



基于摄像头的手势控制小车

谢作如 浙江省温州中学

用手势控制小车,大家往往喜欢用加速度传感器来实现。最简单的做法莫过于在手机上装一个App。这个App能检测手机的运动方向,然后转化为控制信息发给小车。

而我总是希望用一种更加有趣、更加自然的方式去控制小车。微软的Kinect出来后,就有专家写了一本书——《用最自然的方式与机器对话》。的确,在Kinect的支持下,手势控制变得非常简单,其互动游戏很受人欢迎(如图1)。但对于手势识别来说,Kinect并不是唯一的选择,用普通摄像头我们也能够实现。

● 原理分析

在上一期文章中,我们已经大致体验了Processing的功能。作为一款以互动为特色的开源编程语言,Processing拥有大量的扩展库,并且提供了各种具体的应用范例。其中视频库支持外部摄像头,可以实时获取视频画面中的每一个像素点,并进行分析。在有效的算法支持下,我们也可以做到图像分割、物体识别、人脸识别、动作识别、运动跟踪等。同时,Processing支持串口编程,可以通过串口和外部设备(如Arduino)通讯。这些类似的应用,在Processing范

例库中都能够找到,技术门槛并不高。用手势控制小车这件作品的运行原理如图2所示。

但由于我们使用的是普通摄像头,当然实现不了如Kinect的动作识别效果。虽然从理论上说,Processing也能实时识别出人的手势,但是算法非常复杂,不是中小学的学生能够理解的。所以,我试着用另一种简单的方式去实现,即让小车的操控者带上一只特殊颜色的手套或者手上拿一个特定颜色的LED灯,Processing对视频画面的像素点进行遍历,找到这一颜色后,根据坐标对小车发出控制信息。简单地说,就是实现一个能够捕捉视频中特定颜色的程序。

视频中特定颜色和控制信息的关系具体显示为:图3为小车前进,图4为小车后退,图5为小车左转,图6为小车右转,画面中小矩形标志的表示识别出



图1

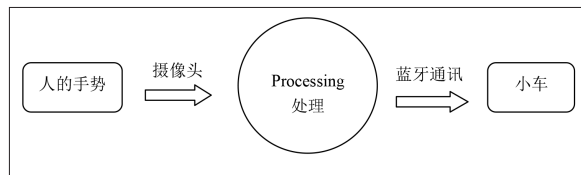


图2

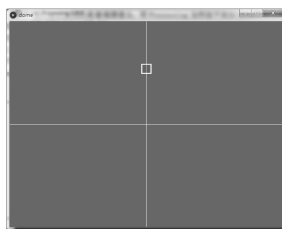


图3

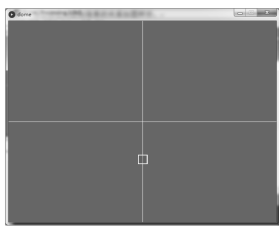


图4

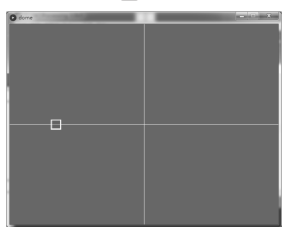


图5

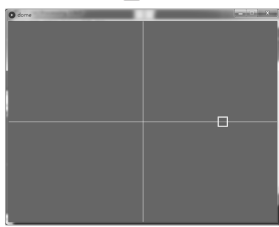


图6

来的颜色。

按照这样的方式定义“上、下、左、右”的控制信息比较容易,也比较好理解。但是,如果坐标在左上角该怎么处

理?如图7所示。

如果给小车装上指南针,应该可以实现非常酷的功能。我们可以想象,特定颜色的坐标在画面的不同方位,就代表控制小车往哪个方向前进。试想一下“走你”的姿势吧,手一挥,小车如离弦的箭一样冲向目标,这种操控的乐趣怎一个“酷”字了得。而且,编程也不会太复杂,Processing只要将这个坐标相对于画面中心点的角度发给小车即可。问题是我在写这个范例时,手头并没有指南针(电子罗盘)模块,只好放弃这一想法。对这种情况,我就当作左转来处理。同时,我把画面上的区域做了规划,设定中间的区域为“停止”(如图8)。

● 制作一辆蓝牙小车

我的手头有好多DFRobot公司的MiniQ小车底盘,来自我的课程“Arduino创意机器人”中使用的机器人套件。“Arduino创意机器人”课程已经全部开源,关于小车的搭建步骤详见我们的课程资源。当然,只要加上一块蓝牙模块,任何一辆Arduino小车都可以。图9为带蓝牙的MiniQ小车。

一般而言,蓝牙模块有4条线需要和Arduino连接,分别为5V、GND、TX和RX。5V和GND要和Arduino上的5V和GND一一对应,但是RX接在Arduino的TX上, TX要接在Arduino的RX上。因为我选择的蓝牙模块也是DFRobot公司的,小车使用的控制板为RoMeo V1,上面已经集成了蓝牙模块接口,插上就可以直接使用,不用另外接线。

对于小车的编程,我使用Mixly。Mixly(中文名为米思齐)是北师大创

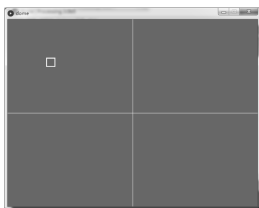


图7

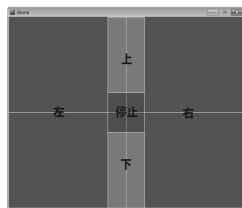


图8

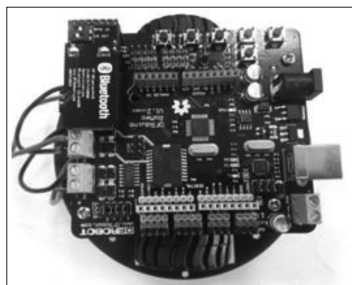


图9

客教育实验室推出的一款新的Arduino图形化编程软件,基于Blockly开发,界面友好,功能比ArduBlock更加强大。Mixly软件的界面如图10所示。

Mixly使用Java

开发,需要先安装

Java环境,建议安装Java 8.0。Mixly为绿色软件,解压后运行Mixly.jar文件即可。Mixly的下载地址:<http://maker.bnu.edu.cn/>,最新版本为0.94。Java的下载地址:http://www.java.com/zh_CN/download/。

Mixly的代码如图11所示。我用1、2、3、4键来表示“上、下、左、右”。下载代码后,建议大家先用手机测试控制是否正常。当然,你的手机要装有蓝牙串口助手之类的App。

● Processing编程

Processing编程是这件作品的重点所在,在“原理分析”部分我已经具体介绍了这个程序的核心功能,即获取摄像头画面中某个色彩,然后分析该色彩的坐标,根据所在区域,通过串口发送控制信息给小车。

代码中的frameRate(8)指画面的

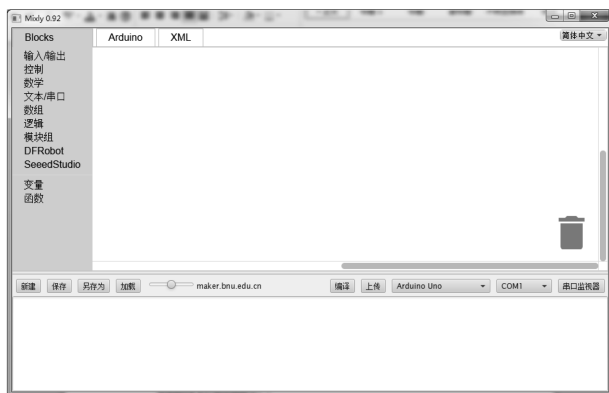


图10

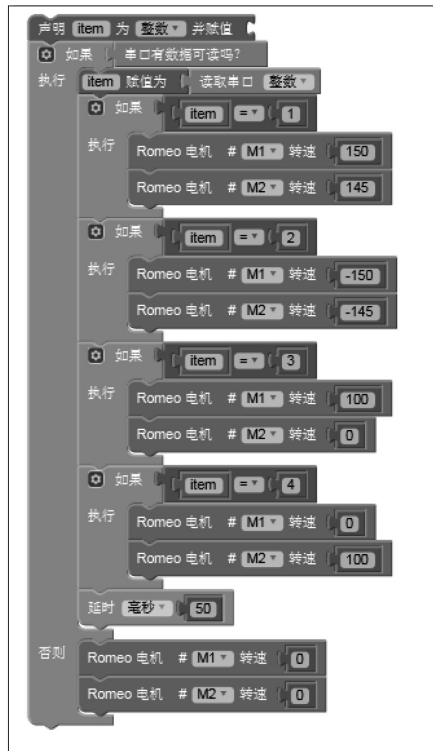


图11

帧数,太小了有延时,太大了又会占用太多的CPU资源。我用cnt变量来控制每秒钟发送的控制信息,这需要在实

```

import processing.video.*;
import processing.serial.*;
Serial myPort; // Create object from Serial class
Capture video;
color findColor;
int mydiff, distance, cnt;
void setup() {
    String portName = Serial.list()[0];
    myPort = new Serial(this, portName, 9600);
    size(640, 480); // Change size to 320 x 240 if too slow at 640 x 480
    frameRate(8);
    video = new Capture(this, 640, 480, 30);
    video.start();
    smooth();
    findColor=color(255, 0, 0);
    mydiff=15;
    distance=100;//
    cnt=0;
}
void draw() {
    if (video.available()) {
        video.read();
    }
    video.loadPixels();
    image(video, 0, 0, width, height);
    fill(255, 0, 0, 50);
    stroke(255);
    rect((width-distance)/2, (height-distance)/2, distance, distance);
    float difference=300;
    float min_d=300;//记录最小的容差数
    int miniX=0, miniY=0;
    for (int x=0; x<width; x++) {
        for (int y=0; y<height; y++) {
            int loc=x*y*video.width;
            color c=video.pixels[loc];
            float r1=red(c);
            float g1=green(c);
            float b1=blue(c);
            float r2=red(findColor);
            float g2=green(findColor);
            float b2=blue(findColor);
            float d=dist(r1, g1, b1, r2, g2, b2);//开始比较
            if (d<difference) {
                difference=d;
            }
            if (min_d>difference) {
                min_d=difference;
                miniX=x;
                miniY=y;
            }
        }
    }
    if (difference < mydiff && cnt % 4==0) {
        fill(findColor);
        stroke(255);
        rect(miniX, miniY, 60, 60);
        if (miniX<(width/2-distance)) {
            myPort.write("3/n");
        } else if (miniX>(width/2+distance)) {
            myPort.write("4/n");
        } else if (miniY<(height/2-distance/2)) {
            myPort.write("1/n");
        } else if (miniY>(height/2+distance/2)) {
            myPort.write("2/n");
        }
    }
    cnt++;
}
void mousePressed() {
    int loc=mouseX*video.width;//选择颜色
    findColor=video.pixels[loc];
}

```

图12

践中不断地调试(如图12)。

● 结语

在实际测试中,我发现自己对小车的控制不是很流畅,还需要进一步优化。但从用手机控制小车的角度来看,我还是达到了预设的目标。当然,如果你对视觉识别感兴趣,最好还是研究OpenCV,这也是一个著名的开源跨平台计算机视觉库,能实现更多的功能。[e](#)

如果对相关内容感兴趣,请关注主持人博客。

