

实验报告

郭泳雨 2018202188

实验标题

树莓派智能机器小车的开发。

实验目的

利用树莓派智能机器小车自身携带的器件，例如摄像头、蜂鸣器、红外感知器等。利用Python的第三方库，结合树莓派智能小车出厂自带的Python程序，对小车的功能进行设计，以更加智能化的方式驱动小车运动。可以结合图像识别、硬件感知（红外感知器或称为红外避障模块）、音频感知等方式，体现出小车的智能化。

实验环境

OS: Windows 10 64bit

IDE: Python 2

实验阶段

第三阶段/最终阶段（截至2020年12月18日）

实验人员

李浩铭 2018202186 信息学院 2018级理科试验班6班

郭泳雨 2018202188 信息学院 2018级理科试验班6班

刘睿衡 2018202200 信息学院 2018级理科试验班6班

实验分工

本次实验报告为最后一次报告，故本次实验着重介绍本人在整个三阶段实验中的工作内容。我们的实验过程分为三部分，分别为：资料查询、代码搭建、最后调试。每一阶段，每一个人负责其中的一个部分，若出现难以解决的问题，则再由三人共同协商。

我主要负责的部分为第一阶段的代码搭建，第二阶段的最初调试，第三阶段的资料查询。

实验成果

第一阶段：

1. 蜂鸣器测试实验，其他各部件的测试。
2. 基于红外传感器的小车避障实验。
3. 基于OpenCV图像分析的小车循迹实验。

第二阶段:

1. 基于OpenCV的人脸识别，通过对图像中的人脸进行识别，并对人脸进行追踪。
2. 基于pillow和zbar（一个Python二维码识别的第三方库），扫描二维码，获取二维码中的内容指令，根据指令驱动小车。

第三阶段:

搜索资料，查询语音识别的方式，利用百度所提供的语音识别，对接收到的语音进行分析。根据语音输入，分析处理后与给定的指令进行对比，根据匹配的结果控制小车进行相应的动作。

实验代码

第一阶段:

第一阶段的代码已于第一次提交（当前目录下src_stage1），内容为：

1. Songs.py —— 音乐播放测试实验
2. Button.py —— 小车按钮测试实验，测试小车按钮是否正常
3. CarMove.py —— 小车运动函数，编写了驱动小车的相关函数
4. AdjustServo.py —— 舵机测试调试，用于调整小车的舵机以控制摄像头方向
5. InfraredModule.py —— 红外避障，根据红外传感器信息驱动小车的避障
6. FindTrial.py —— 小车循迹，根据摄像头传回图像进行分析并循迹运动
7. Adafruit_PCA9685 —— 厂商提供的舵机驱动Python标准库

第二阶段:

第二阶段的代码已于第二次提交（当前目录下src_stage2），内容为：

1. face_rec.py —— 人脸识别情况测试
2. face_move.py —— 人脸识别并追踪人脸
3. qrcode.py —— 二维码识别驱动小车

第三阶段:

第三阶段的代码于本次提交（当前目录下src_stage3），内容为：

1. audio_function_test.py —— 简单的测试，测试小车是否能响应调用
2. audio_test.py —— 测试语音交互能力，看小车的处理能力是否足够支撑交互
3. audio_controller.py —— 通过语音控制小车的行动方式

设计过程（三个阶段本人的工作）

第一阶段:

第一阶段，我主要负责的部分是代码构建。由于是第一次接触对硬件的程序设计和控制，所以在代码构建时依然存在一些困难。故最终完成实现时也 and 负责调试的组员进行了一些沟通配合。第一阶段最先实现的功能是对小车部件驱动的测试，包括蜂鸣器、主板上的控制按钮、四轮驱动、舵机驱动等。后续在各个部件测试正常后，尝试简单的实现了红外避障和小车循迹功能。

1. 各个部件的测试

对蜂鸣器的测试，与搜索资料的组员进行交流后，我采用了发出音响的一般方式，控制发音节拍、每个发音的高低（振动频率）、发音时间。实际代码构建过程，我去网络上搜索了C大调各音符的震动频率，并调整每个音节的振动时间，实现了蜂鸣器的测试。按钮的测试则是调整主板上的按钮按下后产生的信号，代码搭建时则是通过主板上的二极管发光情况来测试按钮是否按下，此测试主要是用于后期实验执行的控制。四轮驱动则是可以理解为调整输给马达正反向电流，实现小车的前进、后退、左右转弯。舵机实验则是通过控制舵机马达，调整其旋转角度，便于后期对摄像头的使用。

2. 红外避障算法实现

(1) 找到控制两个红外避障传感器的针脚参数，根据针脚初始化两个传感器为输入设备，并根据小车车轮的控制针脚设置车轮为输出设备；

(2) 对传感器的遮光情况进行分析，若左侧传感器被遮光则右转，同理右侧传感器被遮光则左转，若都被遮光则后退一定时间，转动车身（实验中设定为默认左转一定角度）到其他方向，否则小车前进，实验中的时间（用于控制小车运动时间或转动角度）、运动速度主要有后续进行调试的队员进行调整；

(3) 重复 (2)，直到外部触发控制小车停止避障。

3. 小车循迹算法实现

(1) 根据各个针脚，初始化摄像头为输入设备，车轮为输出设备；

(2) 调整摄像头到能拍摄到轨迹的角度；

(3) 对摄像头传回的每一帧图像进行分析，首先是进行灰度处理 (cv2.cvtColor)，去除图像色彩；再对图像进行高斯滤波 (cv2.GaussianBlur)，并对图像色彩设置阈值 (cv2.threshold)；腐蚀 (cv2.erode) 和膨胀 (cv2.dilate) 图像，使得图像中的轮廓更加清晰；通过库函数获取到处理过后图中轮廓的坐标 (cv2.findContours)；

(4) 根据获取的轮廓坐标求出中心点 (cx, cy)，设置 cx 在 x 轴的范围 [a, b] 内时小车直行，若 $cx < a$ ，则小车的左转弯， $cx > b$ 则小车的右转弯；

(5) 重复 (3)，(4)，直到外部触发控制小车停止循迹。

实验中的一些参数，如滤波时函数所需要的一些系数，范围 [a, b] 的设置等主要有调试队员进行调试设定，我负责协调处理。

第二阶段：

第二阶段我主要负责的部分是参数调整和部分的代码调整。第二阶段的实验，我们的目标是实现人脸识别与追踪、二维码识别驱动两项功能。主要涉及的实现方式是 OpenCV 图像识别，pillow 图片处理以及 zbar 对二维码信息的提取。

1. 人脸识别追踪调试

人脸识别主要是依靠 OpenCV 对面部特征的提取，考虑到面部识别的特点，我们从 OpenCV 的官网上下载了面部识别和眼部识别的数据文件，以这两个文件训练 OpenCV 的面部识别能力。实验中，经过多方面的测试，选取了窗口大小为 320*240，用于显示摄像头捕捉到的画面，并且作为每一帧图像分析时的图像大小。实验中，我们调试并在捕捉显示的图像中添加矩形识别框，并根据矩形框的中点位置判断小车的追踪方向。实验调试过程中，设矩形框的中点为 (x, y)，主要是判断 x 在图像中的相对位置，调试设置为 $x < 100$ 时小车的左转弯， $x > 220$ 时小车的右转弯，否则小车前进。而调试发现，一方面是摄像头捕捉的画面清晰度较低，另一方面可能是官网给的训练数据还不够充分，面部识别能力较弱，会对一些物体错误识别为人脸。

2. 二维码识别驱动调试

二维码识别，调试过程中主要采用的是识别手机图片中的二维码，而 OpenCV 不具备良好的二维码识别功能，或者其功能不能良好的辅助实验。因此调试并使用了 pillow 库，利用 pillow 对捕捉到的图片进行分析，转换为二进制文件后，再利用 zbar 的二维码识别分析工具，提取出二维码中的内容。调试过程中，由于摄像头自身像素较低，捕捉的图像清晰度较差，同时主板性能较弱，对图像的处理存在缺陷，因此最终调试时遇到了很多困难，使得最终虽然能较好的识别，但是整体来说识别能力还是较差。

第三阶段：

第三阶段我主要负责的是查询语音识别的相关资料。这一阶段，我们期望实现的是语音控制小车，通过语音下达一些简单的特殊指令，例如“前进”、“后退”、“左转”、“右转”等，控制小车进行相应的动作。

我早期的想法是通过对收集的语音进行波形分析，然后识别语音中的内容。但是小车的 CPU 处理能力很差，语音识别涉及的机器学习算法开销较大，小车的工作性能不足以支撑完成实验，为此只能选择

其他的处理方式。

进一步查阅资料，发现可以利用一些现有的语音识别库，通过API调用，简化小车的语音处理过程。查阅一些IT博客，我认为可以选择百度所提供的在线语音识别功能首先利用PyAudio库录制语音指令，保存为受支持的wav音频文件，然后利用百度语音识别库提供的方法实现语音识别，最后检测识别结果。百度语音识别为开发者提供业界优质且免费的语音服务，通过场景识别优化，为车载导航、智能家居和社交聊天等行业提供语音解决方案，准确率达到90%以上，这些介绍也是我最终认为选择百度云识别库的主要原因。语音识别库的提供厂商还有阿里云、腾讯云、讯飞AI等，最终选择百度云是因为它的操作相对简单，容易理解。

后期搜索过程中，发现有资料提到了speech_recognition第三方Python库，操作类似于调用百度语音识别库，且其效果也相对较好，但是此时由于实验代码已经在架构，所以没有提议使用这个库。

但是由于仅仅只实现简单的语音控制行进是较为简单的，于是我认为可以考虑识别一些其他的物体。例如我们可以计划设计一个识别功能，当我们说出某一样物品的名称的时候，小车就开始对摄像头传回的图像进行识别，当找到这个物体的时候，小车就停下，或者说说出某一个地点，小车就自动前往这个地点，当然在基础的情况下，我们可能需要人为的设定一些地点的位置；或者说出行动路线，当小车识别到语音中的地点时，就按照已有的行动路线行走，这可能就需要初始小车的位置应该是事先预定的。这些就涉及到多方面的功能要求，比如需要考虑到图像的识别性能，以及运行过程中会遇到前方有障碍物的情况，因此还要考虑躲避障碍，例如在寻物时若遇到障碍，就旋转方向，随机寻找一条可能的路线。

相关网页资料（部分）

1. Python简单语音识别并响应, <https://www.cnblogs.com/warcraft/p/10112486.html>
2. Python实现语音识别（基于百度语音识别）, https://blog.csdn.net/qg_36973838/article/details/85129273?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-3.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-3.control
3. Python实时语音识别控制, <https://www.cnblogs.com/demodashi/p/9437491.html>
4. python关于调用百度语音识别api的操作, <https://www.cnblogs.com/kunixiwa/p/8609843.html>

实验总结

本学期的实验主要围绕人工智能小车进行了一些编程及调试，过程中加深了对一些人工智能硬件方面的了解。但整体来说我们的实验还存在着一些缺陷，如实验过程中我们主要是调用了一些已有的库解决一些已经存在的问题，我们应该多创新一些内容，或者增加一些算法的功能。