

基于机械臂和深度相机的草莓采摘机器人

小组成员：邱泽宇，马卓远，金冬阳

指导教师：于仕琪

基于机械臂和深度相机的草莓采摘机器人

- 1. 方案设计
 - 1.1 识别模块
 - 1.2 定位模块
 - 1.3 抓取模块
- 2. 开发过程
 - 2.1 开发过程
 - 2.2 开发中遇到的困难
 - 2.2.1. 获取彩色相机的图像
 - 2.2.2. 获取深度流信息
 - 2.2.3. 阻塞识别模块和定位模块后重新启动
 - 2.2.4. 机械臂运动轨迹影响抓取效果
 - 2.2.5. 开发板上配置深度相机驱动和opencv环境后，程序无法运行
- 3. 成果说明
 - 3.1. 成果
 - 3.1.1. 对草莓的识别及定位
 - 3.1.2. 抓取草莓
 - 3.2. 后续可行的改进方向
 - 3.2.1. 提升识别准确率
 - 3.2.2. 提升定位准度
 - 3.2.3. 优化机械臂的运动轨迹
- 4. 测试说明
- 5. 开发环境

1. 方案设计

为实现对草莓的识别、定位和抓取的功能，本组将程序设计大致分为三个模块：识别模块，定位模块，抓取模块。

1.1 识别模块

对于草莓的识别，本组初步决定使用opencv库中提供的方法处理彩色相机获取的RGBD信息。首先使用滤波函数去除杂点，然后识别出取红色部分并提取轮廓，每一次将面积最大的部分的红色部分定位为草莓。使用fitEllipse方法为找到的轮廓绘制近似的椭圆，并将椭圆的中心定位为草莓在RGB图像中的相对位置，将该信息传输到定位模块。

未来可改进方向：本方案只是简单的根据颜色进行识别，没有很好的利用深度信息。未来可以结合深度相机的数据对草莓进行3D重建，辅助识别并判断彩色相机识别的草莓是否真的是草莓，并采用深度学习提高草莓的识别准确度。

1.2定位模块

在得到识别到草莓并得到草莓在彩色图像中的位置信息后，本组对彩色图像和深度图像进行对齐，通过AstraSDK中提供的接口函数实现二维图像中的位置信息转化成三维空间真实位置信息。然后将真实世界坐标传输给抓取模块，并阻塞识别模块和定位模块的运行，等待抓取模块运行完成后继续识别和定位新的下一个草莓。

1.3抓取模块

抓取模块在获得定位模块信息传输后，通过串口将位置信息传输给机械臂，通过反向运动学控制机械臂抓取定位到的草莓。在抓取完成后，通知识别模块和定位模块继续运行，并等待下一次传输位置信息。

2. 开发过程

2.1开发过程

本组本组首先是收集相关方面的机器人结构资料，参考他们的产品进行设计，初步建立了一个机器人的模型设计预想图。然后进行论文文献等相关资料的查询，确定了实现本项目的算法等。

在拿到深度相机前，本组首先使用普通相机和python进行程序开发，测试了基本函数对草莓的识别效果，并且联动控制机械臂，从而实现了对于定位置的草莓的抓取和移动。

在拿到深度相机后，本组使用的是AstraSDK驱动，安装好相关驱动之后测试了AstraSDK提供的样例中彩色流信息和深度流信息的获取代码，运行结果很成功，AstraSDK安装成功。

在测试过AstraSDK驱动的安装情况之后，本组将已有的python代码移植到C++中，并且仿照AstraSDK提供的样例进行相关的改写，增加相关的listener进行监听，从而可以成功读取相关的彩色流信息。

之后测试获取深度流信息，并定位像素点的真实世界坐标函数，然后将识别算法与定位算法联动，完成了草莓识别和定位的大体。

在调试了识别算法与定位算法，并且确定了他们的定位准确度后，本组将算法与抓取模块进行联动，成功实现了所有功能的简易运行。

之后，本组调整了摄像头与机械臂的相关位置，改善了他们相对应的坐标轴转化问题，机械臂的运动轨迹等细节。

最后，本组在开发板上配置了摄像头AstraSDK的驱动和opencv的开发环境，将程序移植到开发板上。

2.2开发中遇到的困难

2.2.1.获取彩色相机的图像

对于彩色相机的使用，本组初步决定使用opencv库中提供的方法处理彩色相机获取的RGBD信息。实验结果如图2.2.1.0所示。

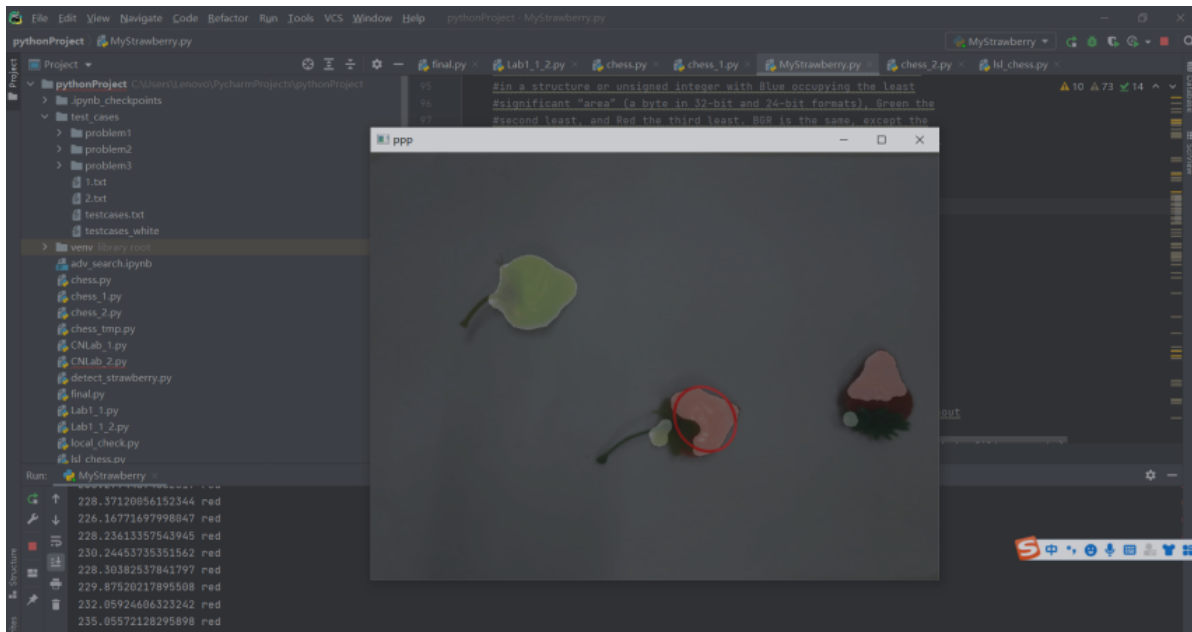


Fig.2.2.1.0

2.2.2.获取深度流信息

获取了深度流信息之后，本组在彩色流信息和深度流信息结合问题上遇到问题，经过讨论分析，得出处理方案，由彩色图像进行初步定位，对彩色图像和深度图像进行对齐，通过AstraSDK中提供的接口函数实现二维图像中的位置信息转化成三维空间真实位置信息。

2.2.3.阻塞识别模块和定位模块后重新启动

本组进行讨论，决定当机械臂传回信息之后重启识别模块和定位模块。

2.2.4.机械臂运动轨迹影响抓取效果

本组发现机械臂的机械抓手夹可能会影响草莓的抓取情况，本组计划在识别草莓后，将机械臂移动至合适位置后，再根据机械臂的位置及角度调整机械抓手夹的角度，从而优化项目。

2.2.5.开发板上配置深度相机驱动和opencv环境后，程序无法运行

在配置AstraSDK驱动的过程中，按照教程操作比较顺利，唯一遇到的问题是如Fig.2.2.5.0所示，然后通过debug解决问题：把前面的文件夹中的.otf文件找到拷贝到执行文件的同一目录中。

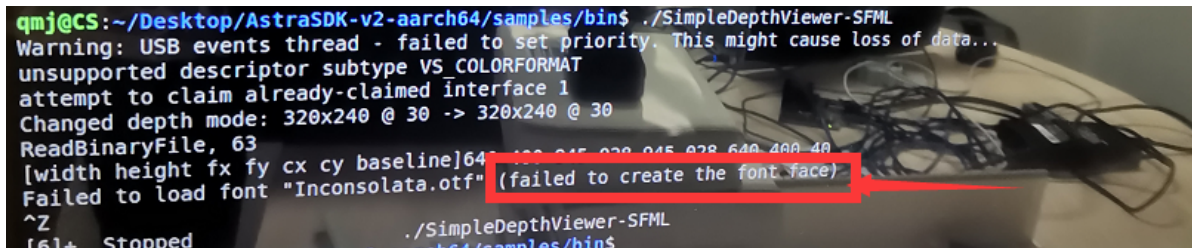


Fig.2.2.5.0

在开发板上配置opencv的过程之中总是配置不成功。第一次本组按照<https://www.cnblogs.com/fx-blog/p/8213704.html>的操作进行配置，但是配完之后，运行程序结果如Fig.2.2.5.1所示，结果失败。



Fig.2.2.5.1

然后本组另外找了一篇指导，按照<https://blog.csdn.net/public669/article/details/99044895>这篇文章进行操作。第一次进行到“sudo make -j8”这一步时，结果如图2.2.5.2所示，一直卡住，结果失败。后来发现是由于开发板性能不佳，无法支持8个线程进行操作，于是改成“sudo make -j1”完成操作。



Fig.2.2.5.2

3.成果说明

3.1.成果

3.1.1.对草莓的识别及定位

由下面的截图可以看到，在彩色流图像中识别到草莓后，程序会创建新窗口并在用椭圆标识识别到的草莓，然后在控制台中输出草莓的真实世界坐标。

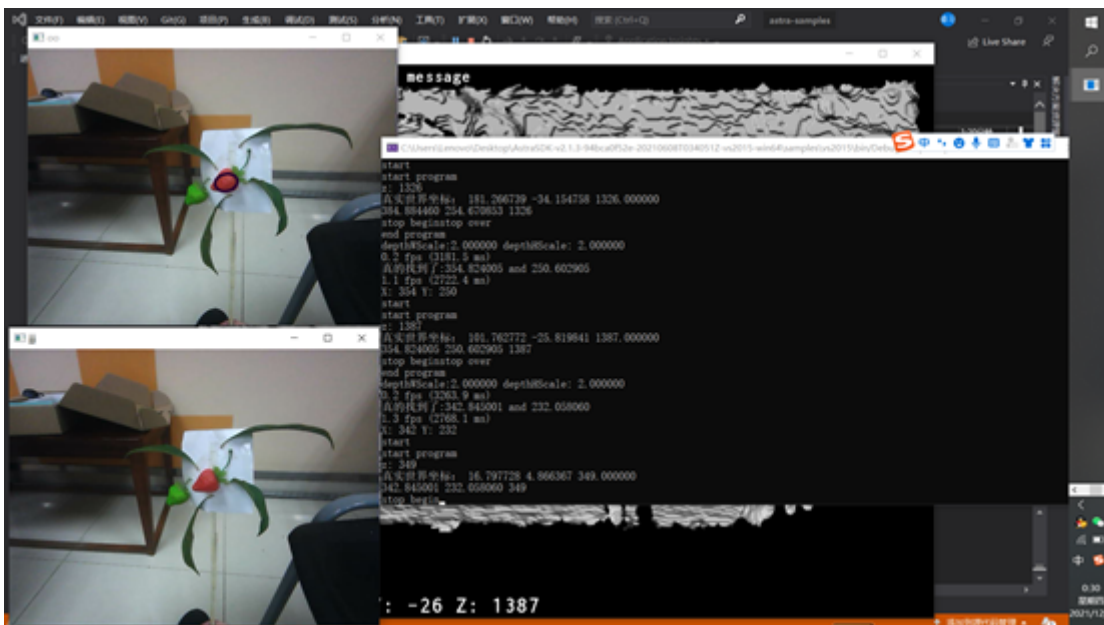


Fig.3.1.1

3.1.2.抓取草莓

由下面的图片可以看到，在左边的深度相机识别并定位草莓后，将真实世界坐标传输给机械臂，通过坐标转换，机械臂会得到识别到的草莓相对自身的位置，并通过反向运动学控制机械臂抓取草莓，然后放到机械臂底座旁边的收纳盒中。



Fig.3.1.2

3.2.后续可行的改进方向

3.2.1.提升识别准确率

根据测试结果，通过图像处理来识别草莓的准确度不够高。本组计划在今后的优化中使用深度相机对草莓进行3D重建，辅助识别判断草莓，提升是别的准确度。

3.2.2.提升定位准度

根据测试结果，目前的定位方法较为简单，仅仅根据二维图像的中心进行定位，准度较低。在后续的优化中，我们计划根据3D点云找到最适合的抓取位置来提升定位准度，优化抓取效果。

3.2.3.优化机械臂的运动轨迹

由于机械臂的运动可能会晃动草莓植株的茎秆导致目标草莓的移动，从而影响抓取效果。在后续的优化中，我们希望机械臂会根据目标环境自动优化抓取草莓过程的运动轨迹。

4.测试说明

本组在测试部分共设计了两大主要内容，一是测试RGB二维图像中草莓的识别准确率，二是测试三维空间定位的准确率。

对于第一部分的测试，本组拍摄了50张不同角度的带有草莓果实的草莓植株图片，并使用程序的识别部分独立测试，统计未成功识别的图片数量，统计结果如下：

图片数量	识别成功数量	识别失败数量	准确率
50	43	7	86%

识别失败原因分析：被背景干扰；果实被叶片遮挡；多个果实重叠在一起

测试结果分析：准确率在可接受的范围内，后续可以通过增加相机数量，结合深度信息等手段提高准确率。现阶段可以暂时采用该方案。

对于第二部分的测试，本组的测试重点在AstraSDK提供的计算三维世界坐标方法中对于深度的计算是否可靠。本组的测试方案为：将深度相机垂直地面放置在与地面平行的平面上。在相机的正前方不同距离放置物体，统计结果。测试结果如下：

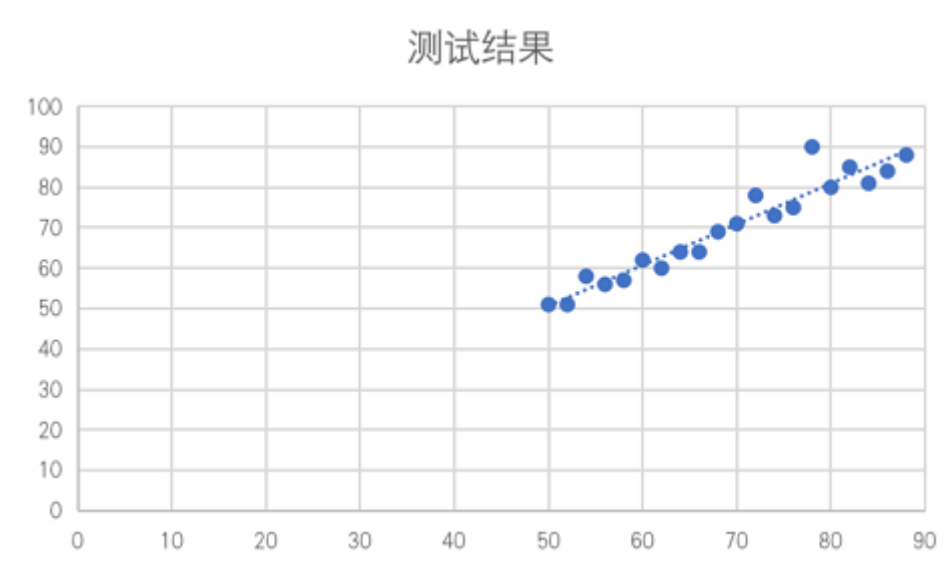


Fig.4.1.0

其中横坐标是距离相机的距离，纵坐标是测试出来的结果。通过拟合可以看到数据散落在一条直线周围，且误差范围不大。

5.开发环境

语言：C++ / Python

开发工具：Visual Studio 2015 / IDE Pycharm

硬件设备：PC (Windows10) / Arm7Bot 机械臂 / Orbbec Astra G系列深度相机