工作目标：机器人携带单目摄像头，跟随某感兴趣区域（红色上衣人），显示距离以及偏离角度

工作方案：

通过颜色提取出大致区域（→形态学处理）→轮廓提取→利用轮廓大小关系找到目标框→获得图片中心点到离框最近点的距离和角度→根据相似三角形计算实际偏离角度及距离工作环境：

编辑器visual studio code1.24.0

python版本 3.6

所需包：cv2,numpy,skimage,math

工作细节：

前两周工作内容主要是熟悉python以及对于整个方案的明确，初步写出一期代码；第三周主要根据讨论结果对于感兴趣区域的提取实现采用矩阵思想来做，以便实现与他人工作接口实现。

细节与总结：

1. 如何将一张带有红色区域的照片转换成0-1这样的二值矩阵（其中0，1表示的是红色和非红色），实际上需要解决的是python如何处理图片并保存到txt

参考链接：<https://jingyan.baidu.com/article/3ea514898bb41152e61bbada.html>

1. 遍历矩阵，找到重心



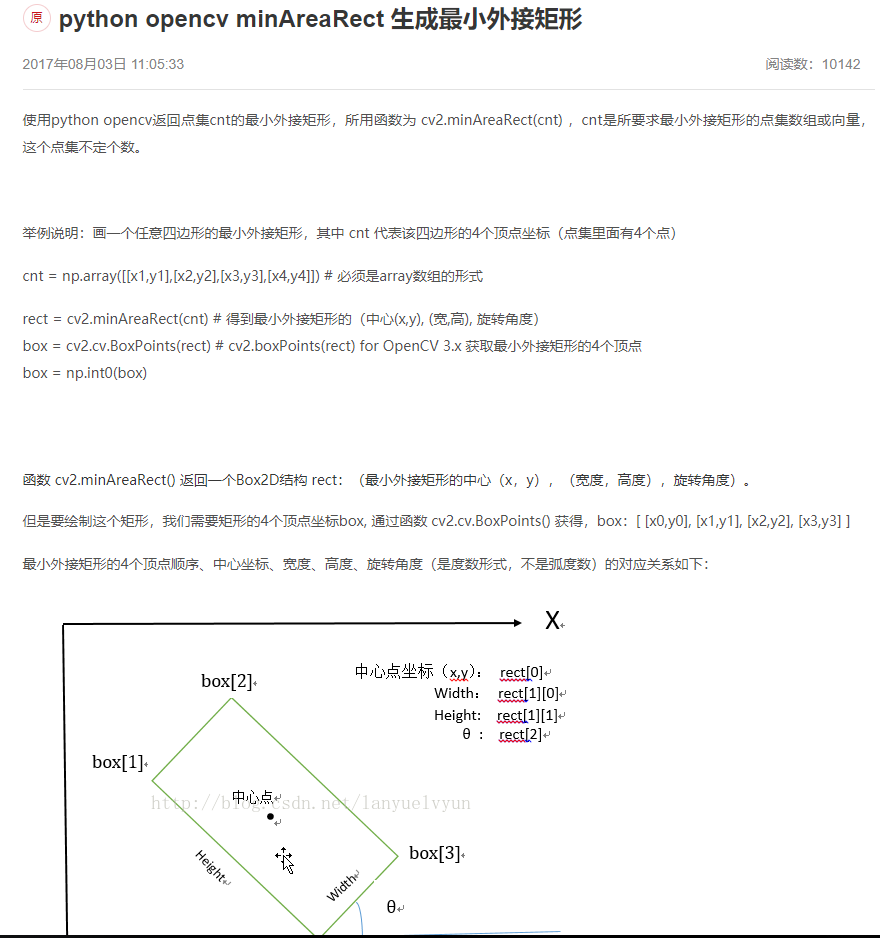
1. 对于矩阵进行遍历，这一部分需要注意numpy中的ndarray类型表示与坐标系之间的出入，也就是python如何查找二维数组的相邻元素

参考链接：

<https://zhidao.baidu.com/question/1829819050746480860.html?fr=iks&word=neighbors%28self%2Carray%2C+radius%2C+x%2C+y%29%3A&ie=gbk>

1. 找到离重心点最短距离，利用重心点往外进行半径的扩张，也就是扩散算法，如果该尺度不合适，便继续增长

5.找到四个关键点，画出轮廓的最小外接矩形



6.对图片进行合适的处理

##### 用相似三角形计算物体或者目标到相机的距离

我们将使用相似三角形来计算相机到一个已知的物体或者目标的距离。

相似三角形就是这么一回事：假设我们有一个宽度为 W 的目标或者物体。然后我们将这个目标放在距离我们的相机为 D 的位置。我们用相机对物体进行拍照并且测量物体的像素宽度 P 。这样我们就得出了相机焦距的公式：

F = (P x D) / W

举个例子，假设我在离相机距离 D = 24 英寸的地方放一张标准的 8.5 x 11 英寸的 A4 纸（横着放；W = 11）并且拍下一张照片。我测量出照片中 A4 纸的像素宽度为 P = 249 像素。

因此我的焦距 F 是：

F = (248px x 24in) / 11in = 543.45

当我继续将我的相机移动靠近或者离远物体或者目标时，我可以用相似三角形来计算出物体离相机的距离：

D’ = (W x F) / P

为了更具体，我们再举个例子，假设我将相机移到距离目标 3 英尺（或者说 36 英寸）的地方并且拍下上述的 A4 纸。通过自动的图形处理我可以获得图片中 A4 纸的像素距离为 170 像素。将这个代入公式得：

D’ = (11in x 543.45) / 170 = 35 英寸

或者约 36 英寸，合 3 英尺。

**如何提取出红色区域？**

**方案一：**

使用PIL（Pillow,是python平台事实上的图像处理标准库）截取图像，然后将RGB转为HSV进行判断，统计判断颜色，最后输出RGB值。

参考链接：https://www.jb51.net/article/62526.htm

解释：

HSV是一种将RGB色彩空间中的点在倒圆锥体中表示的方法。HSV也就是色调，饱和度和明亮度。各通道的取值都是0~360，可是在我们的opencv中H的值为0~180，因为COLOR\_BGR2HSV在转换图像时是将H/2--🡪H.

**方案二：**

也就是本文的处理思路，另外，给出网上HSV颜色分类图

