

Sensored_walker 说明文档

组长：赵元培

组员：胡景文 杨文清

一、阶段目标

- 在第一阶段的基础上，完善小车蓝牙串口的安装
- 实现手机端利用蓝牙发送讯号控制小车
- 完成手机与 PC 端视频讯号的传递（DroidCamApp）
- 学习利用 python 处理图片、视频
- 实现并对比一些简单的计算图片相似度算法，为下一阶段提供思路

二、蓝牙串口说明

1. BT06 串口模块

- VCC：接 Arduino 的 5V
- GND：接 Arduino 的 GND
- TXD：一般表示为自己的发送端，接 Arduino 的 RX
- RXD：一般表示为自己的接收端，接 Arduino 的 TX

2. 手机端控制

- 使用软件：蓝牙小车 APP
- 优点：相比起蓝牙串口 APP，界面更加简洁，操作更方便
- 控制代码（部分）：

```
if(ch == 'F'){
    //前进
    Serial.println("up");
    motor_left.setSpeed(210);
    motor_right.setSpeed(180);
    motor_left.run(BACKWARD);
    motor_right.run(BACKWARD);
}
else if(ch == 'B'){
    //后退
    Serial.println("back");
    motor_left.setSpeed(210);
    motor_right.setSpeed(180);
    motor_left.run(FORWARD);
    motor_right.run(FORWARD);
}
```

三、 视频讯号传递

- 使用软件：DroidCamApp
- 阶段结果：成功实现与 PC 端的视频传递
- 传输延迟：USB < WLAN < Bluetooth

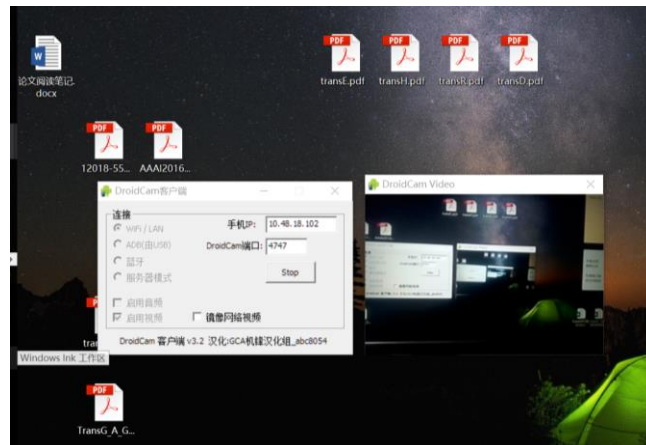


Figure1 在 PC 端显示视频讯号

四、 图像处理初步尝试

1. 环境配置

Python 2.7
openCV
Numpy
Matplotlib

2. 图片相似度算法

- **直方图**

在 Python 中利用 opencv 中的 `calcHist()` 方法获取其直方图数据，返回的结果是一个列表，使用 matplotlib，画出了这两张图的直方图数据图

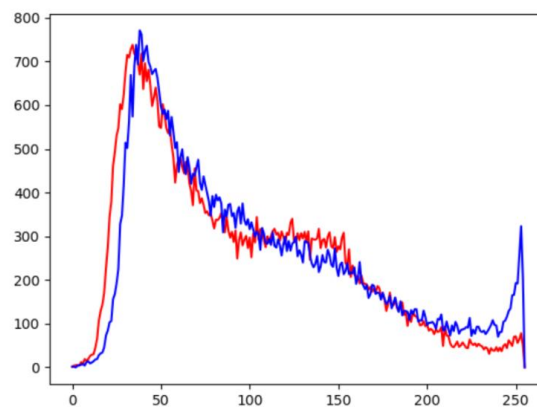


Figure2 直方图示例

- **平均哈希法(aHash)**

此算法是基于比较灰度图每个像素与平均值来实现的

一般步骤：

- a. 缩放图片，一般大小为 8*8，64 个像素值
 - b. 转化为灰度图
 - c. 计算平均值：计算进行灰度处理后图片的所有像素点的平均值
 - d. 比较像素灰度值：遍历灰度图每一个像素，大于平均值记录为 1，否则为 0
 - e. 得到信息指纹：组合 64 个 bit 位，顺序随意保持一致性
- 最后比对两张图片的指纹，获得汉明距离即可。

- **感知哈希算法(pHash)**

平均哈希算法过于严格，不够精确，更适合搜索缩略图，为了获得更精确的结果可以选择感知哈希算法，它采用的是 DCT（离散余弦变换）来降低频率的方法

- **dHash 算法**

相比 pHash，dHash 的速度要快的多，效果要更好，它是基于渐变实现的步骤：

- a. 缩小图片：收缩到 9*8 的大小，以便它有 72 的像素点
 - b. 转化为灰度图
 - c. 获得指纹：如果左边的像素比右边的更亮，则记录为 1，否则为 0.
- 最后比对两张图片的指纹，获得汉明距离即可

3. 算法比较

	去除颜色 保留灰度	绝对位置 设置偏差	灰度整体 基准不同
直方图	0.50784498	0.77417177	0.54242122
改进的直方图	0.48776686	0.79021007	0.48097572
aHash	3	18	11
pHash	6	23	5

4. 结果分析

- 对于不同对比组之间打乱顺序的相似度比较，所有算法的相似度都很低
- 可以看出，对于绝对位置有偏移的图片，直方图算法结果更好
- 对于改变整体灰度或者色调的图片，哈希算法更加有效
- 在之后的视频处理中可以参考借鉴这些方法

五、 人员分工

- 赵元培：编写、调试手机控制小车程序
完成并对比多种处理图片相似度的简单算法
展示阶段成果
- 胡景文：完善蓝牙串口等硬件组装
openCV 等的环境搭建
查阅处理图片相似度的算法
- 杨文清：完善蓝牙串口等硬件组装
寻找更好的手机端控制 APP
查阅处理图片相似度的算法

六、 存在的不足

- 暂时没有使用 scikit 工具进行机器学习算法的应用
- Python 的视频读入存在问题