树莓派 opencv 摄像头巡线

一、简介

小 R 科技树莓派镜像，默认已经集成了摄像头巡黑线的代码。可通过控制软件下发自定义指令来调试、或者退出巡线模式。

FF130800FF：摄像头调试或者退出摄像头巡线模式，

FF130801FF：开始摄像头巡线程序。

而程序主要分了两个部分：巡线控制、图像处理。

其中图像处理部分对摄像头数据进行处理：获取灰度图→二值化→像素点采样→计算黑点（数据 0）X 坐标值→输出坐标值。

巡线控制部分：获取黑点中心坐标值→判断坐标值的范围→根据坐标值范围进行小车方向控制（黑线在左就左转，在右就右转，在中间就直行）。

接下来我们做一个单独的摄像头巡线程序，启动程序后默认执行摄像头巡线操作。

PS：因为我们出货固件默认开通了 mjpeg-streamer 功能，已经对摄像头进行了占用，所以我们在单独开发 opencv 时，需要手动结束主程序进程。

方法 1：在/home/pi/work/wifirobots/startwifirobots.sh 脚本里面，把启动 wifirobots.py 的命令注释掉，重启就不会启动程序。如果需要恢复，就放开注释，重启即可。

方法 2：命令输入 ps -aux 显示所有进程，找到 python wifirobots.py 和 mjpeg_streamerxxxxx 的进程，记住进程号，两个分别在 650 和 1300 左右。

使用指令 sudo kill -9 ID1 ID2 结束对应进程（注意空格）。

二、二值化操作

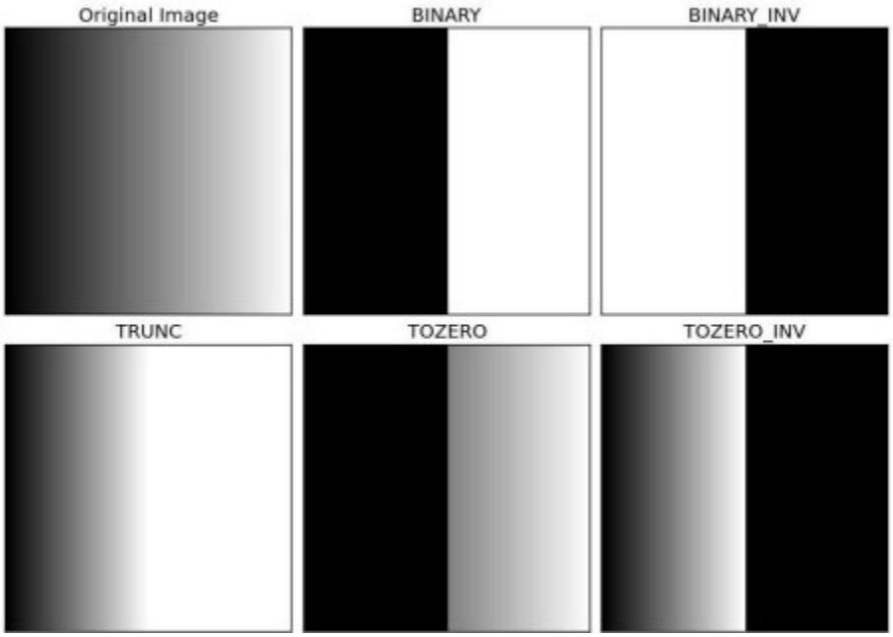
首先看一个常用的图像处理操作“二值化”，opencv 提供了多种二值化形式：

cv2.THRESH_BINARY
cv2.THRESH_BINARY_INV
cv2.THRESH_TRUNC
cv2.THRESH_TOZERO
cv2.THRESH_TOZERO_INV

PS：在灰度图像中，0-255 代表着亮度等级，黑色为 0，白色为 255。标准的二值化就是根据设定的亮度阈值，将超过这个亮度的点赋值为白色的 255，低于这个阈值的点赋值为黑色的 0。

我们目前做巡黑线，使用正常模式 cv2.THRESH_BINARY。如果是巡白线，就只需要修改成反色模式 cv2.THRESH_BINARY_INV 就可以将白色线条转换成黑色点来用同一个程序了。

这几种操作的效果不同，下图是一个灰度图在阈值为 127 下对应的输出图像



三、完整代码解析

```
#coding:utf-8
#Python 中声明文件编码的注释，编码格式指定为 utf-8
from socket import *
from time import ctime
import binascii
import RPi.GPIO as GPIO
import time
import threading
import cv2
import numpy as np
```

```
print '....WIFIROBOTS START!!!...'
global Path_Dect_px
Path_Dect_px = 320
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
########LED 口定义##################
LED0 = 10
LED1 = 9
LED2 = 25
########电机驱动接口定义##################
ENA = 13    #//L298 使能 A
ENB = 20    #//L298 使能 B
IN1 = 19     #//电机接口 1
IN2 = 16     #//电机接口 2
IN3 = 21     #//电机接口 3
IN4 = 26     #//电机接口 4
#########led 初始化为 000##########
GPIO.setup(LED0,GPIO.OUT,initial=GPIO.HIGH)
GPIO.setup(LED1,GPIO.OUT,initial=GPIO.HIGH)
GPIO.setup(LED2,GPIO.OUT,initial=GPIO.HIGH)
#########电机初始化为 LOW##########
GPIO.setup(ENA,GPIO.OUT,initial=GPIO.LOW)
ENA_pwm=GPIO.PWM(ENA,1000)
ENA_pwm.start(0)
ENA_pwm.ChangeDutyCycle(80)
GPIO.setup(IN1,GPIO.OUT,initial=GPIO.LOW)
GPIO.setup(IN2,GPIO.OUT,initial=GPIO.LOW)

GPIO.setup(ENB,GPIO.OUT,initial=GPIO.LOW)
ENB_pwm=GPIO.PWM(ENB,1000)
ENB_pwm.start(0)
ENB_pwm.ChangeDutyCycle(80)
GPIO.setup(IN3,GPIO.OUT,initial=GPIO.LOW)
GPIO.setup(IN4,GPIO.OUT,initial=GPIO.LOW)
##########机器人方向控制############################
def Motor_Forward():
    print 'motor forward'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,True)
    GPIO.output(IN2,False)
    GPIO.output(IN3,True)
    GPIO.output(IN4,False)
    GPIO.output(LED1,False)#LED1 亮
```

```
    GPIO.output(LED2,False)#LED1 亮
def Motor_Backward():
    print 'motor_backward'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,False)
    GPIO.output(IN2,True)
    GPIO.output(IN3,False)
    GPIO.output(IN4,True)
    GPIO.output(LED1,True)#LED1 灭
    GPIO.output(LED2,False)#LED2 亮
def Motor_TurnLeft():
    print 'motor_turnleft'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,True)
    GPIO.output(IN2,False)
    GPIO.output(IN3,False)
    GPIO.output(IN4,True)
    GPIO.output(LED1,False)#LED1 亮
    GPIO.output(LED2,True) #LED2 灭
def Motor_TurnRight():
    print 'motor_turnright'
    GPIO.output(ENA,True)
    GPIO.output(ENB,True)
    GPIO.output(IN1,False)
    GPIO.output(IN2,True)
    GPIO.output(IN3,True)
    GPIO.output(IN4,False)
    GPIO.output(LED1,False)#LED1 亮
    GPIO.output(LED2,True) #LED2 灭
def Motor_Stop():
    print 'motor_stop'
    GPIO.output(ENA,False)
    GPIO.output(ENB,False)
    GPIO.output(IN1,False)
    GPIO.output(IN2,False)
    GPIO.output(IN3,False)
    GPIO.output(IN4,False)
    GPIO.output(LED1,True)#LED1 灭
    GPIO.output(LED2,True)#LED2 亮
##########机器人速度控制############################
def ENA_Speed(EA_num):
    speed=hex(eval('0x'+EA_num))
```

```
    speed=int(speed,16)
    print 'EA_A 改变啦  %d '%speed
    ENA_pwm.ChangeDutyCycle(speed)
def ENB_Speed(EB_num):
    speed=hex(eval('0x'+EB_num))
    speed=int(speed,16)
    print 'EB_B 改变啦  %d '%speed
    ENB_pwm.ChangeDutyCycle(speed)
def PathDect(func):
    global Path_Dect_px    #巡线中心点坐标值
    while True:
        print 'Path_Dect_px %d '%Path_Dect_px  #打印巡线中心点坐标值
        if Path_Dect_px < 260: #如果巡线中心点偏左，就需要左转来校正。
            print("turn left")
            Motor_TurnLeft()
        elif Path_Dect_px > 420:#如果巡线中心点偏右，就需要右转来校正。
            print("turn right")
            Motor_TurnRight()
        else :        #如果巡线中心点居中，就可以直行。
            print("go stright")
            Motor_Forward()
        time.sleep(0.007)
        Motor_Stop()
        time.sleep(0.007)
cap = cv2.VideoCapture(0)    #实例化摄像头
Path_Dect_px_sum    = 0      #坐标值求和
threads = []
t1 = threading.Thread(target=PathDect,args=(u'监听',))
threads.append(t1)
for t in threads:
        t.setDaemon(True)
        t.start()
while True:
    ret,frame = cap.read()  #capture frame_by_frame
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)   #获取灰度图像
    ret,thresh1=cv2.threshold(gray,70,255,cv2.THRESH_BINARY)#对灰度图像进行二值化
    Path_Dect_fre_count = 1
    for j in range(0,640,5):        #采样像素点，5 为步进，一共 128 个点
        if thresh1[240,j] == 0:
            Path_Dect_px_sum = Path_Dect_px_sum + j          #黑色像素点坐标值求和
            Path_Dect_fre_count = Path_Dect_fre_count + 1    #黑色像素点个数求和
    Path_Dect_px = (Path_Dect_px_sum )/(Path_Dect_fre_count)   #黑色像素中心点为坐标和除以个数
    Path_Dect_px_sum = 0
    #cv2.imshow('BINARY',thresh1)          #树莓派桌面显示二值化图像，比较占资源默认注释掉调试时可以打开
```

```
    if cv2.waitKey(1)&0XFF ==ord('q'):#检测到按键 q 退出
        Motor_Stop()
        break
cap.release()
cv2.destroyAllWindows()
```

四、操作演示

我们拿一张纸上打印的黑线来测试程序。

1、代码另存为 03.PathDect.py，并上传到 opencv 文件夹。

2、python 03.PathDect.py 回车运行代码。（需要先结束掉开机自启动的 python 主程序）

3、可以通过调整二值化中的阈值参数 70 为合适的值。

ret,thresh1=cv2.threshold(gray,70,255,cv2.THRESH_BINARY)#对灰度图像进行二值化

4、可以通过打开如下位置的注释符号来开启在远程桌面显示二值化后的图像从而辅助上一步骤的阈值设定啦。

#cv2.imshow('BINARY',thresh1)　　#树莓派桌面显示二值化图像，比较占资源默认注释掉调试时可以打开

官 网：www.xiao-r.com

论 坛：www.wifi-robots.com

官方商城：wifi-robots.taobao.com

微信公众号：