=y_c\cdot f_y+pp_y\cdot z_c, \\ s=z_c, \end{cases}\Rightarrow\begin{cases} u=\dfrac{x_c}{z_c}\cdot f_x+pp_x。 \\ v=\dfrac{y_c}{z_c}\cdot f_y+pp_y。 \end{cases} \tag{6}$$

式(5)与式(6)描述了如何利用内部参数矩阵将 3D 点云映射到 2D 图像中。其中，s 表示比例系数，x_c, y_c, z_c 表示某个点的 3D 空间坐标，u 和 v 则表示该点在图像上的坐标。完成 3D 点云到 2D 图像的映射之后，利用 Graham 算法计算 2D 图像的凸包，获取 2D 图像的目标台轮廓，并记录所有的 2D 凸包点。

在干扰点云去除和平面扩张机制中需要找出以上 2D 凸包点在 3D 点云中的坐标位置。但由于坐标转换过程是不可逆的，即 2D 坐标不能逆向推导出 3D 坐标，因此我们采用了坐标映射的方式，通过建立坐标映射表 *CorTable* 来保存凸包点在 3D 点云与 2D 图像上的坐标映射关系。令凸包点的数量为 *K*，则坐标映射表 *CorTable* 的大小为 *K*×5。*CorTable* 的前 3 列表示凸包点在 3D 点云中的坐标，后 2 列表示其在图像上的 2D 坐标。

根据获取的凸包点以及坐标映射表，将 2D 图像中相邻的 2 个凸包点映射回 3D 点云，并获取其 3D 坐标，再结合 2.2 节中利用传统 RANSAC 算法拟合出的目标台法向量，即可通过计算确定 3D 空间中经过指定的 2 个点且平行于一个指定法向量的平面。将点云坐标(*X, Y, Z*)代入平面方程，计算得到 $A_Ex+B_Ey+C_Ez+D_E$ 的值，并判断其与 0 之间的大小关系，即可确定对应的点是否为干扰物点云。

然而，由于物品的遮挡而导致的点云破碎性，拟合的凸包并没有完全包裹住目标台，导致算法错误地将原本位于目标台边缘的点云判断为干扰物，使得目标台边缘物品的区分精度下降。为此，3Deus 采用了一种平面扩张机制。其步骤如下：

1)输入最小点云阈值 *minPoint*，扩张系数 *exps*。

2)将开始确定的平面向外扩张一定的距离。扩张后的平面方程，除了系数 D_E 外，其他系数与原平面相同。扩张后的平面方程为 $A_Ex+B_Ey+C_Ez+D_{\text{plus}}=0$，$D_{\text{plus}}$ 表示为

$$D_{\text{plus}}=D_E\pm\frac{\sqrt{A_E^2+B_E^2+C_E^2}}{exps}。 \tag{7}$$

3)检查扩张后与扩张前 2 个平面之间的点云数量,若大于最小点云阈值 *minPoint*,则说明 2 个平面内的点属于目标台,将 2 个平面同时向外移动 1 个单位的距离,继续执行步骤 2)3)。假若小于 *minPoint*,则说明当前已经遍历到目标台以外的部分,则退出当前平面,转向下一个平面继续执行以上的算法,直至遍历完所有的平面。

平面扩张机制解决了在干扰点云去除过程中出现的目标物品所处平面点云误删的情况,在删除干扰点云的过程中,进一步判断被删点云所处空间位置以保留原本属于目标物品所处平面的点云,从而提高平面的完整性。

2.4 点云探测算法执行实现

本节将按照图 4 所示的算法执行流程,拆解各个步骤并展示其计算结果,以直观且详细地描述 3Deus 中点云探测的执行逻辑。

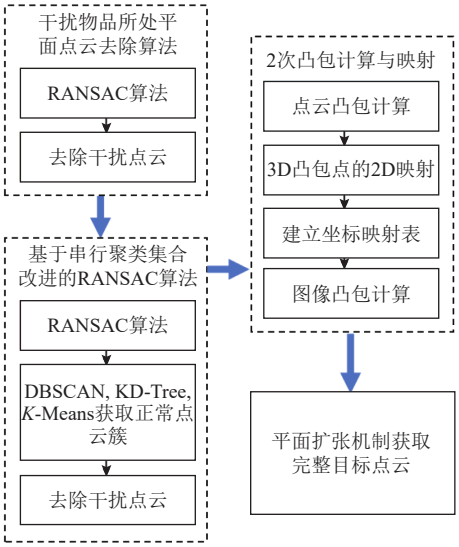


Fig. 4 Overall flowchart of system

图 4 系统整体流程图

通过深度相机获取场景的初始点云与 RGB 图,如图 5 所示。根据 2.1 节中的干扰物所处平面点云去除算法(即算法 1),借助 RANSAC 算法拟合场景中的任意平面并创造一个用于计算投影距离的虚拟平面。根据式(1)计算点云投影距离,设置 $Prec_d$=0.6,根据式(2)计算点云距离阈值 $thres_d$ 并实现地面点云的去除。

在当前场景中,干扰物所处平面即为地面,因此其效果为地面点云被去除,如图 6 所示。

根据图 6 点云信息,再进行一次 RANSAC 平面拟合,即可获取目标台平面,如图 7(a)所示。如图 7(b),观察平面点云在整体点云中的位置可知,该算法将堆叠物的点云拟合到了平面中,并且目标台的点云具有极其严重的破碎性,因此需要借助串行聚类集合作进一步处理。



(a) 初始点云图 (b) 初始RGB图

Fig. 5 Initial scenario

图 5 初始场景

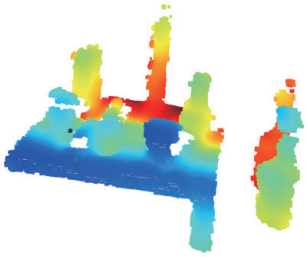
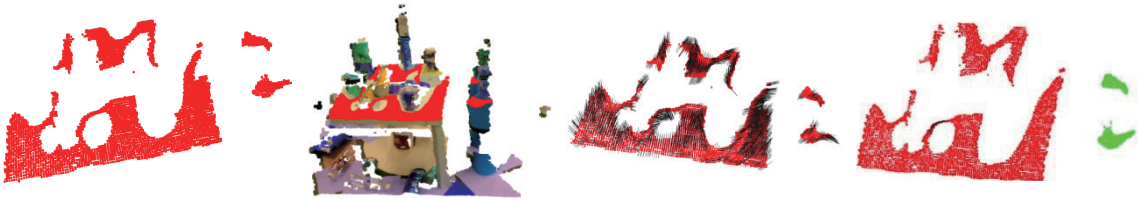


Fig. 6 Removal effect for point cloud of plane that disturbance objects located on

图 6 干扰物所处平面点云去除效果

利用 DBSCAN 算法将平面划分成多个点云簇,该算法实现所给出的案例中总共可划分为 4 个点云簇,每个点云的法向量可视化如图 7(c)所示。可以观察到,异常点云簇的法向量与目标台法向量具有较为明显的差距。通过比较每个点云簇法向量与先前拟合平面的法向量之间的夹角,即可标记并去除



(a) 单独平面点云 (b) 平面点云与物品点云叠加 (c) 法向量可视化 (d) 异常点云簇标记(右侧点云)

Fig. 7 Plane and overall point cloud

图 7 平面与整体点云

异常的点云簇，如图 7(d)所示。

利用 Open3D 中的函数 *compute_convex_hull* 对 3D 点云图进行凸包计算，图 8(a)(b)为凸包的可视化结构，将 3D 凸包点通过坐标转换公式映射到 2D 图中。

如图 8(c)所示，可以观察到在预期目标台中产生了大量的内部干扰点，因而需要借助 Graham 算法再次计算凸包，获取目标台大致轮廓，如图 8(d)所示)。与此同时，生成 2D 与 3D 之间的坐标映射表。借助坐标映射表，将 2D 图中的凸包点映射到 3D 点云空间中，根据式(7)计算得到目标台边缘轮廓平面。



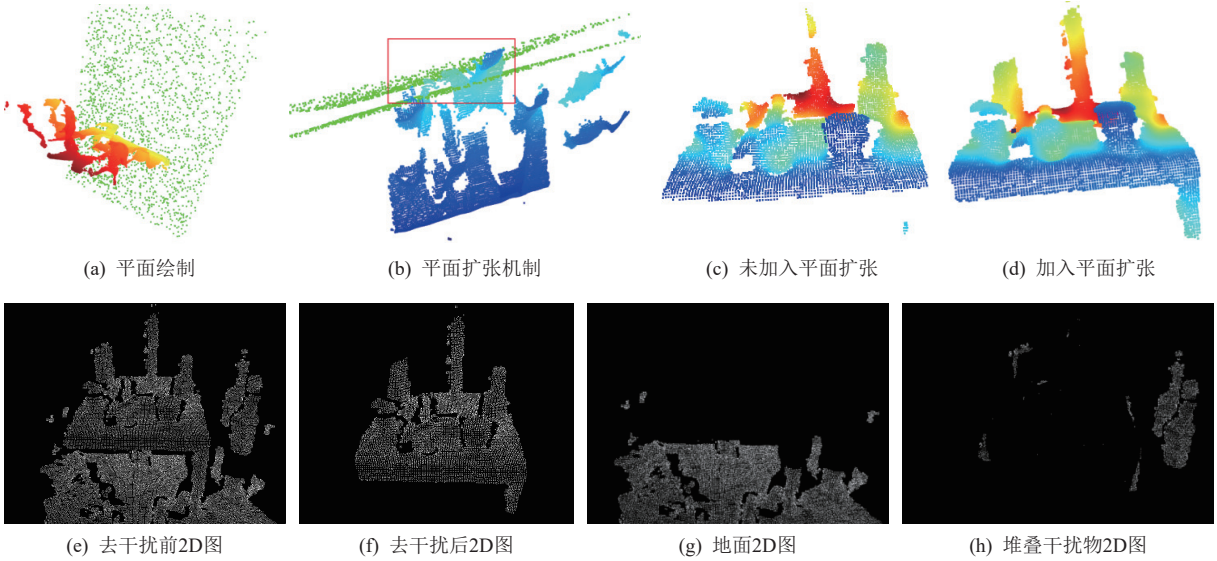
(a) 凸包计算效果1 (b) 凸包计算效果2 (c) 2D凸包点 (d) 2D平面凸包

Fig. 8 Visualization of convex hull calculation and coordinate mapping results

图 8 凸包计算与坐标映射结果可视化

为了直观展示，这里将计算得到的平面绘制在点云图中，如图 9(a)所示。另外，观察图 9(b)我们可以发现，凸包并没有完全包裹住目标台，而是位于平面边缘且略微往目标台中心方向收缩的位置，方框内的所有点属于目标点云，若不采用平面扩张机制，则会导致位于目标点云边缘的点被误删，从而导致目标点云缺乏完整性。

为了防止出现点云误删的情况，我们在 3Deus 中加入了平面扩张机制，通过设定阈值 *minPoint* 来判断 2 个平面之间所包含的点属于目标点云还是属于干扰物。如图 9(c)(d)所示，将加入与不加入平面扩张机制的效果进行对比可以发现，平面扩张机制对保留目标台点云的完整性具有重要的意义，图 9(d)即为最终分析得出的目标台点云。



(a) 平面绘制 (b) 平面扩张机制 (c) 未加入平面扩张 (d) 加入平面扩张

(e) 去干扰前2D图 (f) 去干扰后2D图 (g) 地面2D图 (h) 堆叠干扰物2D图

Fig. 9 Visualization diagram of point cloud detection partial results

图 9 点云探测部分结果可视化图

另外，将目标台点云的 2D 图与最初点云的 2D 图进行对比，如图 9(e)(f)所示，3Deus 去除了桌外的干扰物，较为精准地保留了目标台点云。将先前去除的地面点云与目标台旁的堆叠干扰点云转化到 2D 图，如图 9(g)(h)所示。最后，借助自训练的 YOLOv11 模型，对初始的 RGB 图进行预测，得到每个物品的识别框。将识别框映射到图 9(f)(g)(h)中，根据像素信息即可判断物品处于哪一个位置。

3Deus 不仅在聚焦的桌上目标物、桌下地面干扰、桌旁堆叠干扰共存的场景中表现出优异性能，还可灵活适配多类具有“物品杂乱摆放、多平面干扰、需精准区分空间维度目标与干扰”特征的复杂场景：在智能仓储中，可用于多层货架的物品识别，精准区分各层货架目标物品与立柱、地面干扰物，适配仓储移

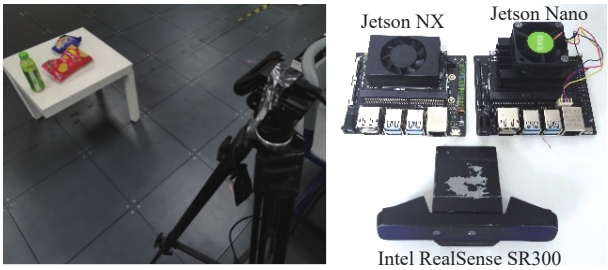
动机器人的现场识别需求;在工业装配线中,能识别传送带上歪斜、堆叠的精密零件,排除两侧工具架干扰,满足装配线实时检测诉求;在智能零售自助结算场景中,可区分结算台表面目标商品与手机、购物袋等干扰物品,匹配自助设备低硬件成本部署需求等。这些场景均需“区分空间干扰、精准识别无序物品”,3Deus 的应用能力可直接适配,进一步验证其应用普适性,拓展了在物联网感知层物品识别任务中的应用范围。

3 实验设置及结果分析

3.1 实验设置

3.1.1 实验场景介绍

目标台与摄像头之间的距离在 1~1.3 m,其相对位置如图 10(a)所示。本实验采用 Intel RealSense SR300 深度摄像头作为场景的点云探测工具,摄像头外观如图 10(b)所示。由于 SR300 摄像头的缺陷以及实验场地的限制,黑色的地板、透明的水瓶、金属制的易拉罐等物品表面会吸收由 SR300 发射出来用于检测场景深度的结构光线,因而使得深度摄像头无法接收反射回来的结构光线而丢失某些表面的深度信息。



(a) 实验场景 (b) 实验设备

Fig. 10 Experimental scenario layout and experimental devices

图 10 实验场景布置及实验设备

在本场景中,大面积的黑色地面会吸收由 SR300 发射出来的结构光而丢失深度信息,因此,为了使得场景更贴近实际应用,我们在地板上放置了大量彩色卡纸,避免地板的深度信息丢失。深度摄像头是本实验的关键,针对当前实验场景,我们仔细调整了摄像头的各个参数,其关键数值如表 1 所示。另外,为了验证算法在嵌入式设备上的可行性,将 3Deus 分别部署在了 Jetson Nano 与 Jetson NX 上,如图 10(b)所示。

在本文提出的算法中,关键参数的设置对系统性能有着显著影响。首先,干扰平面去除阈值 $Prec_d$ 设置为 0.6。若该值过小,会导致地面干扰点云去除不足;若该值过大,则会过度去除点云,误删目标台有效边缘,导致平面残缺。深度相机参数(如 *depth accuracy*, *filter_option* 等)共同决定了原始点云的质量与密度。对于平面扩张机制,参数 *minPoint* 决定了平面扩张的终止条件:其值过小会使平面过度扩张,将邻近干扰物纳入目标平面;其值过大则会导致扩张提前终止,无法得到完整的目标平面轮廓。扩张系数(*exps*)控制着扩张的步长与精度:过小的 *exps* 值会使扩张过程缓慢且对噪声敏感;过大的 *exps* 值则会降低边界定位精度,导致平面轮廓扭曲。算法性能依赖于上述参数的协同优化,若调整任一参数,均会使 3Deus 的整体性能下降。

Table 1 Parameters of the Depth Camera

表 1 深度摄像头参数

参数	取值
深度信息精度/mm	3
跟踪的对象的最大运动范围/m	4
深度滤波级别	6
可靠性阈值	3

针对本文所提出的 3Deus,我们模拟了一系列实验场景,用于测试 3Deus 的效果,实验场景由多张白色方桌与圆桌、多张彩色卡纸、大量目标物品以及数个物品识别模型之外的物品组成;目标台上的物品采用常见的日用品,物品摆放形式多样,包括且不限于正放、倒放、斜放、堆叠,物品之间存在倚靠和遮挡,其目的在于使得目标台平面点云具有一定的破碎性,以测试 3Deus 对于不同场景的适应性。

3.1.2 数据集构建

我们按照之前所描述的实验装置与场景设置,手动拍摄数据集,并根据干扰物的数量、空间位置以及目标台摆放形式,对场景进行分类,具体实验场景如图 11 和图 12 所示。

图 11 中,干扰场景 1 为仅目标台上有目标物品。干扰场景 2 为目标台上有目标物品、目标台下有干扰物。干扰场景 3 为目标台上有目标物品、目标台下有干扰物、目标台旁有堆叠物。干扰场景 4 在干扰场景 3 的基础上将目标台斜放。在以上 4 种干扰场景的目标台上,放置一定数量的目标物品,目标物品皆为深度学习物品识别模型中所涉及的。在目标台下(即地上),也会放置一定数量的物品识别模型内的物品,同时也会在远处放置一定数量与模型无关的物品,但受到深度摄像头的参数限制,远处的物



(a) 干扰场景1 (b) 干扰场景2 (c) 干扰场景3 (d) 干扰场景4

Fig. 11 Specific experimental scenarios

图 11 具体实验场景



(a) 多水平桌面实验场景 (b) 水平加倾斜桌面实验场景 (c) 不平整桌面实验场景

Fig. 12 Specific multi-desktop experimental scenarios

图 12 具体多桌面实验场景

品并不会接收到深度摄像头发射出的结构光，因而不会含有深度信息，也不会对 3Deus 的效果造成干扰，因此可以忽略不计。数据集的每张图片中，目标物品与干扰物的摆放位置与数量随机。对每种干扰场景拍摄 50 张图片，同时保存图片对应的点云图，总共拍摄 200 张图片以及点云图。

同时，为验证 3Deus 算法的场景可扩展性，我们测试了其在多桌面条件下的性能：

1）如图 12(a)所示，设置多目标平面随机布局场景，随机水平摆放 1 台圆桌与 2 台方桌，构成 3 个目标桌面，重点验证算法对多个水平平面以及其上物品的精准识别能力，以及对地面干扰目标的排除能力；

2）如图 12(b)所示，为了验证算法对倾斜桌面的识别，增加一个倾斜桌面的方形桌面与正常的水平方形桌面和水平圆形桌面构成组合式实验环境，其中水平方桌和水平圆桌作为目标平面，倾斜方桌作为非目标干扰平面，通过对多桌面的点云提取，需要在识别并剔除倾斜桌面后，仅对水平桌面上的物品进行有效识别，重点验证算法在多目标平面与倾斜干扰平面共存的复杂场景下，对目标平面的区分识别准确性以及稳定性；

3）如图 12(c)所示，为了验证算法的场景拓展性，设置多目标不平整平面随机布局场景，随机摆放不平整桌面桌子 2 台，其上有不平整桌布以模拟不平整平面，在其表面、台体周边空间及台面下方 3 个空间位置随机摆放物品，重点验证算法在更灵活场景下的可扩展性。

我们对 3 个实验场景中桌面的摆放位置及摆放物品进行随机调整，并采集数据集。每个实验场景采集 200 张图像，新增 3 组实验累计构建 600 张图像的基准数据集。同时保存图片对应的点云图，进一步扩充数据集维度，全面验证 3Deus 算法在多水平平面场景、水平加倾斜平面场景以及不平整平面场景中的性能，以确保 3Deus 在实际应用场景中能有效应对各种干扰以及布局变化。

3.1.3 评价指标

3Deus 的重点在于目标台识别与干扰物去除。在实验中，干扰物包括地面干扰物与堆叠干扰物，为了能够直观地表现模型对目标物与干扰物的区分精度，这里采用了目标检测中常用的指标，即混淆矩阵、准确率(accuracy)、精准率(precision)、召回率(recall)、*F*1 分数。准确率记录了分类正确的样本数占总样本数的比例，精准率记录正确预测为正样本的样本数占预测为正样本总样本数的比例，召回率记录了正确预测为正样本的样本数占实际正样本总样本数的比例。

由于精准率和召回率两者存在一定的权衡关系，即精准度的上升可能会伴随着召回率的下降，因此精准率和召回率并不能完整地对不同方法进行综合衡量，于是，我们引入 *F*1 分数以及整体识别准确率，将桌旁与桌上的识别 *F*1 分数，用均方根误差(root mean square error, *RMSE*)计算，得到平面扩张机制对

点云探测算法的整体提升效果为 30.48%。*RMSE* 的计算式如式(8)所示，其中 *EU* 表示桌上物品识别的 *F*1 分数差，*ED* 表示桌下物品识别的 *F*1 分数差。

$$RMSE=\sqrt{\frac{EU^2+ED^2}{2}}。\tag{8}$$

另外，本实验还需要测试 3Deus 在嵌入式设备上的运行时间，因此在统计实验中将记录 3Deus 在 Jetson Nano 与 Jetson NX 上的平均运行时间，并与 PC 机进行比较。

3.2 对比实验及结果分析

1)单平面场景统计与对比实验

为了检测 3Deus 在强干扰场景下的适应性以及关键步骤，同时验证基于聚类集合改进的 RANSAC 算法和平面扩张机制的重要性，我们引入了 2 个对比实验。第 1 个实验为仅包含地面点云去除的算法，在去除地面点云后，直接使用传统 RANSAC 算法拟合剩余点云中的目标台。第 2 个实验则是去除平面扩张机制的算法，在点云探测的最后阶段不执行平面扩张机制。具体的对比结果展示在表 2~4 中。

Table 2 Confusion Matrix of 3Deus

表 2 3Deus 混淆矩阵

场景	预测种类	真实被预测数量		
		桌上	桌下	桌旁
干扰场景 1	桌上	499	0	1
	桌下	0	4	0
	桌旁	0	0	0
干扰场景 2	桌上	511	12	0
	桌下	0	213	0
	桌旁	12	0	0
干扰场景 3	桌上	519	9	15
	桌下	0	149	0
	桌旁	3	0	91
干扰场景 4	桌上	470	12	23
	桌下	0	186	0
	桌旁	2	0	108

通过对比实验，我们记录了各实验结果的混淆矩阵。表 2 展示了 3Deus 的混淆矩阵，表 3 展示了仅包含地面点云去除算法的结果，表 4 则展示了去除平面扩张机制后的算法效果。

根据实验结果，我们可以从表 2~4 中看到算法在物品识别任务中的表现差异。首先，表 2 展示了 3Deus 的混淆矩阵及其识别效果，表明算法在各种场景下都能够较好地识别目标物品，尤其是在桌上和桌下的物品识别中，表现出较高的精准率和召回率。相比之下，表 3 中的结果显示，在仅去除地面点云的算法中，识别效果有所下降，尤其是在桌旁物品的识别上，误识别和漏识别的情况明显增多。这表明去除地面点云后的传统 RANSAC 算法在面对复杂的点云数据时，存在一定的局限性。

Table 3 Confusion Matrix Only Removing Ground Point Cloud Algorithm

表 3 仅去除地面点云算法的混淆矩阵

场景	预测种类	真实类别		
		桌上	桌下	桌旁
干扰场景 1	桌上	498	0	1
	桌下	0	3	0
	桌旁	1	1	0
干扰场景 2	桌上	515	11	0
	桌下	0	208	0
	桌旁	8	6	0
干扰场景 3	桌上	521	11	106
	桌下	0	146	0
	桌旁	1	1	0
干扰场景 4	桌上	472	14	132
	桌下	0	183	0
	桌旁	0	1	1

Table 4 Recognition Confusion Matrix Without Plane Expansion Mechanism

表 4 无平面扩张机制的识别混淆矩阵

场景	预测种类	真实类别		
		桌上	桌下	桌旁
干扰场景 1	桌上	400	0	1
	桌下	0	4	0
	桌旁	99	0	0
干扰场景 2	桌上	376	7	0
	桌下	0	217	0
	桌旁	147	1	0
干扰场景 3	桌上	368	10	7
	桌下	0	147	2
	桌旁	154	1	97
干扰场景 4	桌上	314	6	7
	桌下	0	186	5
	桌旁	158	6	119

进一步地，表 4 展示了去除平面扩张机制后的算法效果。从结果来看，去除平面扩张机制会导致桌旁物品的识别准确率大幅下降，尤其是在复杂场景中，平面扩张机制的缺失导致了识别精度的显著降

低。这证明了平面扩张机制在处理边缘物品时的重要性，尤其是在处理物品遮挡或点云破碎性问题时，扩张机制能够有效提升算法的鲁棒性和识别准确性。

综合以上分析，3Deus 在所有场景中，对桌上、桌下、桌旁的物品都能达到极高的识别准确率，且关键步骤平面扩张机制和聚类集合改进的 RANSAC 算法对提高物品识别性能具有重要作用。本文提出的基于聚类集合改进的 RANSAC 算法和平面扩张机制对于识别准确率有很大的提升。

为验证所提 3Deus 算法与现有主流算法的性能差异，我们在目标场景中，将 3Deus 与 MO-RANSAC[41]，PlaneRCNN[42]，MAGSAC[43]，PlaneNet[44] 等主流算法开展对比实验，以“识别准确率”为核心评价指标，各对比算法的实验结果如下：PlaneNet 平均识别准确率为 83.47%，PlaneRCNN 为 87.82%，MAGSAC 为 88.65%，MO-RANSAC 为 92.71%，而 3Deus 的识别准确率达到 96.93%，在所有对比算法中实现最优性能；具体来看，3Deus 相较于 PlaneNet 提升 13.46 个百分点，相较于 PlaneRCNN 提升 9.11 个百分点，相较于 MAGSAC 提升 8.28 个百分点，即便与性能较优的 MO-RANSAC 相比，仍实现 4.22 个百分点的准确率提升，实验结果如图 13 所示。

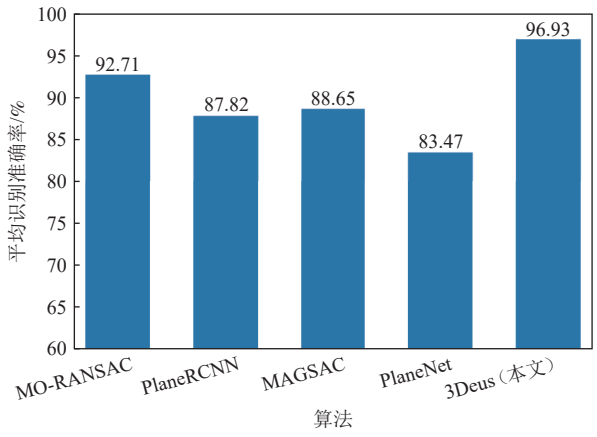


Fig. 13 Performance comparison of 3Deus and other algorithms

图 13 3Deus 与其他算法性能对比

3Deus 算法实现了最高识别率，主要是因为凭借以下关键技术改进：针对传统算法仅能拟合空间中“点云数量最多的扁平区域”，无法区分目标平面与高度平齐的堆叠干扰物（如桌旁堆叠的同类物品）这一难题，3Deus 采用“串行聚类集合改进的 RANSAC 算法，能够借助点云簇的法向量夹角进行判断，精准滤除堆叠干扰点云，避免将干扰物误判为目标；同时，面对其他算法在物品遮挡场景下易出现的“平面破碎、边缘点云误删”问题（如目标台边缘因物品遮挡导致点云断裂，算法误删边缘目标点云），3Deus 通过动态平面扩张机制，可保留原本属于目标平面的边缘点云，修复破碎平面，减少目标物品漏识别。

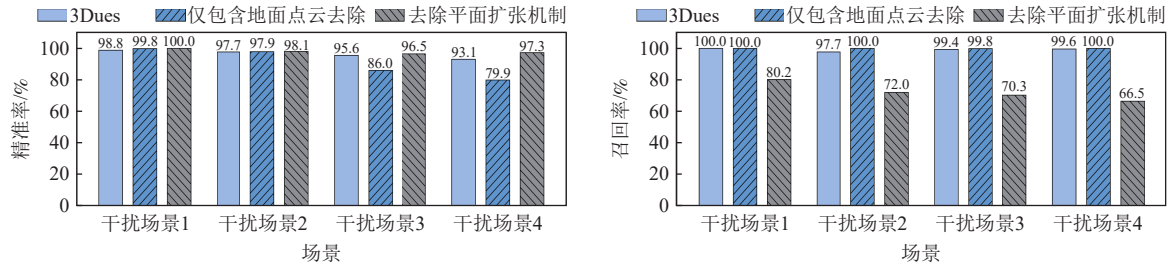
依据生成的不同场景的数据集，结合评价指标，验证算法的准确性与鲁棒性。根据第 2 节中对算法的描述，以目标台与干扰物所处平面为基准，将识别目标细分为以下 3 类：桌上、桌旁和桌下。桌上类属于目标台上需要识别出来的目标物，桌下类和桌旁类属于识别的干扰物。

在 3Deus 的运行过程中，分别将桌上、桌旁和桌下的点云信息映射到 2D 图上，如图 9(f)（g）（h)所示。经 YOLOv11 物品识别模型，得到 RGB 图中每个物品的识别框，并将其映射到点云映射的 2D 图中，根据识别框内白色像素信息的数量从而判断物品的位置情况。对各种位置情况，分别统计混淆矩阵，计算准确率、精准率、召回率、$F1$ 分数。

混淆矩阵是一种评估分类模型性能的表格化方式，但并不能直观地体现模型之间的性能优劣。因此，根据以上统计出的混淆矩阵，我们计算了每一种场景下，桌上、桌下、桌旁物品的识别精准率和召回率，以及整体的识别准确率，将其计算结果绘制在柱状图上。在每一个干扰场景中，将 3 种对比算法绘制在一起以方便比较，如图 14~16 所示，可以直观地对比出不同算法在不同干扰场景下对物品识别的水平，反映出探测算法中关键步骤的重要性。图 14~16 分别表示桌上物品、桌下物品、桌旁物品的识别精准率与召回率。

由图 14~16 观察得出，3Deus 在每一个干扰场景，对所有位置物品的识别效果都保持在极高的水准；对于桌下物品，3 种算法都能达到接近 100% 的精准率，这是因为 3 种算法都包含了底端点云去除步骤，能够区分出桌下（即地面）的干扰物。因此该算法具有极高的鲁棒性，可适配各类场景并有效去除地面干扰物。地面干扰物的去除在算法实现上较为简单，后续实验结果分析中不再赘述。

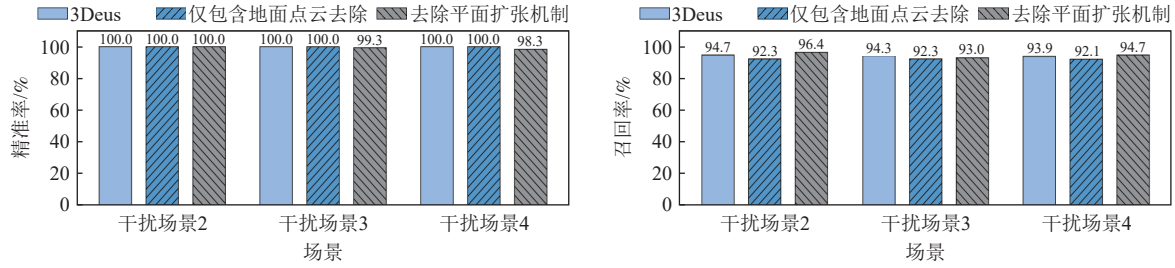
与 3Deus 相比，另外 2 种算法对于桌上物品和桌旁物品的识别效果有一定程度的下降。去除平面扩张机制的算法对桌上物品识别的召回率相较于另外 2 个算法有 27 个百分点左右的下降比例（如图 14(b)所示），这是因为破损的平面会导致凸包算法不能完整地包裹住目标台，使得目标台点云的边缘部分被过多地删除，而桌上目标物品的识别需要依靠目标台点云信息，因而造成了在物品识别过程中遗漏了



(a) 桌上物品识别精准率 (b) 桌上物品识别召回率

Fig. 14 Recognition results for object on the table

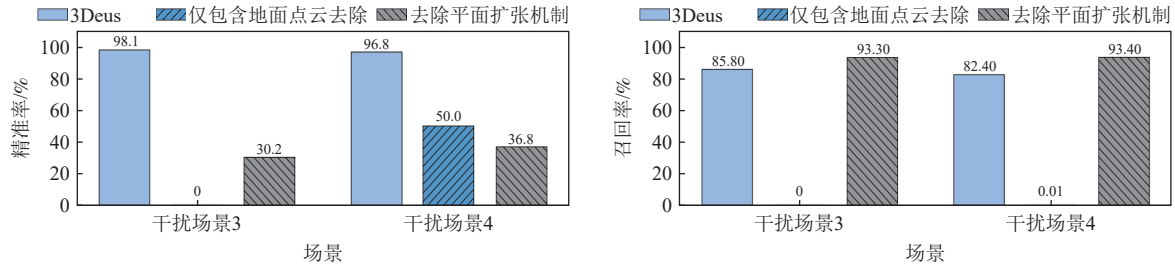
图 14 桌上物品识别结果



(a) 桌下物品识别精准率 (b) 桌下物品识别召回率

Fig. 15 Recognition results for objects under the table

图 15 桌下物品识别结果



(a) 桌旁物品识别精准率 (b) 桌旁物品识别召回率

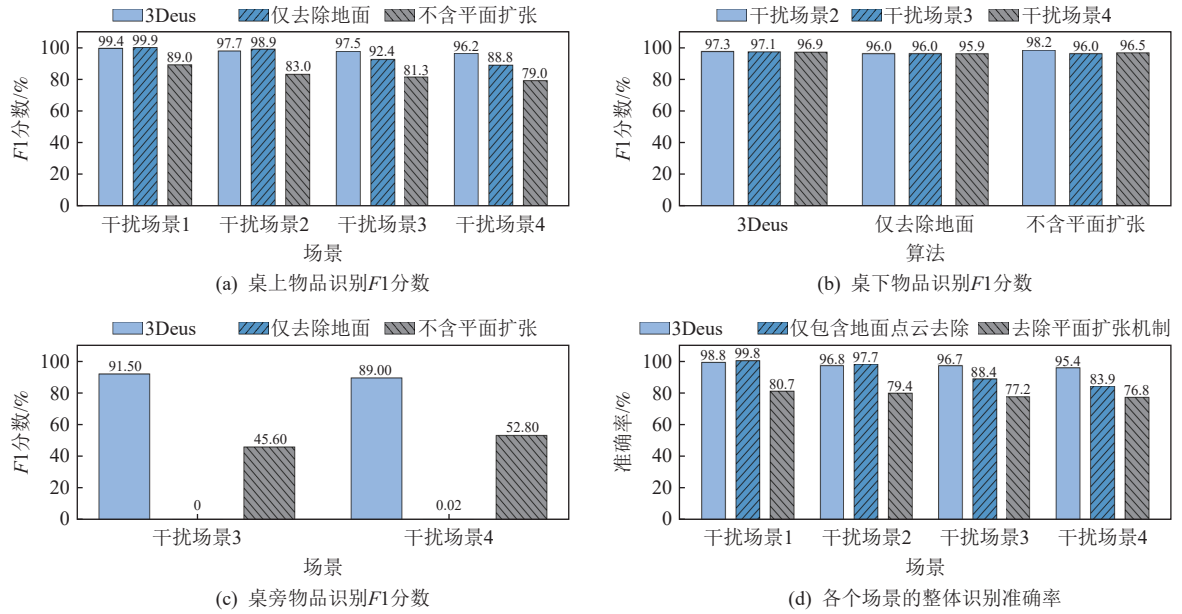
Fig. 16 Recognition results for objects beside the table

图 16 桌旁物品识别结果

位于目标台边缘位置的物品。或者说，错误地将其识别为桌旁物品，从而导致其召回率下降。但桌下物品与桌旁物品不会被错识别为桌上物品，因此对于桌上物品的识别，3 种算法都具有较高的精准率(如图 14(a)所示)。相对应的，由图 16(a)可以得知，去除平面扩张机制将导致桌上物品大量地被识别为桌旁物品，导致对于桌旁物品识别的精准率急剧下降。然而，由图 16(b)可以得知，原本位于桌旁的物品却能准确识别出来。因此，对于桌旁物品识别具有较高召回率。

此外，仅包含地面点云去除的算法，虽然对于桌下物品和桌上物品的识别能达到较高的精准率，但该算法在去除干扰物所处平面点云之后，直接采用传统的 RANSAC 算法进行平面拟合，目标台两端的堆叠物全部纳入到目标台平面点云中，因此无法对桌旁物品和桌上物品进行区分。观察图 14 和图 16 中的干扰场景 3 和干扰场景 4 的统计数据可知，3Deus 对于桌上物品识别具有较高的召回率和较低的精准率，这是因为原本的桌旁物品几乎全部被识别为桌上物品，而原本位于桌上的物品却能够全部被识别正确。对于桌旁物品，其精准率与召回率下降接近于 0，由此也可以体现出传统 RANSAC 算法在堆叠物与目标台识别上的局限性。

如图 17 所示。总体而言，3Deus 的性能普遍高于另外 2 种算法，在不同干扰场景下对目标物以及干扰物的区分率以及 $F1$ 分数均达到了 95% 以上。为了体现平面扩张机制在整个算法中的重要性，这里将去除平面扩张机制的算法与 3Deus 进行比较，在桌上物品的识别效果上，3Deus 平均提升了 15.15 个百分点的 $F1$ 分数，在对桌旁物品的识别效果上提升



(a) 桌上物品识别$F1$分数 (b) 桌下物品识别$F1$分数

(c) 桌旁物品识别$F1$分数 (d) 各个场景的整体识别准确率

Fig. 17 Overall recognition results of each scenario

图 17 各场景整体识别结果

了 40.35 个百分点，在整体识别准确率上提升了 13.8 个百分点。但是由于整体识别准确率中包含了对桌下物品的识别效果，因此其不具备参考价值。

仅包含去除地面点云的算法在干扰场景 1 和干扰场景 2 中对桌上和桌下物品都有较高的 $F1$ 分数和准确率，这是因为干扰场景 1 和干扰场景 2 中不包含桌旁物品，使得传统的 RANSAC 算法能够适应。然而在干扰场景 3 和干扰场景 4 中，该算法性能则急剧下降，这也进一步体现了传统的 RANSAC 算法无法区分桌上物品和桌旁物品。单平面场景识别结果如图 18 所示。

2）多平面复杂场景统计与对比实验

首先，将原来单一桌面扩展到 3 个目标桌面。在 3 个目标桌面、桌体周边及桌体下方 3 个空间位置随机设置摆放物品。不在桌面上的物品均视为干扰物。通过该实验旨在验证算法在“物品杂乱分布、多平面干扰共存、需精准区分空间维度目标与干扰”的复杂场景下的识别精度与稳定性，多平面复杂场景识别结果如图 19 所示。

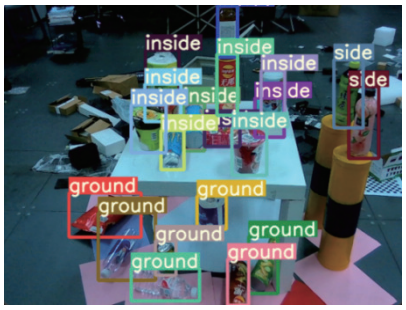
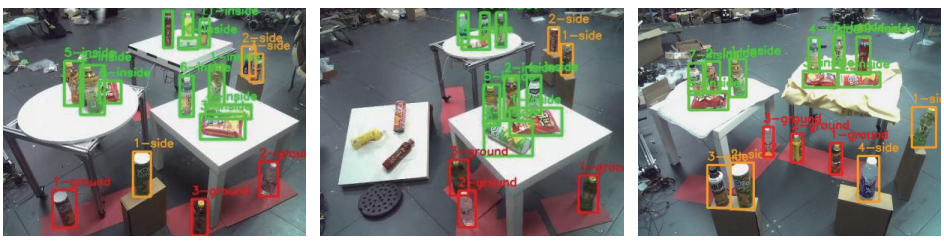


Fig. 18 Recognition results under single-plane scenario

图 18 单平面场景下的识别结果

表 5 展示了 3Deus 在多水平桌面实验场景中的识别效果。可以看到，在桌上与桌下类别中，算法依旧保持了较高的精准率与召回率。从混淆矩阵的各场景分类结果可以直观发现，尽管场景中存在多个水平平面以及不同高度的物体干扰，算法仍能够有效捕捉目标物品的空间位置特征（桌上、桌下、桌旁），并在大多数情形下完成准确区分。



(a) 多水平桌面实验场景 (b) 水平加倾斜桌面实验场景 (c) 不平整桌面实验场景

Fig. 19 Recognition results under multi-plane complex scenarios

图 19 多平面复杂场景下的识别结果

Table 5 Confusion Matrix of 3Deus Under Multi-horizontal Desktop Experimental Scenarios

表 5 多水平桌面实验场景下 3Deus 的混淆矩阵

场景	预测种类	真实被预测数量		
		桌上	桌下	桌旁
干扰场景 1	桌上	535	0	0
	桌下	0	0	0
	桌旁	10	0	0
干扰场景 2	桌上	562	16	0
	桌下	0	49	0
	桌旁	7	3	0
干扰场景 3	桌上	561	4	22
	桌下	0	121	0
	桌旁	14	0	174
干扰场景 4	桌上	541	5	23
	桌下	0	169	8
	桌旁	16	0	189

在 4 个干扰场景下算法的准确率、召回率和 $F1$ 分数整体维持在较高水平，其中干扰场景 1 的性能最为突出，准确率达到 100%，$F1$ 分数亦接近 99%。这一结果主要得益于核心平面面积较大、结构边界清晰，使算法能够充分利用稳定且显著的几何特征，实现对桌上与桌下物品几乎无误判的精确识别。相比之下，干扰场景 2、干扰场景 3 与干扰场景 4 的部分指标略有下降(但仍基本保持高于 90%)，如图 20 所示。这种变化趋势与表 5 混淆矩阵中的分类结果相吻合。随着场景复杂度不断提升，3 个平面之间的空间关系愈发交织，桌旁区域与邻近平面在高度层级和局部几何特征上更易产生重叠，导致算法在这些区域出现一定程度的特征混淆，进而引发少量误判与漏检。尽管如此，整体性能仍处于较高水准，充分说明算法在面对多平面共存、遮挡增强以及物体布局密集的复杂 3D 场景，依旧具备稳定的识别能力与良好的空间特征适配性。

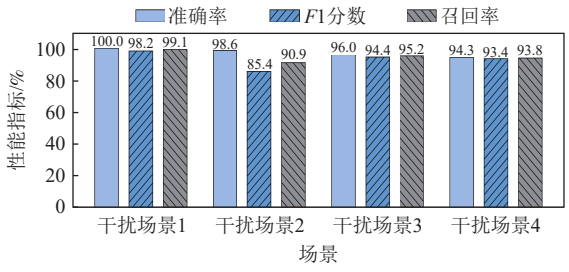


Fig. 20 Performance of the algorithm under the multi-horizontal desktop experimental scenario

图 20 多水平桌面实验场景下的算法性能

其次，进一步考虑实际场景中多目标平面与干扰平面交织共存的复杂情况，为全面验证 3Deus 算法的场景适应性，本文构建“水平加倾斜桌面实验场景”的复合测试场景，以水平圆桌与水平方桌作为双目标平面，搭配倾斜方桌作为非目标干扰平面，重点考察算法在多目标与非水平干扰交织环境下，对目标平面的区分准确率与定位精度及目标物品识别性能的稳定性。

表 6 展示了 3Deus 在“水平加倾斜桌面实验场景”的混淆矩阵及其识别效果，根据表格结果我们可以得出算法在这一交织型复杂场景下都能够较好地识别目标物品。

Table 6 Confusion Matrix of 3Deus Under Horizontal Plus Inclined Desktop Experimental Scenarios

表 6 水平加倾斜桌面实验场景下 3Deus 的混淆矩阵

场景	预测种类	真实被预测数量		
		桌上	桌下	桌旁
干扰场景 1	桌上	507	0	0
	桌下	8	0	0
	桌旁	6	0	0
干扰场景 2	桌上	518	0	0
	桌下	4	154	0
	桌旁	19	0	0
干扰场景 3	桌上	514	0	2
	桌下	6	137	1
	桌旁	27	0	106
干扰场景 4	桌上	513	0	5
	桌下	8	103	0
	桌旁	27	1	159

从图 21 中可以看出，在“水平加倾斜桌面实验场景”的复合测试场景中，算法在 4 个干扰场景下的准确率、召回率和 $F1$ 分数整体依旧维持在较高水平，但随着场景复杂度的提升呈现出一定幅度的波动。

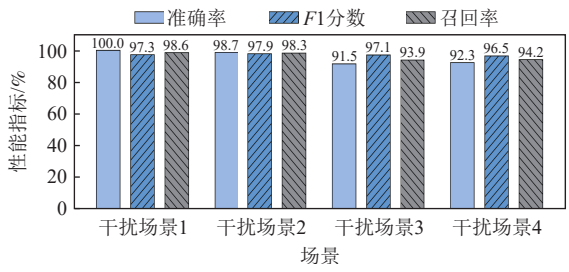


Fig. 21 Performance of the algorithm under the horizontal plus inclined desktop experimental scenario

图 21 水平加倾斜桌面实验场景下的算法性能

其中，干扰场景 1 与干扰场景 2 的表现最为优异，各项指标几乎均超过 98%。

这一结果主要归因于 2 张目标平面占据了主要空间区域，使整体几何结构清晰、边界分明，算法能够高效区分主目标平面与倾斜干扰平面，从而显著降低特征混淆，特别是在桌上与桌下类别中表现出更强的识别优势。相比之下，干扰场景 3 与干扰场景 4 的指标略有下降（但仍保持在 91%~97%）。这类指标下降主要源于双核心桌面与周围物体布局更加密集交织，使场景的空间边界更加复杂，局部几何特征更易受到干扰，尤其是桌旁区域在复杂空间结构影响下更容易产生特征混淆，进而导致少量误判与漏检。尽管如此，整体性能仍保持在较高水准，表明 3Deus 在多平面相互作用、结构复杂度显著提高的条件下仍具备良好的适应性与鲁棒性。

最后，为了验证 3Deus 对非理想平面的适应能力及场景拓展性，我们设置多目标不平整平面随机布局场景：随机摆放 2 台覆盖不平整桌布的桌子以模拟表面起伏环境，并同样在其表面、台体周边空间及台面下方 3 个空间位置随机摆放物品。本实验重点考察算法在平面存在起伏和形变干扰的复杂情形下，能否依然准确提取目标支撑面并有效识别不同空间维度上的目标。

表 7 展示了 3Deus 在不平整桌面实验场景中的识别效果。可以看出，尽管桌布的覆盖导致平面几何特征出现不规则波动，算法在“桌上”与“桌下”类别的判定中依然保持了极高的准确性。分析混淆矩阵可知，在干扰场景 1 和干扰场景 2 中，针对核心区域物品的误判极少；而在干扰场景 3 和干扰场景 4 中，出现的少量混淆主要集中在“桌旁”与“桌上”的边缘交界处，这主要是由于表面的不平整特性在一定程度上干扰了平面区域的扩张与边界界定。尽管如此，整体超过 93% 的准确率充分表明，算法依旧能够鲁棒地提取出有效的目标支撑面特征，并实现精准的空间区域划分。

Table 7 Confusion Matrix of 3Deus Under Uneven Desktop Experimental Scenarios

表 7 不平整桌面实验场景下 3Deus 的混淆矩阵

场景	预测种类	真实类别		
		桌上	桌下	桌旁
干扰场景 1	桌上	554	0	0
	桌下	1	0	0
	桌旁	11	0	0
干扰场景 2	桌上	547	0	0
	桌下	0	155	0
	桌旁	16	0	0
干扰场景 3	桌上	527	2	7
	桌下	0	63	0
	桌旁	10	2	134
干扰场景 4	桌上	577	9	3
	桌下	0	108	2
	桌旁	23	0	129

在 4 个干扰场景下算法的准确率、召回率和 $F1$ 分数整体维持在较高水平，如图 22 所示，其中干扰场景 1 与干扰场景 2 的性能最为突出，各项指标均十分优异。这一结果主要得益于尽管桌面存在表面起伏，但核心平面的整体几何特征没有完全丢失，使算法能够有效克服微小形变的干扰，实现对桌上与桌下物品几乎无误判的精确识别。

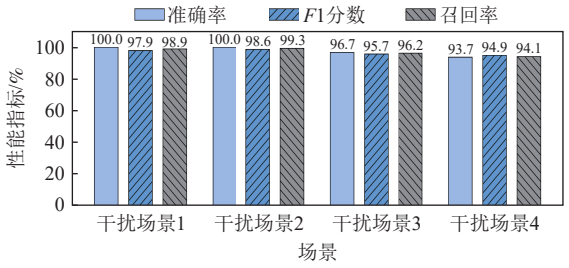


Fig. 22 Performance of the algorithm under the uneven desktop experimental scenario

图 22 不平整桌面实验场景下的算法性能

相比之下，干扰场景 3 与干扰场景 4 的部分指标略有下降（但 $F1$ 分数仍保持在 94% 以上）。这主要是因为不平整桌面的边缘几何特征相对模糊，当物品处于桌旁或边缘位置时，褶皱的桌布容易与临近物体产生局部点云粘连，增加了特征分割的难度，进而导致少量误判。尽管如此，整体性能仍处于较高水准，充分说明算法在面对不平整平面与复杂空间布局共存的挑战时，依旧具备稳定的识别能力与优秀的场景可扩展性。

综上所述，以上 3 个实验结果充分验证了 3Deus 不仅在单一固定研究场景中具备优异的目标识别性能，更在多水平平面场景、水平加倾斜平面场景以及不平整平面场景下，均能实现精准、稳定的识别效果，充分彰显了算法强大的场景适配性与灵活扩展能力。

3）时间开销评估

将 3Deus 分别部署在 Jetson Nano 与 Jetson NX 上，记录其在每种场景下的运行平均时间，并与 Windows 操作系统下的 PC 机进行对比，PC 的 CPU 配置为 Intel® Core™ i7-10510U，GPU 配置为 NVIDIA

GeForce MX350。

从表 8 可以看出，尽管 PC 机的运行时间明显短于嵌入式设备，Jetson NX 和 Jetson Nano 的运行时间分别为 4.24 s 和 12.56 s，这表明该算法在嵌入式设备上运行的条件已经具备。虽然 Jetson NX 性能较强，能够在较短时间内完成任务，但 Jetson Nano 也能在合理的时间范围内完成计算，特别适合资源受限的应用场景。因此，该算法不仅适用于高性能 PC 机，也能够在嵌入式设备上高效运行，满足不同硬件环境下的需求，具备在嵌入式设备上运行的条件。

Table 8 Running Time on Different Devices

表 8 不同设备上的运行时间 s

设备名称	平均运行时间
Jetson Nano	12.56
Jetson NX	4.24
PC 机	2.35

4 结 论

当前的物品识别技术作为物联网应用感知层的重要组成部分，在面对复杂场景时适应能力偏弱。为此本文提出了一种基于点云探测的复杂场景下物品识别方法——3Deus。该方法利用聚类集合改进了传统的 RANSAC 算法，通过准确地去除干扰物所处平面点云获取目标台点云。在此基础上进一步利用凸包算法、平面扩张机制以及 3 维空间关系，准确建立目标物品及其所处平面的点云模型，并结合物品识别模型的输出与点云探测的结果，同步实现目标物品高精度识别与干扰物的准确区分。通过对比实验，我们验证了将 3Deus 应用在多个复杂场景之后，一旦去除干扰物品后，能够有效提高目标物品的识别精度，且该方法适合部署在计算资源受限的嵌入式设备上。

在未来的工作中，继续探索 3Deus 在实时性方面的进一步优化的可能性，使其能够在更加复杂的场景中实现更高效的物品识别。

作者贡献声明：邵智超明确研究挑战，提出解决思路，设计并完成实验和撰写论文；邱杰凡提出问题，参与解决方案设计，对论文撰写与实验方案设计进行指导；戴光麟参与方案设计和撰写论文；李乔宏调研文献以及参与算法设计；陈翰墨调研问题背景，对实验、论文写作进行指导；肖克江对整体研究方向和思路进行指导。

参 考 文 献

[1] Park J, An G, Lee B N, et al. Real-time CNN-based object detection of prohibited items for X-ray security screening[J]. Radiation Physics and Chemistry, 2025, 232: 112681

[2] Zu Mengmeng, Jia Jia, Wang Yuan, et al. Research on the application of target detection and recognition technology based on deep learning in industrial automation[C]//Proc of the 2nd Int Conf on Power, Electrical Engineering, Electronics and Control. Piscataway, NJ: IEEE, 2024: 615−620

[3] Pei Jiabin, Wu Xiaoming, Liu Xiangzhi, et al. SGD-YOLOv5: A small object detection model for complex industrial environments[C]//Proc of the 41st Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 2024: 1−10

[4] Li Shiyang, Chen Jing, Peng Weimin, et al. A vehicle detection method based on disparity segmentation[J]. Multimedia Tools and Applications, 2023, 82(13): 19643−19655

[5] Wang Yiqing. An analysis of the Viola-Jones face detection algorithm[J]. Image Processing on Line, 2014, 4: 128−148

[6] Hassan M M, Hussein H I, Eesa A S, et al. Face recognition based on gabor feature extraction followed by fastica and LDA[J]. Computers, Materials and Continua, 2021, 68(2): 1637−1659

[7] Ramu G, Babu S B G T. Image forgery detection for high resolution images using SIFT and RANSAC algorithm[C]//Proc of the 2nd Int Conf on Communication and Electronics Systems. Piscataway, NJ: IEEE, 2017: 850−854

[8] Ravi N, Gabeur V, Hu Yuanting, et al. SAM2: Segment anything in images and videos[J]. arXiv preprint, arXiv: 2408.00714, 2024

[9] Khanam R, Hussain M. YOLOv11: An overview of the key architectural enhancements[J]. arXiv preprint, arXiv: 2410.17725, 2024

[10] Liu Honyi, Tseng Y, Chang K, et al. A denoising FPN with transformer R-CNN for tiny object detection[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62: 1−15

[11] Zhu Ran, Chen Silu, Li Xiaocong, et al. Recognition and localization method for high-speed robotic sorting in complex environments based on image interference assessment[J]. International Journal of Intelligent Robotics and Applications, 2025, 9(3): 1173−1186

[12] Sidehabi S W, Azis M F, Asbar M. Development of a color-based image recognition system for robotic sorting and picking[J]. INTEK: Jurnal Penelitian, 2024, 11(2): 77−82

[13] Titu M F S, Haque S M R, Islam R, et al. Experiments with cooperative robots that can detect object's shape, color and size to perform tasks in industrial workplaces[J]. International Journal of Intelligent Robotics and Applications, 2024, 8(1): 179−192

[14] Ali S, Pandey A K. ArthroNet: A monocular depth estimation technique with 3D segmented maps for knee arthroscopy[J]. Intelligent Medicine, 2023, 3(2): 129−138

[15] Liu Sixing, Liu Mingmiao, Chai Yan, et al. Recognition and location of pepper picking based on improved YOLOv5s and depth camera[J]. Applied Engineering in Agriculture, 2023, 39(2): 179−185

[16] Thanh P T H, Bui M Q V, Nguyen D D, et al. Transfer multi-source knowledge via scale-aware online domain adaptation in depth estimation for autonomous driving[J]. Image and Vision Computing, 2024, 141: 104871

[17] Yang Tao, Gu Feifei. Overview of modulation techniques for spatially structured-light 3D imaging[J]. Optics & Laser Technology, 2024, 169: 110037

[18] Zheng Yuanzhou, Liu Peng, Qian Long, et al. Recognition and depth estimation of ships based on binocular stereo vision[J]. Journal of Marine Science and Engineering, 2022, 10(8): 1153

[19] Zhang Hongxin, Lee S. Advancing the robotic vision revolution: Development and evaluation of a bionic binocular system for enhanced robotic vision[J]. Biomimetics, 2024, 9(6): 371

[20] Bamji C, Godbaz J, Oh M, et al. A review of indirect time-of-flight technologies[J]. IEEE Transactions on Electron Devices, 2022, 69(6): 2779−2793

[21] Paredes A L, Song Qiang, Conde M H. Performance evaluation of state-of-the-art high-resolution time-of-flight cameras[J]. IEEE Sensors Journal, 2023, 23(12): 13711−13727

[22] Cao Xin, Wang Haoyu, Zhu Qiuquan, et al. PointStaClu: A deep point cloud clustering method based on stable cluster discrimination[J]. Remote Sensing, 2024, 16(13): 2423

[23] Wang Jingyi, Liu Yu, Tan Hanlin, et al. A survey on weakly supervised 3D point cloud semantic segmentation[J]. IET Computer Vision, 2024, 18(3): 329−342

[24] Noichl F, Collins F C, Braun A, et al. Enhancing point cloud semantic segmentation in the data-scarce domain of industrial plants through synthetic data[J]. Computer-Aided Civil and Infrastructure Engineering, 2024, 39(10): 1530−1549

[25] Singh D P, Yadav M. Deep learning-based semantic segmentation of three-dimensional point cloud: A comprehensive review[J]. International Journal of Remote Sensing, 2024, 45(2): 532−586

[26] Han Bing, Zhang Xinyun, Ren Shuang. Survey of convolution operations based on 3D point clouds[J]. Journal of Computer Research and Development, 2023, 60(4): 873−902 (in Chinese)
(韩冰, 张鑫云, 任爽. 基于三维点云的卷积运算综述[J]. 计算机研究与发展, 2023, 60(4): 873−902)

[27] Lü Zhuo, Yao Zhicheng, Jia Yuxiang, et al. A heterogeneous approach for 3D object detection[J]. Journal of Computer Research and Development, 2021, 58(12): 2748−2759 (in Chinese)
(吕卓, 姚治成, 贾玉祥, 等. 3D 物体检测的异构方法[J]. 计算机研究与发展, 2021, 58(12): 2748−2759)

[28] Liao Ruijie, Yang Shaofa, Meng Wenxia, et al. SegGraph: An algorithm for loop-closure detection in outdoor scenes using 3D point clouds[J]. Journal of Computer Research and Development, 2019, 56(2): 338−348 (in Chinese)
(廖瑞杰, 杨绍发, 孟文霞, 等. SegGraph: 室外场景三维点云闭环检测算法[J]. 计算机研究与发展, 2019, 56(2): 338−348)

[29] Zeng Wenhao, Su Gongbing, Su Zixuan, et al. Research on point cloud registration and stitching fusion algorithm based on GCN-PRFNet[J]. IEEE Access, 2025, 13: 43384−43397

[30] Eggen K, Fan Hongchao. WAPooling: An adaptive plug-and-play module for feature aggregation in point cloud classification networks[J]. International Journal of Applied Earth Observation and Geoinformation, 2025, 138: 104439

[31] Zheng Yufei, Lu Jian, Chen Xiaogai, et al. MATNet: Semantic segmentation of 3D point clouds with multiscale adaptive transformer[J]. Computers and Electrical Engineering, 2024, 119: 109526

[32] Li Chongshou, Liu Yuheng, Li Xinke, et al. Deep hierarchical learning for 3D semantic segmentation[J]. International Journal of Computer Vision, 2025, 133(7): 4420−4441

[33] Wu Qiaoyun, Zhang Quanxiao, Tan Chunyu, et al. Point-to-spike residual learning for energy-efficient 3D point cloud classification[C]// Proc of the 38th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2024, 6092−6099

[34] Holz D, Holzer S, Rusu R B, et al. Real-time plane segmentation using RGB-D cameras[C]//Proc of the 15th Robot Soccer World Cup. Berlin: Springer, 2011: 306−317

[35] Poppinga J, Vaskevicius N, Birk A, et al. Fast plane detection and polygonalization in noisy 3D range images[C]//Proc of the 21st IEEE/RSJ Int Conf on Intelligent Robots and Systems. Piscataway, NJ: IEEE, 2008: 3378−3383

[36] Jin Zhi, Tillo T, Zou Wenbin, et al. Robust plane detection using depth information from a consumer depth camera[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 29(2): 447−460

[37] El-Sayed E, Abdel-Kader R F, Nashaat H, et al. Plane detection in 3D point cloud using octree-balanced density down-sampling and iterative adaptive plane extraction[J]. IET Image Processing, 2018, 12(9): 1595−1605

[38] Mouzenidis P, Louros A, Konstantinidis D, et al. Multi-modal variational faster R-CNN for improved visual object detection in manufacturing[C]//Proc of the 18th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2021: 2587−2594

[39] Song Guozhu, Wang Jian, Ma Rongting, et al. Study on the fusion of improved YOLOv8 and depth camera for bunch tomato stem picking point recognition and localization[J]. Frontiers in Plant Science, 2024, 15: 1447855

[40] ABB Robotics. PickMaster twin technical specification[EB/OL]. [2026-01-26].https://www.abb.com/global/en/areas/robotics/products/software/pickmaster-twin

[41] Lim S, Yoo Y, Lee J K, et al. Multi-object RANSAC: Efficient plane clustering method in a clutter[C]//Proc of the 41st IEEE Int Conf on Robotics and Automation. Piscataway, NJ: IEEE, 2024: 3079−3085

[42] Liu Chen, Kim K, Gu Jinwei, et al. PlaneRCNN: 3D plane detection and reconstruction from a single image[C]//Proc of the 30th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 4450−4459

[43] Barath D, Matas J, Noskova J. MAGSAC: Marginalizing sample consensus[C]//Proc of the 30th IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 10197−10205

[44] Liu Chen, Yang Jimei, Ceylan D, et al. PlaneNet: Piece-wise planar reconstruction from a single rgb image[C]//Proc of the 31st IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 2579−2588



Shao Zhichao, born in 2001. Master candidate. Student member of CCF. His main research interests include object detection and artificial intelligence.

邵智超, 2001 年生。硕士研究生，CCF 学生会员。主要研究方向为目标检测、人工智能。



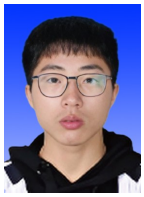
Qiu Jiefan, born in 1984. PhD, associate professor. Member of CCF. His main research interests include embedded operating systems, Internet of things, and artificial intelligence.

邱杰凡, 1984 年生。博士，副教授。CCF 会员。主要研究方向为嵌入式操作系统、物联网、人工智能。



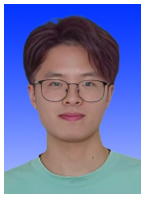
Dai Guanglin, born in 1979. PhD, professor. Member of CCF. His main research interests include intelligent computing, medical image processing, and big data analysis.

戴光麟, 1979 年生。博士，教授。CCF 会员。主要研究方向为智能计算、医学图像处理、大数据分析。



Li Qiaohong, born in 2004. Bachelor. His main research interests include artificial intelligence and object detection.

李乔宏, 2004 年生。学士。主要研究方向为人工智能、目标检测。



Chen Hanmo, born in 2001. Bachelor. His main research interests include machine vision and artificial intelligence.

陈翰墨, 2001 年生。学士。主要研究方向为机器视觉、人工智能。



Xiao Kejiang, born in 1984. PhD, professor. His main research interests include multi-source information fusion, artificial intelligence, intelligent perception interconnection, and Internet of things systems.

肖克江, 1984 年生。博士，教授。主要研究方向为多源信息融合、人工智能、智能感知互联、物联网系统。