板载实时人脸追踪系统

目录

一 课题研究背景与现状 3

1.1 研究背景 3

1.2 现状 3

二 研究目的及意义 3

2.1 研究目的 3

2.2 意义 3

三 方案设计 3

四 研究的主要内容 5

后续改进： 8

五 结论 10

六 成果应用前景 10

七 附录 11

一 课题研究背景与现状

1.1 研究背景

近年来对于人脸识别和追踪的研究可谓如火如荼。而人脸追踪，就是要在检测到人脸的前提下，在后续帧中继续捕获人脸的位置等信息的技术。

1.2 现状

人脸追踪技术在当下的生活场景中有着巨大的用处，在海关、机场、银行、电视电话会议等等场合中，都需要对特定的人脸目标进行跟踪。而当下人脸跟踪技术大致可分为四类：基于模型跟踪，基于运动信息跟踪，给予人脸局部特征跟踪和基于神经网络跟踪。

二 研究目的及意义

2.1 研究目的

我们项目硬件采用舵机云台，摄像头，arduino，和树莓派，基于 opencv 和 python，如果检测到面部，则将其聚焦于绿框内通过计算其中心和整个显示中心的差距，来调整摄像头的移动已让识别到人脸居于显示中心，达到对于人脸的跟踪效果。

2.2 意义

我们通过算法和硬件实现的简易人脸追踪，可以用于实现物体或小车的自动锁定对象，是其拥有可以自动跟随行进的能力。

三 方案设计

我们希望实现的功能是摄像头的人脸追踪，并可以将其运用在避障小车等其他作品上。为此我们需要使用两个舵机分别控制摄像头水平方向的旋转和俯仰方向的旋转，而使用一个微处理器来控制这两个舵机，另外我们需要人脸检测的算法，以实现对人脸的定位。

这里微处理器我们选用了 Arduino。Arduino 是一款便捷灵活、方便上手的开源电子原型平台，通过烧录程序可以控制灯电机、与其他程序通信等，同时它的开发环境类似于 C 语言，便于我们使用。

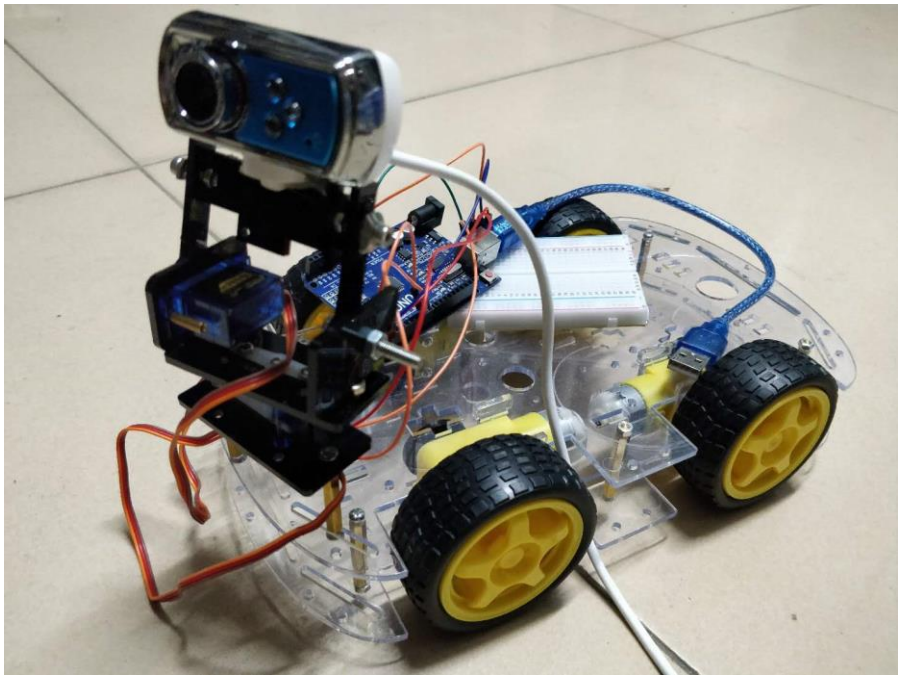
然而Arduino的运算性能不足以进行图像处理、人脸检测这样的复杂运算工作，因此我们将图像处理部分单独交给树莓派进行。树莓派是一款微型电脑主板，体积小巧，功能齐全，还拥有诸多硬件扩展配件。将摄像头连接至树莓派上，并运行 python 的人脸检测程序，通过 pyserial 可以与 Arduino 进行通信。

人脸检测的方法很多，比如 CNN 卷积神经网络。它虽然更准确但是也需要消耗更多的资源,树莓派毕竟资源受限,因此我们采用 Haar 分类器。Haar 分类器对 Haar-like 特征用积分图对特征求值加速，再用 AdaBoost 训练分类器并级联到一起提高准确率。因此检测速度较快。

人脸检测得到人脸的在画面中的位置后将坐标传递给 Arduino。为了将人脸移动到画面正中央，只需分别计算人脸坐标与中心位置的偏差，并向相反的方向移动即可。为了更加迅速而又平滑地实现这一目标我们还可以使用 PID 算法，但是需要受到前一步人脸检测的限制，因此仍然需要改进。

如此通过 Arduino，树莓派，以及 Haar 分类器，我们可以实现摄像头对人脸的追踪拍摄。

四 研究的主要内容



我们的目标是使用户可以保持在画面的正中央，一般情况下会使用一个二轴云台来完成，我们使用两个伺服电机组成一个具有两个自由度的云台，其中一个伺服马达负责左右旋转，另

一个负责俯仰角度变化。

树莓派作为控制端，对输入的图片信息进行处理，通过对人脸进行识别，找到人脸中心的坐标，并通过串口将坐标传送至 Arduino。

树莓派运行的 Python 代码如下：

```
import numpy as np
import serial
import time
import sys
```

```
import cv2

arduino = serial.Serial('/dev/cu.wchusbserial1410', 9600)
time.sleep(2)
print("连接中。。。连接成功")


face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)

while 1:
    ret, img = cap.read()
    cv2.namedWindow('img')
    #cv2.resizeWindow('img', 1000,500)
    cv2.line(img,(500,250),(0,250),(0,255,0),1)
    cv2.line(img,(250,0),(250,500),(0,255,0),1)
    cv2.circle(img, (250, 250), 5, (255, 255, 255), -1)
    gray    = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3)

    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),5)
        roi_gray    = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        arr = {y:y+h, x:x+w}
        print (arr)

        print ('X :' +str(x))
        print ('Y :'+str(y))
        print ('x+w :' +str(x+w))
        print ('y+h :' +str(y+h))

        xx = int(x+(x+h))/2
        yy = int(y+(y+w))/2

        print (xx)
        print (yy)

        center = (xx,yy)
```

```
            print("Center of Rectangle is :", center)
            print(xx)
            print(yy)
            data = "X{0:d}Y{1:d}Z".format(int(xx), int(yy))
            print ("output = '" +data+ "'")
            arduino.write(data.encode())


    cv2.imshow('img',img)

    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
```

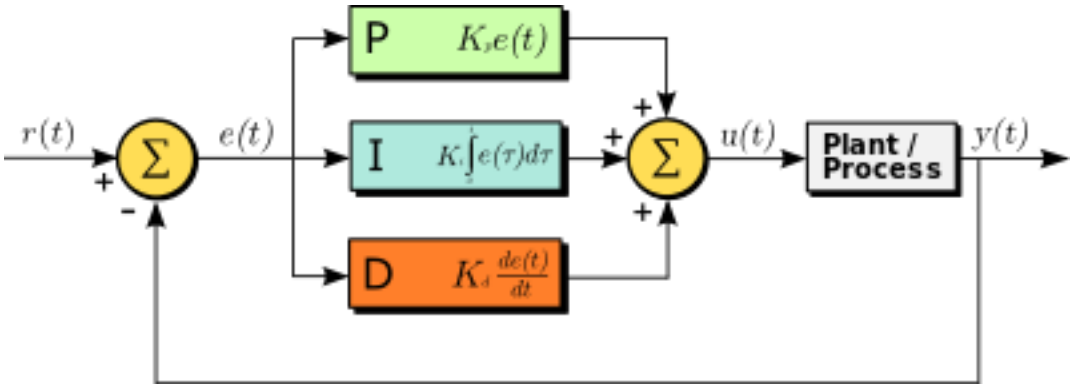
Arduino 在收到坐标后对两个伺服电机进行控制，让摄像头对准用户。

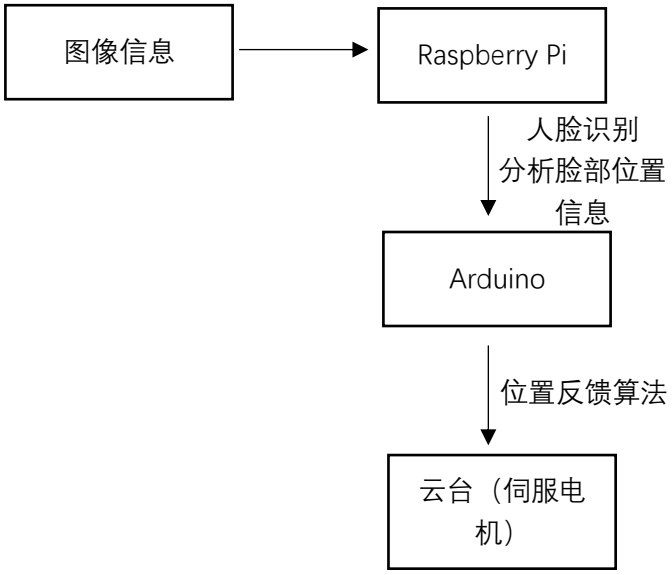
```
#include<Servo.h>

Servo servoVer; //Vertical Servo
Servo servoHor; //Horizontal Servo

int x;
int y;
int servoX = 90;
int servoY = 90;

int prevX;
int prevY;

void setup()
{
  Serial.begin(9600);
  servoVer.attach(5); //横的伺服电机连接 Pin 5
  servoHor.attach(6); //俯仰伺服电机连接 Pin 6
}

void Pos()
{
  if(prevX != x || prevY != y)
  {
```

```
      //int servoX = map(x, 640, 0, 70, 179);
      //int servoY = map(y, 475, 0, 95, 179);
      int dx = x - 320;
      servoX = servoX - dx*0.01;
      int dy = y - 474/2;
      servoY = servoY - dy*0.01;


      servoX = min(servoX, 179);
      servoX = max(servoX, 0);
      servoY = min(servoY, 179);
      servoY = max(servoY, 0);

      servoHor.write(servoX);
      servoVer.write(servoY);
   }
}

void loop()
{
   if(Serial.available() > 0)
   {
      if(Serial.read() == 'X')
      {
         x = Serial.parseInt();
         if(Serial.read() == 'Y')
         {
            y = Serial.parseInt();
           Pos();
         }
      }
      while(Serial.available() > 0)
      {
         Serial.read();
      }
   }
}
```

后续改进：

目前的方案在用户快速移动时会出现无法实时跟踪的问题，可续可使用 PID 算法优化控制算法。





五 结论

我们使用 arduino，树莓派，摄像头，舵机云台，基于 opencv 和 python 所实现的人脸追踪项目，在识别和追踪精度上都有了较为不错的完成度。但由于时间和知识储备的局限，我们最后的追踪还是有点慢，帧数也比较低，但可以通过后期修改算法再进一步提升。

六 成果应用前景

随着时代的发展，人脸识别和追踪这一类生物识别技术已经逐渐成为热门和重点发展的技术。这类技术的高精度，易于使用，稳定性高，难仿冒安全的特性，决定了其广阔的应用前景。它可以应用于 VR,AR 等应用程式，在生活中，在会议，关口安检等方面也有重要作用。

七 附录

视频见附件