

基于点云的果园环境苹果定位技术研究

汪一昕, 贺磊盈

浙江理工大学机械工程学院, 浙江 杭州

收稿日期: 2023年2月26日; 录用日期: 2023年5月3日; 发布日期: 2023年5月10日

摘要

复杂枝叶遮挡的果园环境中的苹果准确检测和定位对于苹果采摘的成功率至关重要。本研究提出了一种目标检测网络和点云处理的方法, 用于检测遮挡和未被遮挡的苹果以及计算果实中心及主轴。使用深度相机获取RGB彩色图和深度图像。使用YOLOv5s目标检测网络检测苹果果实, 并获取目标点云, 通过点云特征以及主成分分析法计算苹果的中心坐标位置; 利用最优化方法计算苹果的果轴方向。最后为验证所提出方法的有效性, 采用VREP仿真平台部署苹果、果树枝叶遮挡和相机, 验证该算法的有效性和稳定性, 并在实地果园场景中部署算法和果实采摘机器人, 验证了算法在实际场景中的有效性和鲁棒性。

关键词

苹果位姿计算, 点云处理, YOLOv5s, VREP仿真

Research on Apple Positioning Technology in Orchard Environment Based on Point Cloud

Yixin Wang, Leiying He

School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou Zhejiang

Received: Feb. 26th, 2023; accepted: May 3rd, 2023; published: May 10th, 2023

Abstract

Accurate detection and location of apples in orchard environment with complex branches and leaves is very important for the success rate of apple picking. In this study, a target detection network and point cloud processing method is proposed, which are used to detect occluded and unobstructed apples and calculate the fruit center and main axis, using depth camera to obtain RGB color map and depth image. The YOLOv5s target detection network is used to detect the apple

fruit, and the target point cloud is obtained, and the central coordinate position of the apple is calculated by point cloud characteristics and principal component analysis, using the optimization method to calculate the fruit axis direction of apple. Finally, in order to verify the effectiveness of the proposed method, the VREP simulation platform is used to deploy apple and fruit tree branches and cameras to verify the effectiveness and stability of the algorithm, and the algorithm and fruit-picking robot are deployed in the field orchard scene to verify the effectiveness and robustness of the algorithm in the actual scene.

Keywords

Apple Pose Calculation, Point Cloud Processing, YOLOv5s, VREP Simulation

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,随着人口老龄化的发展,水果种植园劳动力严重不足,而苹果采摘窗口期短等问题进一步加剧了劳动力短缺带来的影响,因此由苹果收获机器人机械自动化采摘苹果成为了未来的趋势。苹果收获机器人的视觉定位算法是核心模块,采摘机械臂需要通过视觉算法的引导来实现对果实的定位摘取,其相关技术的研究在国内外已开展多年。

世界各研究团队开发了针对各种蔬果的定位算法,日本团队的 Yuki Onishi 等人设计了一款苹果收获机器人的果实定位算法[1]。该算法采用 SSD 作为目标检测网络,并通过立体摄像机来获取苹果坐标。Plebe 和 Grasso 开发了一款基于立体摄像机的定位算法[2]。通过立体摄像机处理的数据来匹配橙子的质心实现定位。Williams H.A.M.等人研究了猕猴桃定位算法[3],该算法通过双目视觉系统来检测水果并在三维空间中确定它们的位置。Kondo 等人开发了利用激光测距仪来定位的蔬果定位算法[4]。Christopher Lehnert 等人通过 RGBD 相机获取甜椒的点云,并于预设的超椭球体拟合到分割的甜椒上以获得甜椒的姿态信息[5]。Hanwen Kang 等人通过获取苹果点云的大致朝向来估计苹果姿态并规划机械臂采摘路径[6]。Hao Li 等人通过分析甜椒的点云特征来获取甜椒果实的对称平面和主轴以估计其姿态[7]。Marcus Gualtierid 设计了一个可以通过 RGB 图估计水果姿态的卷积神经网络[8]。赵静[9]等人使用结构开运算对苹果果梗位置进行判别。姚立健[10]等人通过计算果实的惯性主轴的方法计算水果的主轴信息。此类定位算法采用测距、模型配准、分析几何轮廓等方法来计算果实中心或主轴,其虽在理想环境下有较明显的效果,但此类算法对果实的生长姿态、果实的表面完整度都有较高的要求,在实际果园的复杂遮挡环境中,获取到的数据往往是姿态各异且表面由于枝叶遮挡导致残缺的苹果,这些定位算法的性能往往会受到大幅度影响甚至直接威胁到算法的有效性。

在实际苹果果园环境中,果实数量多且生长姿态多样,部分苹果在倾斜的同时会在相机视角下露出果梗位置,同时果实与果实、枝叶之间都可能出现重叠形成遮挡,这对常规的定位方法造成挑战。苹果果实生长在茎干上,不同的采摘方式对采摘效果影响很大,错误的采摘不仅需要较大的力,并且会对苹果表面产生严重损伤。成功率最高的采摘方式是机械爪顺着苹果的果轴方向,握住苹果果实旋转后摘取,这样果实从茎干上脱落需要的力最小,采摘成功率最高,且机械爪遇到果树枝条干涉的几率也最小,因此苹果果实主轴方向的计算至关重要。要实现苹果在果园环境下的采摘则需要开发出一套能在被遮挡和

未被遮挡条件下的果实中心定位、主轴估计算法。

本文聚焦于果园的实际场景, 目标苹果受到枝叶遮挡, 树叶、枝条对果实检测、定位、采摘的干涉明显。本文采用 YOLOv5 深度神经网络对 RGB 图像中的果实目标进行检测, 利用深度相机提取果实目标区域的三维表面点云, 用分割算法获得果实的表面点云, 对点云进行处理从而计算果实中心和主轴。使用 VREP 仿真平台搭建果树模型, 获取虚拟环境下被遮挡苹果的点云, 用于验证该算法的精度, 并用实验验证了算法的有效性。

2. 果实检测

目标检测网络 YOLOv5

为在果园的复杂环境下实现苹果的自主定位, 需要能在 RGB 图中获得目标苹果的检测框。由于苹果颜色各异(主要为红绿两种)且遮挡严重, 故采用深度卷积神经网络算法对苹果目标进行检测。

YOLOv5 目标检测网络由输入端、主干网络、Neck、预测输出层四个部分组成。其中输入端采用了 Mosaic 数据增强、自适应锚框计算、自适应图片缩放等算法, 提升了后续的推理精度、速度。主干网络采用了 Focus、CSP 结构, Neck 部分采用 FPN + PAN 的结构。输出端 Bounding box 损失函数则采用 CIOU_Loss, 针对后续的目标框筛选采用 NMS 非极大值抑制方法来进行。

YOLOv5s 目标检测网络模型轻量化程度高, 速度快, 对于果实目标的检测可维持较高的精度。本文采用 Pytorch 框架下的 YOLOv5s 模型进行训练, 数据集由 1824 张实地场景采集的苹果图片采用数据增强扩充 4 倍共 7296 张图片组成。

3. 果实位姿计算

苹果位姿计算需要用到果实侧表面点云数据, 通过 YOLOv5 网络获取到的果实目标框信息, 利用深度相机获取目标点云, 点云数据包含果实表面点云、遮挡物(果树枝叶)点云、背景物点云等。为去除多余点云的干扰, 提取出果实侧表面点云信息, 首先通过直通滤波将点云中距离镜头过远的背景点云滤除; 接着, 通过体素滤波器来下采样降低点云团密度; 然后, 计算果实表面点云中每个点的曲率, 利用曲率阈值分割, 将曲率过大的点滤除实现去除果实表面果梗果柄部分点云以及遮挡物与果实表面连接处点云; 利用分割算法将遮挡物点云分离, 获得苹果果实侧表面点云; 最后进行苹果果实中心、主轴计算。果实位姿计算流程如图 1 所示:

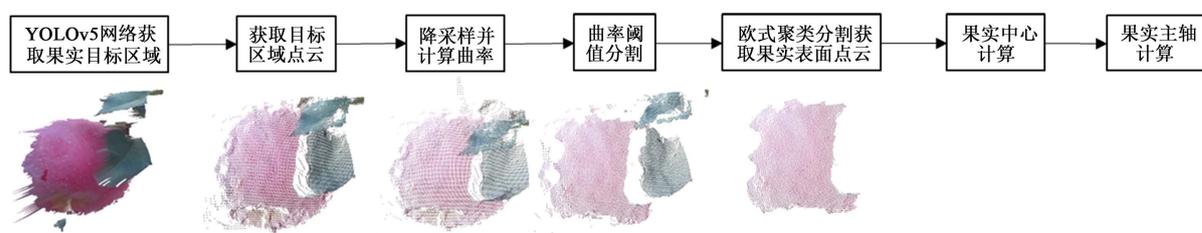


Figure 1. Flow chart of apple pose calculation

图 1. 苹果位姿计算流程图

3.1. 苹果侧表面点云的获取

在某些情况下, 枝叶等遮挡物会直接附着在苹果果实表面, 造成遮挡物点云与果实表面点云直接相连。由于苹果果实果柄、果梗部分、遮挡物与果实表面接触部分弯曲幅度较大, 而果实侧表面弯曲弧度较小, 故可通过计算果实表面点云中每个点的曲率。利用曲率阈值分割, 将曲率高于阈值的点滤除即可

去除果梗、果柄部分点云以及遮挡物与果实表面连接处的点云。首先计算每个点的法向量和曲率, 法向量可由主成分分析(PCA)方法计算, 果实表面点云中每个点的曲率 δ 可由以下公式近似描述:

$$\delta = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (1)$$

其中 λ_0 、 λ_1 、 λ_2 分别为 PCA 算法中获得的三个从小到大排序的特征值。

经过以上处理后, 枝叶遮挡物、果实侧表面的点云之间可能出现空隙, 可利用不同点云块之间的空隙来完成分割。使用基于欧式距离的聚类算法实现不同点云团间的分离, 将点云团分为若干距离差距较大的点云簇。由于点云团中果实表面点云点数量占比最大, 基于欧式距离的分割算法获得的点云簇中, 点数量最多的即为苹果果实表面点云。

3.2. 苹果 3D 中心定位

苹果果实的整体外观可近似为一个球体, 球面的法向量指向球体中心, 因此可用此假设估算球体中心来近似表示苹果果实的中心。将苹果点云分割为 2 大块, 分别计算两块点云指向内部的法向量; 由于苹果果实并非标准球体, 且表面存在噪声干扰, 因此这两条法向并不会在刚好中心点相交, 但都有可能从中心点附近经过。针对这个问题, 本文通过计算两法线中垂线的交点估计果实中心点。由于获取的果实侧表面点云数据必定小于苹果半球, 在这种情况下, 点云中三维坐标变化值最小的方向为朝向球心的方向, 故可用此原理, 获取该点云矩阵最小的正交分量向量, 作为点云块指向果实内法向量。

由于计算果实 3D 中心坐标需要借助到两块不同的果实表面点云数据来计算, 故需要将获得的果实侧表面点云分为两块。果实获取到的表面点云大体分为两种: 1) 未露出果梗果柄的果实即相机采集到的点云仅为果实侧表面; 2) 露出果柄或果梗的果实, 相机采集到的数据包含果柄、果梗位置点云。在计算中心的过程中, 每块点云的点数量应大致相等, 且不能出现环状点云否则会对中心估计产生较大影响。而在露出果梗果柄的果实表面点云中, 经过曲率分割处理后果实表面点云容易出现空心区域形成环状点云。为解决这个问题同时实现点云的一分为二, 可在相机坐标系下以点云中点 x 坐标为基准, 将点云切成左右两块, 这样分离的点云既可以实现两点云块的点数量差异小, 也能避免点云块中环形区域的出现。具体中心计算过程如下:

a) 取苹果果实表面点云, 计算点云 x 方向的均值, 将点云以 x 的均值 x_0 为基准, 将点云一分为二。取分割后的两块点云块, 分别求其质心 C_1 、 C_2 。

b) 利用 PCA 算法计算分别获取两块点云团的朝内法向量, 将每块点云矩阵进行 PCA 计算, 取出每个矩阵最小的特征值对应的特征向量作为该每块云团的朝果实内法向量, 将这两条单位法向量定义为 p_1 、 p_2 。

c) 设 $p_3 = p_1 \times p_2$, 将 p_3 作为 p_1 和 p_2 的中垂线方向向量, 即 p_3 与 p_1 、 p_2 垂直。

设 r_1 、 r_2 分别为向量 p_1 、 p_2 的长度, 形成直线段 l_1 、 l_2 , 将 p_3 的方向作为 l_1 、 l_2 的公垂线方向, 设 k 为一个标量, 其数值为直线 l_1 、 l_2 之间的中垂线的长度的 $1/2$ 。

联立等式:

$$C_1 + r_1 \cdot p_1 + k \cdot p_3 = C_2 + r_2 \cdot p_2 - k \cdot p_3 \quad (2)$$

解方程组, 得到参数, 计算 l_1 和 l_2 的中垂线中心点作为果实中心 C_a 。

具体流程如图 2 所示, 图 2(a)、图 2(b) 分别为同一块残缺点云不同视角下的中心定位示意图。计算点云块 1 (红色点) 的点云质心坐标, 通过 PCA 算法获取该点云块以其质心为起点的朝内直线方向 l_1 (红色线)。用同样的方法获得点云块 2 (蓝色点) 的质心和 l_2 (蓝色线)。求 l_1 、 l_2 的中垂线 l_3 (绿色线), 并计算两直线内公垂线中点(绿色线中点), 其中黑点坐标即为计算的果实中心坐标 C_a 。

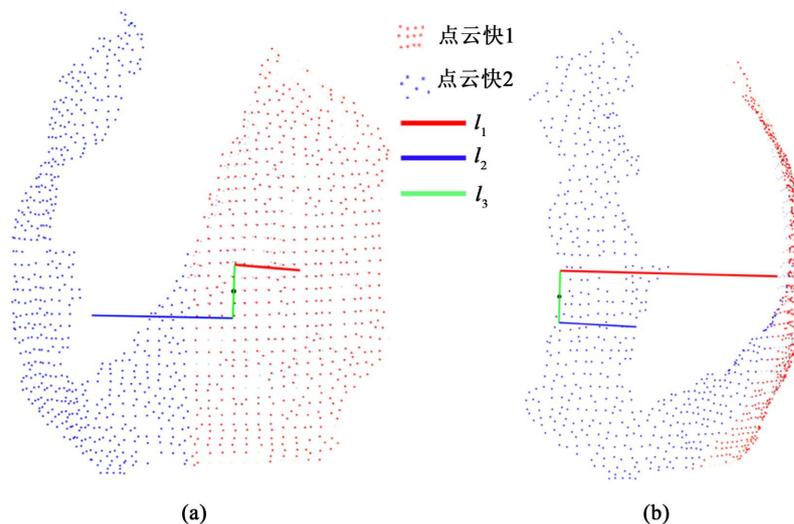


Figure 2. Schematic diagram of fruit center positioning
图 2. 果实中心定位示意图

3.3. 苹果主轴估计

计算果实主轴需要利用果实的具体三维特性, 假设苹果果实为一个类球形, 如图 3 所示。果梗柄位置弯曲程度较大, 而果实的侧表面则呈现出平滑圆润的曲面。如图 3(a)所示果实水平方向剖面可以看作是一个圆形, 其表面法向量向内汇于中心; 如图 3(b)所示果实垂直剖面是一个曲率变化的曲面, 其表面上的各点可用不同半径的椭圆拟合, 椭圆的中心位于果实中心所在位置的垂直剖面。若干个椭圆的中心点可以近似拟合出一条坐落于果实水平剖面的直线, 只需求出这个水平剖面的法向则可作为果实的主轴方向。

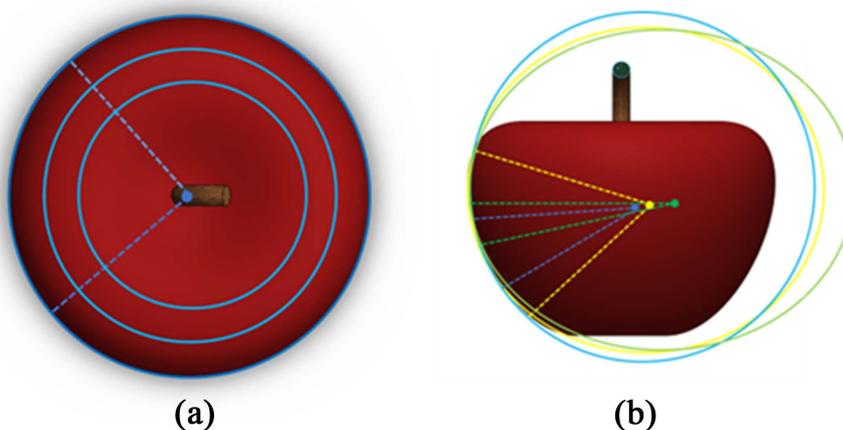


Figure 3. Schematic diagram of calculation principle of apple fruit axis
图 3. 苹果主轴计算原理示意图

为验证该假设, 本文计算了果实表面点两两内法线的公垂线中心点, 如图 4 所示。实验中发现分割开的两块点云中, 每块点云计算的公垂线中点都近似分布在穿过果实中心且平行于果实鼓面的一条直线上, 且这两条线都与果实主轴线垂直。因此可以通过最优化方法计算获取两条线的方向向量, 通过计算两条直线的方向向量叉乘获得果实主轴线的方向。

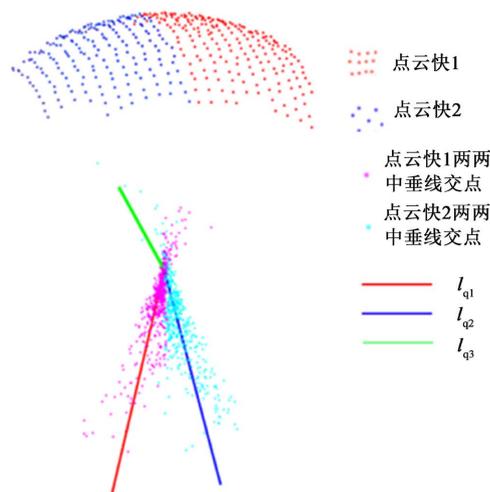


Figure 4. Schematic diagram of apple fruit axis calculation
图 4. 苹果果实主轴计算示意图

具体算法流程如下:

1) 取果实表面点云块 1, 遍历计算苹果表面点云中每两个距离大于 3 cm 的点的内法线的中垂线交点 Q , 交点总数设为 n 。

2) 假设拟合直线 l_{q1} 过果实中心点 $C_a(C_a.x, C_a.y, C_a.z)$ 且其方向向量为单位向量 $\mathbf{lx}_1(x, y, z)$, 向量 \mathbf{m} 代表中垂线交点 $Q(Q_n.x, Q_n.y, Q_n.z)$ 与果实中心 C_a 两点之间的向量。其中果实中心 C_a 在直线 l_{q1} 上, \mathbf{p} 代表拟合直线的三维方向向量。

$$\mathbf{m} = (Q_n.x - C_a.x, Q_n.y - C_a.y, Q_n.z - C_a.z) \quad (3)$$

$$\mathbf{p} = (x_a, y_a, z_a) \quad (4)$$

3) 建立目标函数, 通过最优化算法, 求出拟合的直线 l_{q1} (红色线) 单位向量 \mathbf{lx}_1 的最优化结果。并对果实表面点云块 2 进行同样操作, 获得拟合直线 l_{q2} (蓝色线) 的单位向量 \mathbf{lx}_2 。其中, 目标函数和约束可由如下公式表示。目标函数的含义是点云块每两个点的中垂线交点和拟合直线的距离, 约束条件的含义是拟合直线的方向向量为单位向量。

$$\min \text{Obj} = \sum_{i=1}^n |\mathbf{m} \times \mathbf{p}|^2 \quad (5)$$

$$s.t. \ x_a^2 + y_a^2 + z_a^2 = 1 \quad (6)$$

4) 对点云块 2 用同样的方法重复以上步骤获得向量 \mathbf{lx}_2 , 设 $\mathbf{lx}_3 = \mathbf{lx}_1 \times \mathbf{lx}_2$, 将 \mathbf{lx}_3 标准化, 获得果实主轴线 l_{q3} (绿色线) 的方向向量。

4. 实验

4.1. YOLO v5s 检测实验

训练该网络使用的 GPU 是 NVIDIA GTX 3090, 图像总数为 7296 张, 这些图像中的苹果被人为地逐个标记, 苹果果实目标根据未被枝叶遮挡和被枝叶遮挡分为两类, 分别标记为 ap 和 ap1。其中 80% 的图像用于训练, 20% 的用于验证。在目标检测中精度、召回率和平均值通常用于评估准确性。训练使用 300epoch、batchsize 设置为 4、图片输入像素尺寸为 640 * 640。验证集部分检测结果如图 5 所示。



Figure 5. Result of the verification set detection

图 5. 验证集检测结果

苹果目标检测模型评价指标如表 1 所示:

Table 1. Model evaluation metrics

表 1. 模型性能指标

precision	recall	mAP_0.5	mAP_0.5:0.95
0.9923	0.9910	0.9943	0.8718

4.2. VREP 果园环境仿真实验

为验证苹果中心定位、主轴估计算法的精确度, 本文采用 VREP 虚拟机器人仿真平台来测试该算法的精度和有效性。该平台可部署 3D 苹果模型以及遮挡树叶模型, 通过深度相机获取点云, 用来仿真模拟实际场景中获取的点云图。

建立苹果果树枝叶模型和 5 种不同形状直径为 10~13 cm 不等的苹果果实模型, 预先设置好果实中心位置、果实主轴方向, 通过 VREP 仿真平台中深度相机获取模拟实际果园采摘场景中的点云, 进行定位算法的计算。将计算获得的果实中心数据以及主轴数据与预设的数据对比, 从而评估该算法的性能。为模拟实际场景中因为各类环境因素造成的随机干扰, 测试算法的鲁棒性, 将获取到的点云数据中添加高斯噪声 $N(0, 0.02 \text{ cm})$ 。

仿真实验中总共设置了 5 个场景, 每个场景 11 个苹果, 包含被枝叶遮挡和未被枝叶遮挡的苹果。苹果总数 55 个, 其中受到枝叶遮挡的苹果 30 个, 遮挡率在 20%~50% 不等, 苹果主轴距相机坐标系 y 轴有 $20^\circ \sim 60^\circ$ 的偏移度。有 33 个苹果在相机镜头视角下露出了果梗位置即包含了果梗位置的点云。在数据预处理模块, 曲率分割阈值设置为 0.003, 欧氏距离分割阈值设置为 0.5 cm。部分场景如图 6 所示, 图 6(a) 中展示了 VREP 仿真平台中果树模型, 图 6(b) 展示在该场景中由深度相机获取到的点云数据。

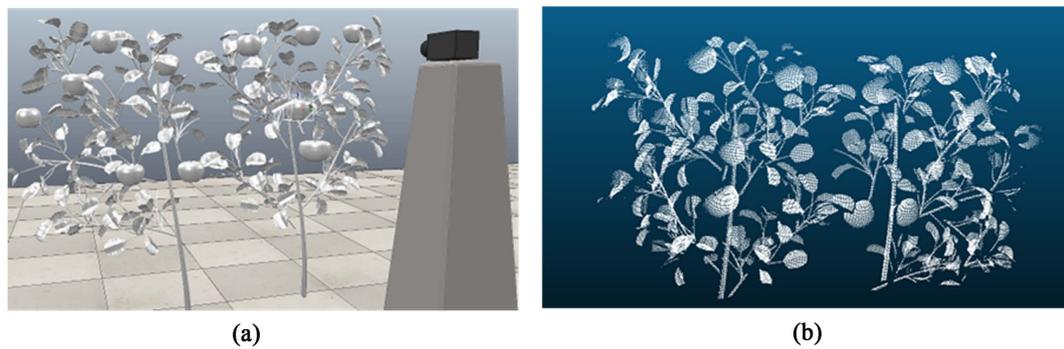


Figure 6. VREP simulation experiment
图 6. VREP 仿真实验

由于苹果果实模型在建立过程中包含了果柄，果实中心点位于包裹果实的正方体中央，而计算果实中心数据时并不会计算果柄数据，因此中心计算时往往会有一个在相机坐标系下垂直方向上的系统误差，即果柄长度，约 2 厘米左右。同时，其主轴方向计算容易受到各种情况的干扰从而形成误差。考虑到苹果果实为类球体形状，机器人采摘时对主轴具有较大的容忍度，并不需要非常精确的结果， 20° 左右的偏差对采摘结果影响有限。经计算，实验结果如表 2 所示：

Table 2. Average error data of apple pose calculation
表 2. 苹果位姿计算平均误差数据

编号	名称	中心定位平均误差(cm)	主轴计算误差($^\circ$)
Ec	总误差	2.59	19.3
Ec1	露出果梗果实误差	2.79	24.0
Ec2	受遮挡果实误差	2.91	18.4
Ec3	未露出果梗果实误差	2.47	16.1

通过箱型图来显示中心定位实验结果，如图 7 所示。蓝紫色框内代表了 25%~75%数据的分布情况，红色线代表数据的中位数分布，蓝色圆圈代表了数据的平均数位置。通过该箱型图可以看出，中心定位误差基本分布在 2~3 cm 范围内，满足预测的误差精度范围。

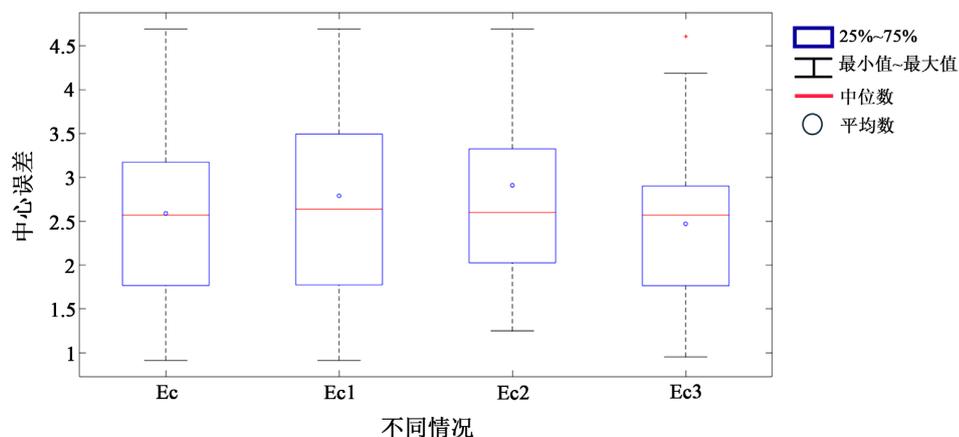


Figure 7. Box diagram of apple center positioning error
图 7. 苹果中心定位误差箱型图

通过分析比较结果发现, 各类情况对定位的影响不大。而在主轴计算中, 露出果梗的情况对果实主轴估计的误差最大, 其原因或为由于视角因素, 果实侧表面点云获取部分较小, 从而导致估计精度严重下降。个别数据出现精度非常低的情况, 经检查主要原因是由于果梗位置和枝叶遮挡的影响, 造成果实侧表面点云数量过少, 从而直接影响了计算结果的准确性。

4.3. 苹果果实采摘机器人实地场景实验

为验证苹果果实定位算法的有效性, 将算法部署在自研发的苹果果实采摘实验平台上, 于 2022 年 10 月在山东威海苹果果园进行采摘作业。该算法使用 Visual Studio 平台在 windows 系统中部署, 使用 C++ 语言、OPENCV 库和 PCL 点云库编写程序。所有程序以 MFC 界面工程的形式整合, 连接 Intel RealSense L515 相机, 通过该相机采集的数据来调试验证算法。

在实地场景中, 苹果果树高度不超过 3 米, 按列种植在田地里, 果实果柄与果树枝干直接相连, 有近一半多的果实受到枝叶的不规则遮挡, 苹果果园实验环境如图 8 所示, 其中图 8(a)为苹果果实采摘实验平台, 图 8(b)为采摘机械臂和末端机械手。



Figure 8. Orchard field test
图 8. 果园实地测试

实验过程中, 成功定位苹果的判定方法为采摘爪的能完整包裹住生长在树上的苹果, 并能成功将苹果从树上拽下, 放入果实传送带上。在完整连续的记录中, 采摘爪共尝试采摘 60 个苹果, 成功 51 个, 失败 9 个。如图 9 所示, 成功的采摘如图 9(a)、图 9(b)所示。其中, 失败的 9 个苹果中:

1) 2 个苹果由于受到苹果枝条遮挡, 且枝条过细, 欧式聚类未能分割出该枝条遮挡物点云, 导致未能识别枝条遮挡物的存在, 机械夹爪在采摘苹果并拖拽的过程中, 受到另一个苹果枝条的干涉, 苹果果实从夹爪中滑落, 采摘失败, 如图 9(c)、图 9(d)所示;

2) 3 个苹果由于定位不准, 末端执行器夹爪并未抓紧苹果, 导致采摘完成后机械臂在拖拽苹果的时候果实从机械爪中脱落, 如图 9(e)、图 9(f)所示;

3) 1 个苹果由于主茎问题, 末端执行器夹爪在旋转过程中受到枝条遮挡, 无法使果梗与枝条分离导致采摘失败, 如图 9(g)、图 9(h)所示;

4) 2 个苹果由于外表形态原因, 末端执行器夹爪在闭合过程中, 机械爪将果实推开, 导致抓取失败, 如图 9(i)、图 9(j)所示;

5) 1 个苹果由于末端执行器夹爪在行进路线中碰撞苹果, 导致末端执行器采摘位点在到达苹果中心点之前, 夹爪碰撞到苹果造成弯折, 抓取不成功, 如图 9(k)、图 9(l)所示。经验证, 该算法可以在果园复杂环境中稳定作业。

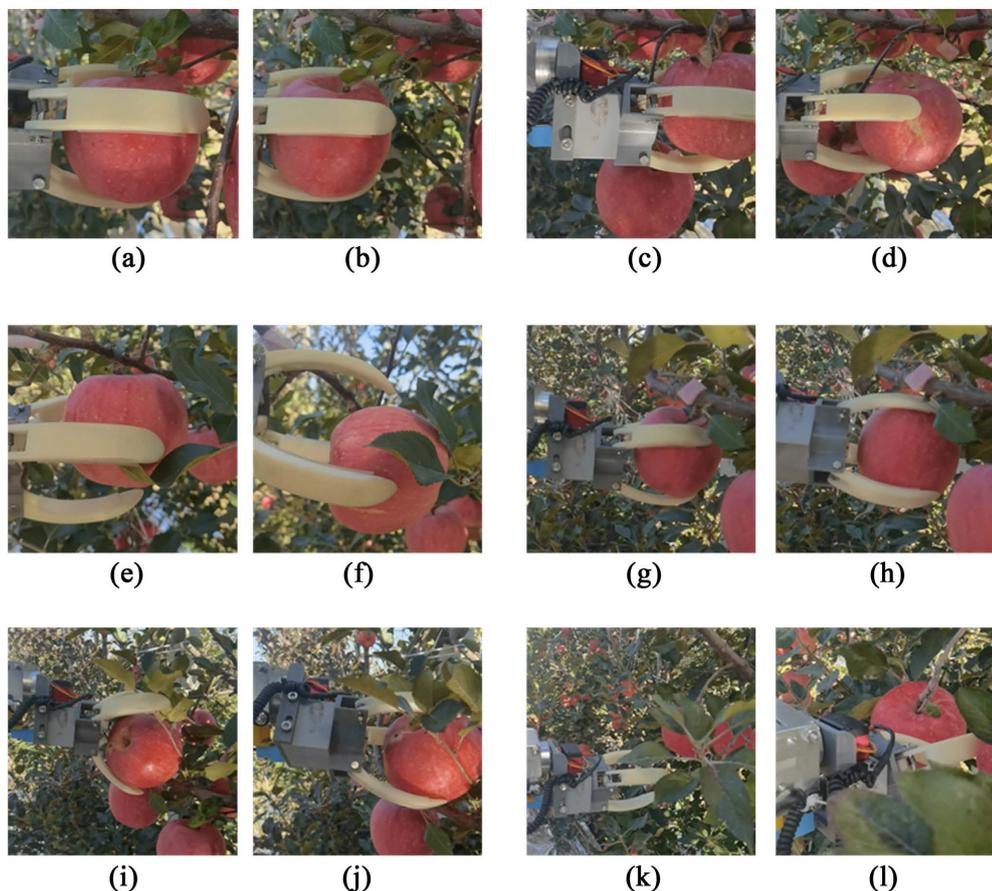


Figure 9. Schematic diagram of apple fruit grabbing
图 9. 苹果果实抓取示意图

5. 结论

1) 使用 YOLOv5 目标检测网络, 训练了一个用于果园环境中的苹果果实检测模型, 该模型可实现 RGB 图中被遮挡和未被遮挡的苹果果实的目标检测, 同时利用深度相机获取目标区域点云。

2) 利用 PCA 算法和最优化算法, 提出了一种果园环境中的苹果果实位姿定位算法, 该算法可针对被遮挡和未被遮挡的苹果果实点云计算其中心坐标和主轴方向, 用于引导机械臂进行采摘作业。

3) 利用 VREP 机器人仿真平台, 建立苹果、果树树枝、树叶模型, 模拟实际采摘场景中被遮挡和未被遮挡的苹果, 计算并证明了该果实定位算法的准确性和有效性。苹果中心点定位总的定位平均误差 2.59 cm, 主轴平均误差为 19.3°。在实地场景中部署果实采摘机器人和该果实中心定位算法, 实现苹果的连续采摘, 采摘成功率为 85%, 验证了算法的有效性和鲁棒性。

参考文献

- [1] Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H. and Iwai, A. (2019) An Automated Fruit Harvesting Robot by Using Deep Learning. *ROBOMECH Journal*, **6**, Article No. 13. <https://doi.org/10.1186/s40648-019-0141-2>
- [2] Plebe, A. and Grasso, G. (2001) Localization of Spherical Fruits for Robotic Harvesting. *Machine Vision and Applications*, **13**, 70-79. <https://doi.org/10.1007/PL00013271>
- [3] Williams, H.A.M., Jones, M.H., *et al.* (2019) Robotic Kiwifruit Harvesting Using Machine Vision, Convolutional Neural Networks, and Robotic Arms. *Biosystems Engineering*, **181**, 140-156. <https://doi.org/10.1016/j.biosystemseng.2019.03.007>

- [4] Kondo, N., Yamamoto, K., Shimizu, H., *et al.* (2009) A Machine Vision System for Tomato Cluster Harvesting Robot. *Engineering in Agriculture, Environment and Food*, **2**, 60-65. [https://doi.org/10.1016/S1881-8366\(09\)80017-7](https://doi.org/10.1016/S1881-8366(09)80017-7)
- [5] Lehnert, C., Sa, I., McCool, C., Upcroft, B. and Perez, T. (2016) Sweet Pepper Pose Detection and Grasping for Automated Crop Harvesting. *IEEE International Conference on Robotics and Automation (ICRA) Stockholm, Sweden*, 16-21 May 2016, 2428-2434.
- [6] Kang, H.W., Zhou, H.Y., Wang, X. and Chen, C. (2020) Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting. *Sensors*, **20**, Article 5670. <https://doi.org/10.3390/s20195670>
- [7] Li, H., Zhu, Q.B., Huang, M., Guo, Y. and Qin, J.W. (2018) Pose Estimation of Sweet Pepper through Symmetry Axis Detection. *Sensors*, **18**, Article 3083. <https://doi.org/10.3390/s18093083>
- [8] Gualtieri, M., Pas, A., Saenko, K. and Platt, R. (2016) High Precision Grasp Pose Detection in Dense Clutter. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, 9-14 October 2016, 598-605. <https://doi.org/10.1109/IROS.2016.7759114>
- [9] 赵静, 鲁力群. 果形识别中果梗判别与果轴确定方法[J]. 山东理工大学学报(自然科学版), 2004, 18(5): 28-32.
- [10] 姚立健. 基于惯性主轴的水果姿态检测[J]. 湖北农业科学, 2012, 51(17): 3862-3865.