

利用点云处理的实时的类别级位姿估计

吴继春¹⁾, 杨永达¹⁾, 张斋武¹⁾, 张平¹⁾, 范大鹏²⁾

¹⁾ (湘潭大学机械工程与力学学院 湘潭 411105)

²⁾ (国防科技大学智能科学学院 长沙 410073)

(wujichun@xtu.edu.cn)

摘要: 针对传统的视觉算法无法准确地测量物体 3D 质心和尺寸、无法检测数据集中未出现过的物体等问题, 提出一种基于点云的类别级物体质心与位姿估计算法。首先通过 2D 分割算法对物体待检测区域进行分割, 并通过 BlendMask 提高实时性; 然后将分割区域生成点云, 并结合先验形状输入到神经网络中重建物体模型; 最后输出一个变形场以及对应矩阵, 并通过 Umeyama 算法对物体进行位姿估计。实验结果表明, 所提算法达到了较高的精度; 在质心估计方面也优于包围盒中心作为质心的算法; 实时速率达到了 13 帧/s。在 GPU NVIDIA 2060 上进行仿真实验, 证明了所提算法计算的质心优于对比算法, 以及该算法的准确性及可靠性。

关键词: 点云; 类别级; 质心确定; 位姿估计

中图分类号: TP391.41

DOI: 10.3724/SP.J.1089.2024.19985

Real-Time Category-Level Pose Estimation Utilizing Point Cloud Processing

Wu Jichun¹⁾, Yang Yongda¹⁾, Zhang Zhaiwu¹⁾, Zhang Ping¹⁾, and Fan Dapeng²⁾

¹⁾ (School of Mechanical Engineering, Xiangtan University, Xiangtan 411105)

²⁾ (College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073)

Abstract: Aiming at the problems that traditional vision algorithms cannot accurately measure the 3D center of mass and size of an object, and cannot detect the objects that have not appeared in the dataset, we propose a category-level object center of mass and position estimation method based on point cloud. Firstly, a 2D segmentation algorithm is used to segment the region to be detected, and BlendMask is used to improve the real-time performance; then, a point cloud is generated from the segmented region and combined with the a priori shapes to reconstruct the object model in a neural network; finally, a deformation field and its corresponding matrix are outputted, and the position of the object is estimated by the Umeyama algorithm. The experimental results show that the proposed method achieves high accuracy, outperforms the center-of-envelope-box algorithm in center-of-mass estimation, and achieves a real-time rate of 13 FPS. Simulation experiments on a GPU NVIDIA 2060 demonstrate that the center-of-mass of the proposed method outperforms the comparative methods, and that the method is both accurate and reliable.

Key words: point cloud; category level; the center of mass to determine; pose estimation

收稿日期: 2022-09-23; 修回日期: 2023-03-31. 基金项目: 中国湖南联合区域创新基金(U19A2072); 湖南省自然科学基金(2021JJ30678). 吴继春(1979—), 男, 博士, 教授, 硕士生导师, 主要研究方向为机器视觉与运动控制、PLCOPEN 运动控制; 杨永达(1999—), 男, 硕士研究生, 主要研究方向为机器视觉; 张斋武(1999—), 男, 硕士研究生, 主要研究方向为机器人运动轨迹规划; 张平(1998—), 男, 硕士研究生, 主要研究方向为 PLCOPEN 运动控制; 范大鹏(1964—), 男, 博士, 教授, 博士生导师, 主要研究方向为超精密测量、机床运动误差分析、运动控制、特种装备等。

随着深度学习的发展, 2D 目标检测已经应用于物体分拣中, 但是基于视觉分拣的方法一般是通过处理 2D 图像^[1]来检测物体的质心并对物体进行分拣. 郑哲等^[2]提出的多目标检测算法, 首先利用骨干网络提取多层特征作为多尺度注意力特征金字塔网络(multi-scale attention feature pyramid network, MAFPN)的输入, MAFPN 结合特征融合和注意力机制, 在融合多个尺度的特征映射的基础上, 使用通道域注意力和空间域注意力机制来抑制噪声, 增强有效特征的复用, 提高网络对目标多尺度特征的自适应性.

杨旭升等^[3]提出实例级位姿估计方法, 采用基于最小二乘的点云平面法向量估计方法实现药盒的位姿估计. 实验采集存在视点变化、旋转变换和尺度变化的 600 幅图像进行测试, 实验结果表明药盒位姿估计速度可达 11 帧/s, 定位偏差小于 5 mm, 姿态估计偏差在 2°以内. 这种方法虽然可以很好地估计位姿, 但是无法满足本文提出的分拣要求, 特别是对于一些不规则物体, 每个待检测的物体的信息都是未知的, 采用实例级算法根本无法完成任务. 因此, 为了完成分拣任务, 本文采用了类别级的位姿估计算法, 可以检测数据集中没有的对象实例, 即可以检测同一类别未见过物体.

已有的一些类别级算法中, Wang 等^[4]使用 Umeyama 的算法导出归一化的对象坐标空间(normalized object coordinate space for category-level 6D object pose and size estimation, NOCS)并恢复姿态, 但该方法的精度并不高, 没有显式地考虑类内形状的变化. 而 GPV-Pose(category-level object pose estimation via geometry-guided point-wise voting)^[5]采用平均尺寸作为先验形状进行位姿估计, 首先将要检测的物体分割出来; 从反投影的 3D 点云中采样 1 028 个点, 并将它们输入到位姿网络中去; 然后, 提取全局和点的特征; 此外, 还附加了 3 个并行分支, 用于直接置信度感知的位姿预测、对称感知的点云重建和逐点包围盒投票. 利用物体的几何特性很好地提高了检测准确度. 但是此方法泛化能力不强, 对于一些数据集以外的场景检测效果不佳; 且因采用 Mask R-CNN(mask region-based convolutional neural network)^[6]进行 2D 分割, 使得检测速率仅为 8 帧/s (NVIDIA 2060), 无法满足实时的检测. 本文方法解决了这些问题, 达到了泛化性和实时速率的要求.

本文通过输入单幅 RGBD 图像预测物体 3D 信息, 包括对物体进行位姿估计、分类和质心确定,

避免了传统方法通过六边形包围盒确定质心偏差较大、无法测量未见过物体尺寸等问题; 提出一种类别级数据集的快速制作方法; 解决了已知类别未见过对象的物体的质心与位姿估计问题.

1 基于 CAMERA 的快速构建数据集的方法

针对类别级数据集制作困难、制作成本高、制作复杂等问题, 本文提出了一种类别级数据集的快速制作方法. 将 Wang 等^[4]提出的 CAMERA (context-aware mixed reality)数据集作为一个库, 其中包括瓶子、碗、相机、罐子、笔记本电脑和马克杯 6 类物品; 同时, 利用了点云的无标签性以及无序性, 从 CAMERA 数据集中选取与要检测物体形状类似的类别作为先验形状.

首先, 对要检测物体从 CAMERA 数据集中选取类别, 然后通过 2D 分割的方法对数据集进行制作. 使用 labelme 对数据集打标签, 并通过 BlendMask^[7]对数据集进行训练. 选取香菇作为示例, 从 CAMERA 中选取碗作为香菇头部的类似形状, 选取瓶子作为香菇尾部的类似形状. 具体制作流程如图 1 所示. 图中使用 labelme 进行数据集标注, 文件夹结构按照图中所示进行分布, 即可以通过 BlendMask 对数据集进行训练.

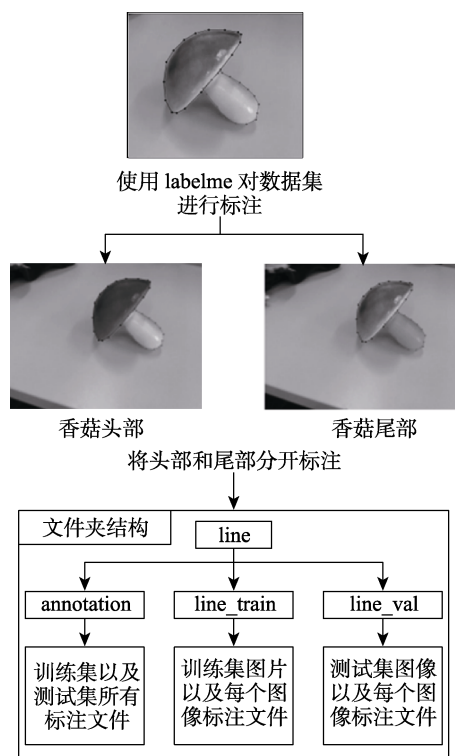


图 1 数据集制作流程图

2 位姿估计算法

(1) 将图像进行裁剪, 生成一幅图像面片, 将深度图与图像面片结合生成观测到的点云. 同时通过自动编码器得到一个平均先验形状. (2) 将平均先验形状、图像面片和观测到的点云输入网络中, 生成一个变形场; 通过变形场将先验形状变形得到重建的物体的点云, 再将重构后的点云与对应矩阵相乘得到 NOCS 下的坐标, 并通过 Umeyama^[8]算法计算观测到的点云与重建后的点

云的比例、旋转矩阵以及平移矩阵. (3) 选取重建后的模型的 9 个特征点(8 个包围盒的顶点和 1 个质心)并将其变换至观测到的实例上去, 计算出物体的位姿和质心, 从而确定物体的 3D 信息. 图 2 所示为位姿估计算法的流程图. 其中, C 表示图像分割和裁剪图像面片, S 表示图像与深度图合成点云, + 表示矩阵加法, $\times$ 表示矩阵乘法, U 表示用 Umeyama 算法计算出 2 个点云之间的比例、旋转以及平移矩阵, T 表示对 9 个特征点进行缩放、平移和旋转变换.

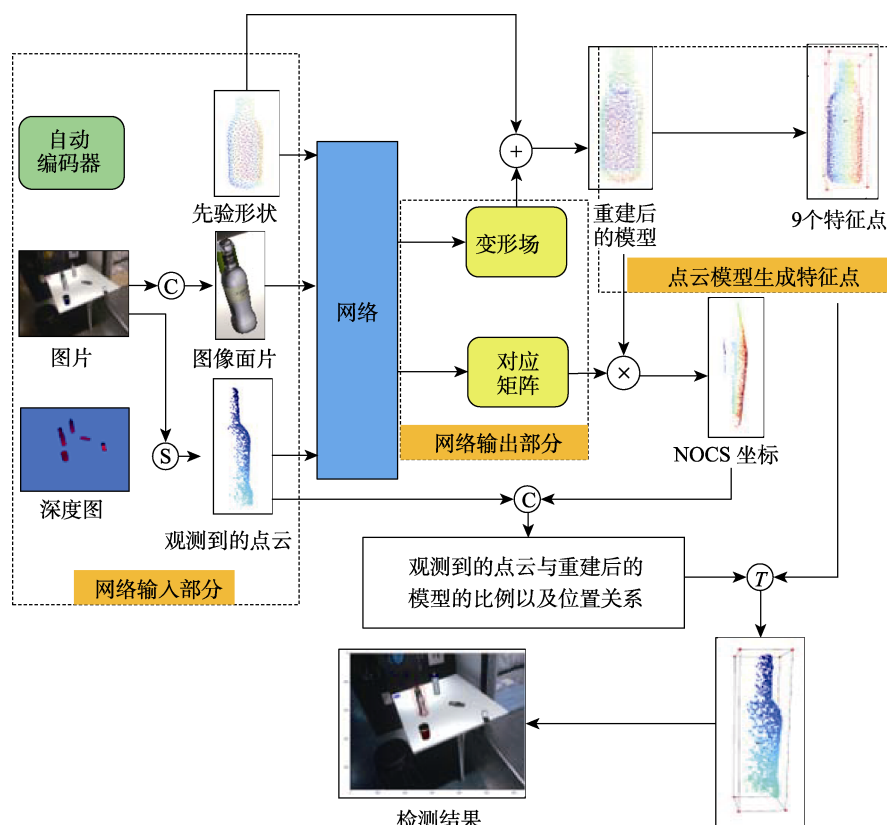


图 2 位姿估计算法流程图

2.1 生成平均先验形状

由于本文采用预先生成平均先验形状的方法, 因此需要一个编码器来自动生成平均先验形状, 本文采用自动编码器^[9]. 输入为 $n \times 3$ ($n \times 1 \times 3 \times 1$, 点数 $\times$ 长 $\times$ 宽 $\times$ 通道数), 共有 64 个 1×3 的卷积核, 每个卷积核对点云进行卷积操作, 并生成一个通道, 则 64 个卷积核共计生成 64 个通道; 得到的结果为 $n \times 64$ ($n \times 1 \times 1 \times 64$, 点数 $\times$ 长 $\times$ 宽 $\times$ 通道数)成功将数据升维到 64 通道. 再将 $n \times 64$ 与 128 个 1×1 的 64 通道卷积核相乘, 将 64 维升到 128 维. 通过共享多层感知机(shared multi-layer perceptron, shared MLP)输出为 $n \times 128$; 然后经过最大池化,

将 n 个具有 128 维特征的数据转化为一个 1×128 的全局特征向量. 最后, 将全局信息附在每一个局部点描述的后面, 形成 $n \times 256$ 的向量. 经过相同的操作, 通过 shared MLP 和最大池化, 将数据输出为 1×1024 的全局特征向量; 再通过全连接层输出 512 个点云分类值; 之后通过一个解码器, 包括 3 个全连接层, 输出 $n \times 3$ 的数据. 自动编码器的流程如图 3 所示.

形状集合 $M = \{M_c^i | i=1,2,\dots,N; c=1,2,\dots,C\}$, 其中, M_c^i 是来自类别 c 的实例 i 的 3D 点云模型. 本文独立地对每个模型应用相似性变换, 使得它在 NOCS

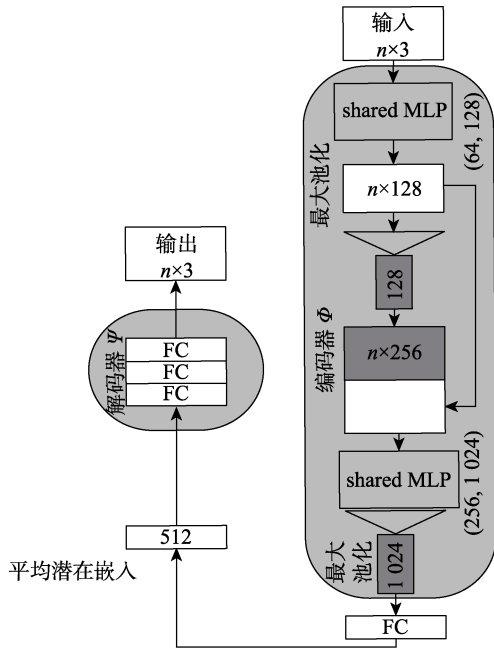


图 3 自动编码器的流程

中适当地对齐, 以确保学习的形状先验具有与要重建的目标形状相同的比例和方向. 编码器 Φ 获取点云并输出低维特征向量, 即潜在嵌入 $Z_c^i \in \mathbb{R}^n$. 解码器 Ψ 获取该特征向量并输出重建的点云

$$\bar{M}_c^i = (\Psi \circ \Phi)(M_c^i) = \Psi(Z_c^i) \quad (1)$$

重建后的误差由倒角距离(chamfer distance, CD)测量, 公式为

$$d_{CD}(M_c^i, \bar{M}_c^i) = \sum_{x \in M_c^i} \min_{y \in \bar{M}_c^i} \|x - y\|_2^2 + \sum_{y \in \bar{M}_c^i} \min_{x \in M_c^i} \|x - y\|_2^2 \quad (2)$$

通过最小化误差使自动编码器对形状集合训练生成平均形状, 一旦训练收敛, 就得到了 M 中所有实例的潜在嵌入 Z_c^i ; 然后计算每个类别的平均潜在嵌入后, 通过解码器来构造形状先验, 公式为

$$M_c = \Psi(Z_c) = \Psi\left(\frac{1}{N_c} \sum_i Z_c^i\right) \quad (3)$$

图 4 所示为这个自动编码器生成的平均先验形状, 至此, 为需要检测的对象预先生成了一个平均先验形状.

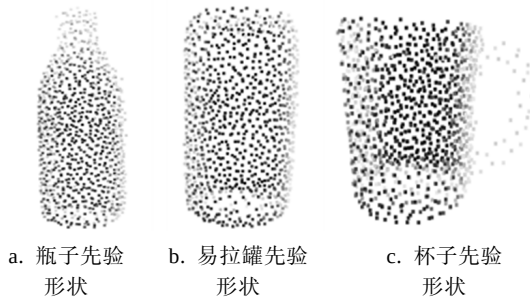


图 4 自动编码器生成的先验形状

2.2 实例物体的点云获取

本文采用 RGBD 相机结合 BlendMask 算法对图像进行分割, 生成观测到的实例物体的点云. 从深度图获取点云, 即从像素坐标系下的点 u, v 到相机坐标系下的点 x_c, y_c, z_c .

将图像中的 2D 信息转换为 3D 信息时, 涉及相机小孔成像原理. 需要经过图像坐标系($XO_I Y$), 相机坐标系($X_c O_c Y_c$), 完成 2D 坐标系到 3D 坐标系之间的转换. 总体转换关系如图 5 所示.

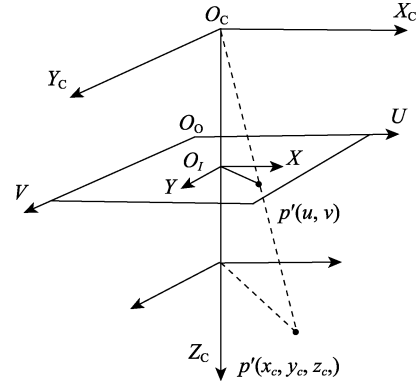


图 5 坐标总体转换图

2.2.1 像素坐标系到物理坐标系

如图 6 所示, O_I 是图像物理坐标系的中点, 位于像素坐标系(U, V)的中点, 设 O_I 的值为 (u_0, v_0) ; $XO_I Y$ 是图像坐标系, 其 X 轴与 Y 轴都与 U 轴与 V 轴平行. 设单位像素点在 X 轴上的真实物理长度为 k , 在 Y 轴上的真实物理长度为 l , 图像像素坐标点的位置可以看作基于物理坐标的平移, 因此可以得到点 p' 在图像物理坐标系下的坐标 (x_u, y_u) 为

$$\begin{cases} u = \frac{x_u}{k} + u_0 \\ v = \frac{y_u}{l} + v_0 \end{cases} \quad (4)$$

运用线性代数中换将 n 维值向量以 $n+1$ 维向量的表示方法, 将式(4)转换成矩阵形式为

$$\begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k} & 0 & -u_0 k \\ 0 & \frac{1}{l} & -v_0 l \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (5)$$

2.2.2 物理坐标系到相机坐标系

相机坐标系是描述在相机的视角下目标物体的角度和位置. 如图 5 所示 $X_c O_c Y_c$ 为相机坐标系.

原点 O_c 在相机的光轴 Z_c 上; 相机坐标系的 X_c 轴和 Y_c 轴分别与图像坐标的 U 轴和 V 轴平行,

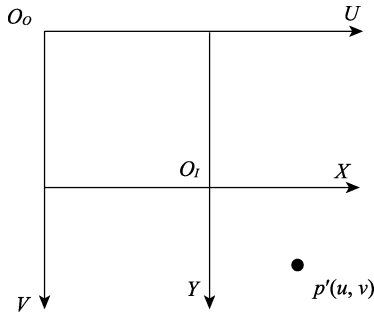


图6 图像像素坐标系与物理坐标系关系图

光轴 Z_c 垂直于图像平面且与相机平面的交点垂直, 即图像物理坐标的原点 O_I ; 相机原点 O_C 到图像坐标原点 O_I 的距离为相机的焦距, 焦距 $O_C O_I$ 用 f 表示. 当已知点 p' 在图像坐标系下的坐标 (x, y) , 焦距 f 时, 由三角形相似定理可得点 p' 在相机坐标系下的坐标 (x_c, y_c, z_c) 为

$$\begin{cases} x_u = x_c \cdot \frac{f}{z_c} \\ y_u = y_c \cdot \frac{f}{z_c} \end{cases} \quad (6)$$

用齐次矩阵将式(6)变换为

$$z_c \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (7)$$

其中, x_c, y_c, z_c 表示每个点在相机坐标系下的坐

标. 至此就完成了从图像结合深度图生成观测到的点云.

2.3 生成变形场以及对应矩阵的网络部分

网络部分采用 Tian 等^[10]提出的网络, 输入为图像、点云和先验形状, 输出为对应矩阵和变形场. 第1分支输入一幅 $H \times W \times 3$ 的图像, 经过卷积神经网络生成 $H \times W \times 64$ 的数据, 然后规范为 $n \times 64$ 的数据; 第2分支输入一组 $n \times 3$ 的点云数据, 经过 shared MLP 生成 $n \times 64$ 的数据; 将一个 $n \times 64$ 的局部特征附在另一个 $n \times 64$ 的局部特征后面, 生成一个 $n \times 128$ 的局部特征; 经过 shared MLP 和平均池化生成一个 1×1024 的全局特征; 再将 I 和 V 生成的全局特征与 M_c 生成的全局特征附在 $n \times 128$ 的局部特征后, 生成了一个 $n \times 2176 (1024 + 1024 + 128)$ 的向量, 经过 shared MLP 和 Softmax 回归输出一个 $n \times m$ 的对应矩阵. 在输出先验形状部分, 输入 $m \times 3$ 的数据, 经过 shared MLP 生成 $m \times 64$ 的局部特征, 通过 shared MLP 和平均池化生成一个 1×1024 的全局特征; 再将 I 和 V 生成的全局特征与 M_c 生成的全局特征附在 $m \times 64$ 的局部特征后, 生成一个 $n \times 2112 (1024 + 1024 + 64)$ 的向量; 最后, 经过 shared MLP 输出一个 $m \times 3$ 的变形场. 网络部分流程图如图7所示. 至此, 通过将图像、点云和先验形状输入到网络中, 输出了一个对应矩阵和一个变形场.

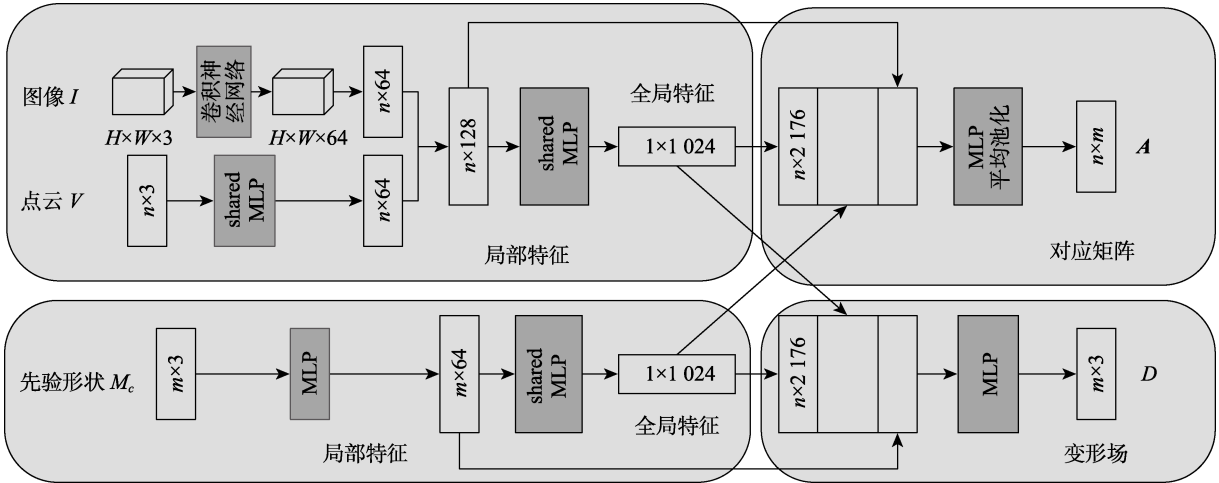


图7 生成变形场以及对应矩阵的网络部分

2.4 点云质心的计算

已有的估计位姿抓取物体质心的方法多采用包围盒的中心作为质心, 但会存在很大的偏差, 在抓取时甚至导致物体掉落, 或者可能因为物体形状导致质心落在物体之外. 存在的偏差将对物体

抓取产生很大的影响, 可能导致抓取失败.

针对此问题, 本文提出使用物体完整的点云数据计算物体的质心, 再将质心进行旋转平移到实例上, 得到实例物体的质心. 该方法很好地提高了抓取物体的准确度, 避免一些抓取失败的情形,

实用性更强。

2.5 3D 点云的配准方法

Huang 等^[11]和 Qin 等^[12]提出的点云配准方法准确度很高,但是由于它们是对每个点进行配准,计算量很大,可能会花费较长时间,而且需要准备的数据量也比较大。因此,本文采用只计算 8 个包围盒顶点和质心的方法。

本文在进行点云配准时,采用 Umeyama 算法计算 NOCS 坐标下的特征点云和观测到的点云的缩放系数、旋转矩阵和平移矩阵。对重建后的点云最小包围盒 8 个顶点和观测到的点云的 9 个点,同时使用 RANSAC (random sample consensus)算法^[13]去除噪点,转换结果如图 8 所示。

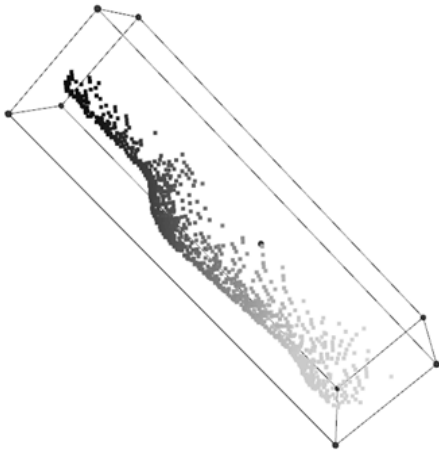


图 8 观测到的点云 9 个特征点图

Umeyama 算法的目标是计算一组 $\mathbf{R}, \mathbf{t}$ 使目标函数最优,公式为 $\frac{1}{n} \sum_{i=1}^n \|q_i - (c\mathbf{R}p_i + \mathbf{t})\|_2^2$ 。可以看出,函数最优的判断标准是距离的平方和。最终的计算结果为

$$\mathbf{T} = \begin{bmatrix} c\mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (8)$$

由重建后点云经过缩放计算出的 9 个特征点变换到观测到的点云的坐标变换公式为

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \mathbf{t} \quad (9)$$

其中, x_i, y_i, z_i 表示重构后的模型的包围盒 8 个顶点和质心; x_p, y_p, z_p 表示观测到的点云的包围盒 8 个顶点和质心; $\mathbf{R}$ 表示旋转矩阵, $\mathbf{t}$ 表示平移矩阵。

2.6 图像检测结果的可视化

观测到的点云在 3D 空间下获得 9 个特征点后,需要通过相机坐标系到图像坐标系的变换,才能

在图像上画出包围盒和质心,逆变换为

$$\begin{cases} u = \frac{x_c \cdot f}{k \cdot z_c} + u_0 \\ v = \frac{y_c \cdot f}{l \cdot z_c} + v_0 \end{cases} \quad (10)$$

其中, u 和 v 表示像素坐标系下的坐标,完成了从世界坐标系到像素坐标系的转换。

3 实验及结果分析

3.1 本文算法实验

预先训练目标物体的权重文件。使用 Ubuntu18.04 系统, NVIDIA 2060 显卡进行 GPU 加速,一共检测 19 220 幅图像,共花费时间 1 335.76 s,因此图像的平均检测的速度是 13 帧/s。

选取 4 幅图像,包括瓶子和杯子 2 个类别,应用本文算法对图像中的物体进行检测。选取了 2 个类别,分别是瓶子以及杯子。其可视化结果如图 9 所示,通过将 8 个包围盒的顶点按顺序连接起来,并且画出质心所在位置。图中方框表示物体的位姿估计结果,点表示物体的质心估计。



图 9 实例物体检测结果

3.2 与 NOCS 的对比

在 REAL-275 数据集上,采用本文算法和 NOCS 算法^[4]对给定类别中不同的和不可见的对象实例进行实验,结果如表 1 所示。其中,第 2~4 列表示数据集中检测到的 3D 边界框与边界框真值分别大于 25%, 50%和 75%重合度的比例;第 5~7 列表示在 6 个类别位姿分别使用(5°, 2 cm), (5°, 5 cm), (10°, 5 cm)度量的平均检索精度(mean average precision, MAP)。首先导出 NOCS 并恢复姿态,然后训练基于区域的神经网络,以直接推断从观察到的像素到该共享对象表示(NOCS),以及其他对象

信息的对应关系.可以看出,这些预测可以与深度图相结合,联合估计混乱场景中多个对象的度量6D姿态和维度.NOCS还提出了一种新的上下文感知技术来生成大量完全注释的混合现实数据.

表1 2种算法在REAL-275数据集上的对比

算法	3D ₂₅	3D ₅₀	3D ₇₅	5°		10°
				2 cm	5 cm	5 cm
NOCS ^[4]	84.9	80.5	30.1	7.2	10.0	25.2
本文	83.4	77.3	53.2	19.3	21.4	54.1

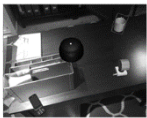
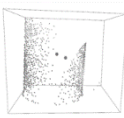
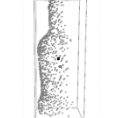
注:粗体表示最优值.

3.3 与GPV-Pose对比

3.3.1 质心的对比

GPV-Pose算法^[5]将最小包围盒的8个顶点的中心作为质心,而本文算法是用检测到的物体重建得到点云模型确定质心.在抓取物体时质心的确定很重要,虽然这对于方正的物体来说可能影响不大,但对于一些非对称的物体或者是一些特殊结构的物体,质心的确定对于抓取结果是否可靠影响很大.2种算法对于2个物体的质心检测结果如表2所示.从表2可以看出,对于杯子这种带“把手”的特殊物体,本文算法的优势更加明显,对于一些特殊的结构,如杯子把手过长,本文算法可以很准确地对物体进行抓取.

表2 2个物体质心检测结果对比

检测对象	检测图像	杯子点云模型	观测到的点云包围盒和质心
杯子			
瓶子			

2种算法抓取物体时,质心之间的距离对比如表3所示.可以看出,物体结构越复杂,2种算法计算出来的质心之间的差距就越大.

表3 2种算法质心之间的距离比 mm

检测对象	2种算法质心之间的距离
杯子	9.8
瓶子	0.8
平均	5.3

3.3.2 检测速度对比

采用NVIDIA 2060显卡加速的情况下,2种算法的实时检测速度对比如表4所示.可以看出,GPV-Pose算法不能达到实时检测的要求,画面存在卡顿;本文算法可以达到实时检测的要求.

表4 2种算法实时检测速度比较 帧/s

算法	实时检测速度
GPV-Pose ^[5]	8
本文	13

3.3.3 本文数据集制作方法对比

选取香菇作为实验对象,根据本文提出的数据集制作方法,从CAMERA中选取碗作为香菇头部的类似形状,瓶子作为香菇尾部的类似形状.从同一个视角将2种方法进行多组实验,结果如图10所示.其中,上、下2个点表示香菇头部和尾部2部分各自的质心;中间点表示整个物体的质心.可以看出,本文方法能够更好地估计位姿,生成的包围盒更加贴合实际物体,表明其在质心检测方面也更加准确.同时,验证了本文提出的数据集制作方法可行,在泛化能力方面也更优.

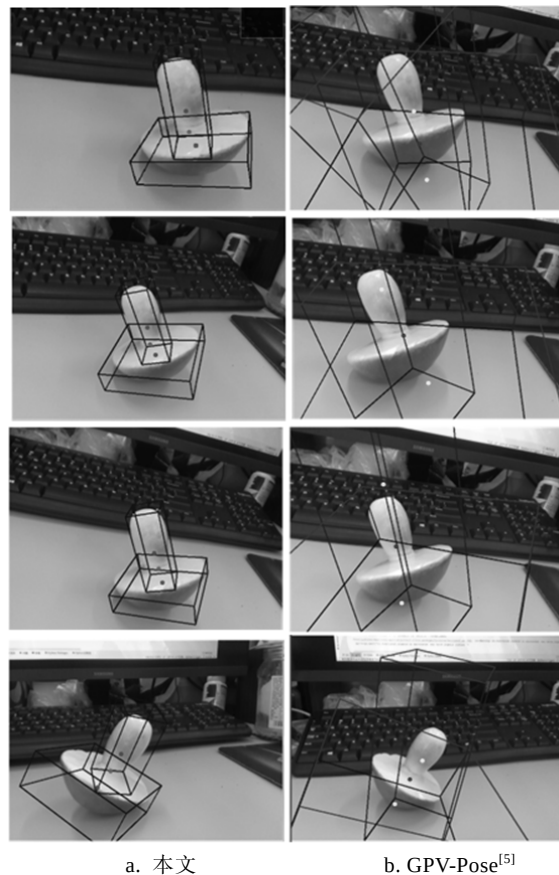


图10 2种算法在视频流中检测效果对比

4 结 语

针对同一类别的未见过物体, 本文提出一种通过 RGB 和深度信息估计类别级物体的 3D 质心和位姿估计方法, 能够实现对不同类别的物体进行分类检测, 并且可以对同一类别未见过物体进行检测. 实验结果表明, 本文方法对同一类别未见过物体具有较好的性能; 采用 BlendMask 进行 2D 分割, 在 NVIDIA 2060 显卡加速情况下, 实时检测速度达到 13 帧/s, 证明了该方法对质心和位姿估计检测的有效性和高效性; 与现有方法进行对比, 该方法检测速度和精度都能够达到分拣的要求, 且泛化能力更强. 但本文方法实时性还可以进一步提高, 接下来将对网络进行剪枝, 以提高实时性, 同时可以结合机械臂来对物体进行抓取, 进一步验证本文方法的精度.

参考文献(References):

- [1] Xie Yuhong, Xie Yuan, Chen Liang, *et al.* Object detection in real-world hazy scene[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2021, 33(5): 733-745(in Chinese)
(解宇虹, 谢源, 陈亮, 等. 真实有雾场景下的目标检测[J]. *计算机辅助设计与图形学学报*, 2021, 33(5): 733-745)
- [2] Zheng Zhe, Lei Lin, Sun Hao, *et al.* FAGNet: multi-scale object detection method in remote sensing images by combining MAFPN and GVR[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2021, 33(6): 883-894(in Chinese)
(郑哲, 雷琳, 孙浩, 等. FAGNet: 基于 MAFPN 和 GVR 的遥感图像多尺度目标检测算法[J]. *计算机辅助设计与图形学学报*, 2021, 33(6): 883-894)
- [3] Yang Xusheng, Wang Shuaiyang, Zhang Wen'an, *et al.* Research on pose estimation method of medicine box based on secondary key point matching[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2022, 34(4): 570-580(in Chinese)
(杨旭升, 王帅扬, 张文安, 等. 基于二次关键点匹配的药盒位姿估计方法[J]. *计算机辅助设计与图形学学报*, 2022, 34(4): 570-580)
- [4] Wang H, Sridhar S, Huang J W, *et al.* Normalized object coordinate space for category-level 6D object pose and size estimation[C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos: IEEE Computer Society Press, 2019: 2637-2646
- [5] Di Y, Zhang R D, Lou Z Q, *et al.* GPV-pose: category-level object pose estimation via geometry-guided point-wise voting[C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos: IEEE Computer Society Press, 2022: 6771-6781
- [6] He K M, Gkioxari G, Dollár P, *et al.* Mask R-CNN[C] // *Proceedings of the IEEE International Conference on Computer Vision*. Los Alamitos: IEEE Computer Society Press, 2017: 2980-2988
- [7] Chen H, Sun K Y, Tian Z, *et al.* BlendMask: top-down meets bottom-up for instance segmentation[C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos: IEEE Computer Society Press, 2020: 8570-8578
- [8] Umeyama S. Least-squares estimation of transformation parameters between two point patterns[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991, 13(4): 376-380
- [9] Qi C R, Su H, Mo K, *et al.* PointNet: deep learning on point sets for 3D classification and segmentation[C] // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos: IEEE Computer Society Press, 2017: 77-85
- [10] Tian M, Ang Jr M H, Lee G H. Shape prior deformation for categorical 6D object pose and size estimation[C] // *Proceedings of the 16th European Conference on Computer Vision*. Heidelberg: Springer, 2020: 530-546
- [11] Huang S Y, Gojcic Z, Usvyatsov M, *et al.* PREDATOR: registration of 3D point clouds with low overlap[C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos: IEEE Computer Society Press, 2021: 4265-4274
- [12] Qin Z, Yu H, Wang C J, *et al.* Geometric transformer for fast and robust point cloud registration[C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos: IEEE Computer Society Press, 2022: 11133-11142
- [13] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. *Communications of the ACM*, 1981, 24(6): 381-395