周围的人都比你厉害，你才会慢慢变强

OpenCV---直线检测

目录

直线检测相关

Opencv学习笔记-----霍夫变换直线检测及原理理解

OpenCV-Python教程（9、使用霍夫变换检测直线）

Hough变换是经典的检测直线的算法。其最初用来检测图像中的直线，同时也可以将其扩展，以用来检测图像中简单的结构。

随笔档案

最新评论

1. Re:python---websocket的使用
网上的方法都不行，
换成gbk会报如下的错：
IndexError: index out of range
--缘分天空0320

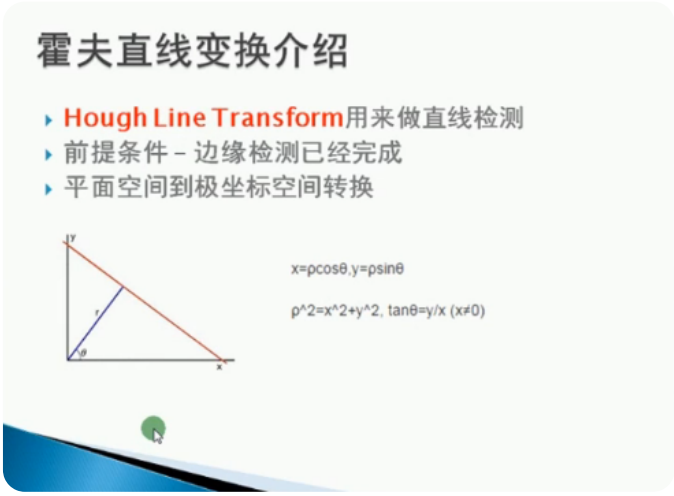
2. Re:python---websocket的使用
@ 缘分天空0320太久没用，忘了。应该是字符编码问题吧。这类问题网上应该可以很容易找到方法解决。你看看Python的默认编码和代码是不是一致。一般就是gbk和utf8之间出错...
--山上有风景

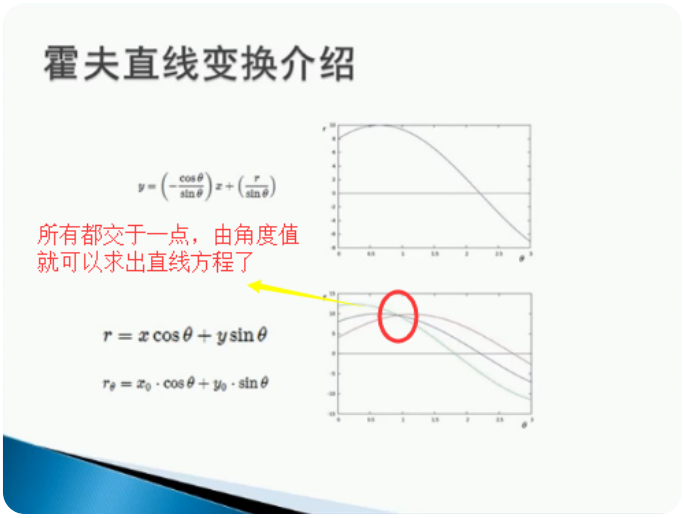
3. Re:python---websocket的使用
您好，用了您的代码，报如下错误，麻烦问下如何解决呢？Traceback (most recent call last): File "server3.py", line 101, in <module...
--缘分天空0320

4. Re:数据结构（六）查找---线性索引查找
请问最后倒序排序的那个代码怎么实现的？可以发一下吗？
--Viki-

5. Re:SDN实验---Ryu的源码分析
@ 山上有风景谢谢！...
--iRoy_33

阅读排行榜





变换图示

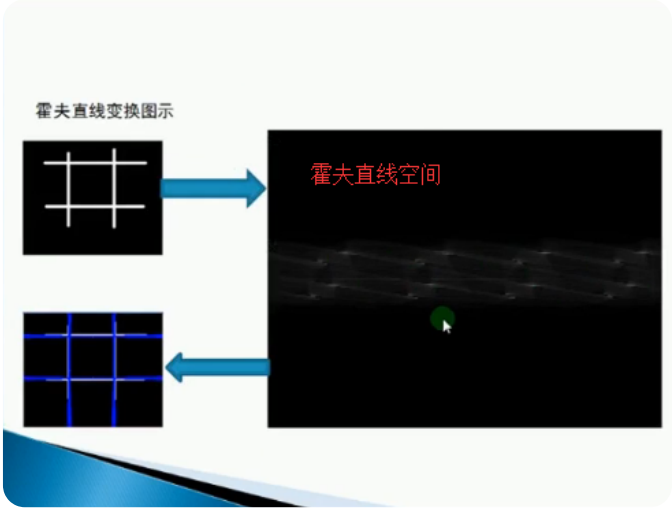
2. OpenCV---图像二值化(12513)
3. OpenCV---模板匹配matchTemplate(11534)
4. OpenCV---直线检测(8208)
5. python---基础知识回顾（九）图形用户界面-------WxPython(7986)

评论排行榜

1. python---基础知识回顾（九）图形用户界面-------Tkinter(4)
2. python---websocket的使用(3)
3. SDN实验---Ryu的源码分析(3)
4. 数据结构（三）串---KMP模式匹配算法之获取next数组(2)
5. 数据结构（四）树---树的存储结构(1)

推荐排行榜

1. 数据结构（七）排序---堆排序(11)
2. python---aiohttp的使用(6)
3. python---websocket的使用(4)
4. python---基础知识回顾（九）图形用户界面-------WxPython(3)
5. Python图像处理库PIL中图像格式转换(3)



霍夫直线检测的两种方法

1.获取灰度图像
2.canny边缘检测
3.获取霍夫直线信息
4.算出直线位置，画出每条直线

一：HoughLines霍夫变换

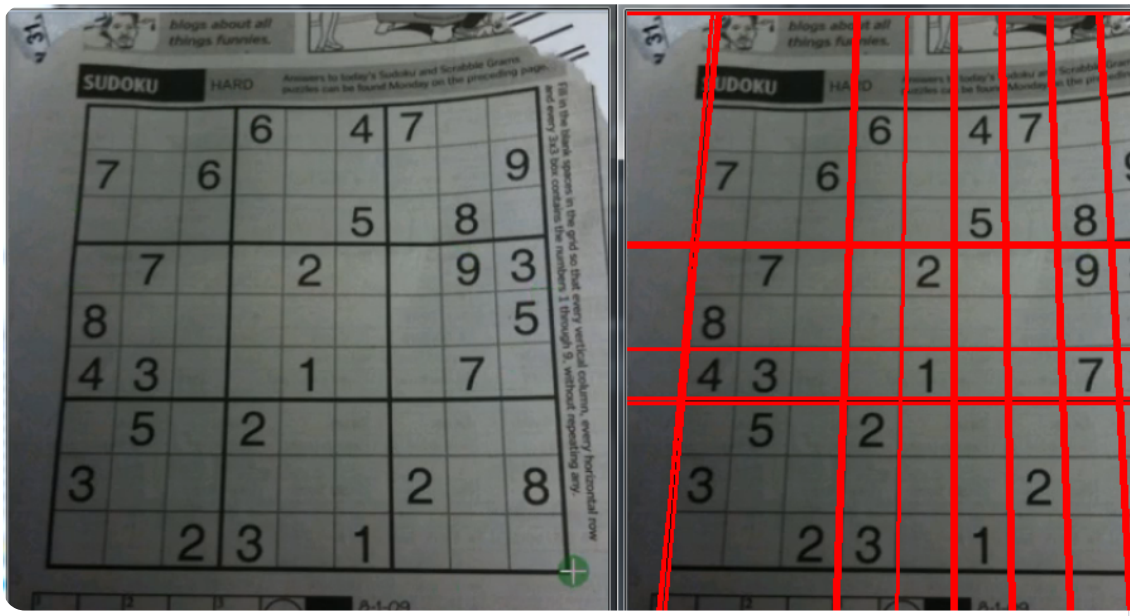
```
def line_detection(image):
    gray = cv.cvtColor(image,cv.COLOR_BGR2GRAY)
    edges = cv.Canny(gray,50,150,apertureSize=3)     #apertureSize是sobel算子大小，只能为1,3,5, 7
    lines = cv.HoughLines(edges,1,np.pi/180,200)   #函数将通过步长为1的半径和步长为π/180的角来搜索所有可能的直线
    for line in lines:
        rho,theta = line[0]    #获取极值ρ长度和θ角度
        a = np.cos(theta)    #获取角度cos值
        b = np.sin(theta)    #获取角度sin值
        x0 = a * rho    #获取x轴值
        y0 = b * rho    #获取y轴值    x0和y0是直线的中点
        x1 = int(x0 + 1000*(-b))    #获取这条直线最大值点x1
        y1 = int(y0 + 1000*(a))     #获取这条直线最大值点y1
        x2 = int(x0 - 1000 * (-b))  #获取这条直线最小值点x2
        y2 = int(y0 - 1000 * (a))   #获取这条直线最小值点y2    其中*1000是内部规则
        cv.line(image,(x1,y1),(x2,y2),(0,0,255),2)    #开始划线
    cv.imshow("image line",image)



src = cv.imread("./1.png")  #读取图片
cv.namedWindow("input image",cv.WINDOW_AUTOSIZE)     #创建GUI窗口,形式为自适应
cv.imshow("input image",src)     #通过名字将图像和窗口联系

line_detect_possible_demo(src)

cv.waitKey(0)    #等待用户操作，里面等待参数是毫秒，我们填写0，代表是永远，等待用户操作
cv.destroyAllWindows()  #销毁所有窗口
```



相关知识补充

（一）HoughLines方法

```
def HoughLines(image, rho, theta, threshold, lines=None, srn=None, stn=None, min_theta=None, max_theta=None):
# real signature unknown; restored from __doc__
```

```
cv.HoughLines(edges,1,np.pi/180,200)
```

cv2.HoughLines函数输出的是[float, float]形式的ndarray，**其中每个值表示检测到的线(ρ , θ)中浮点点值的参数。**

第一个参数image：是canny边缘检测后的图像

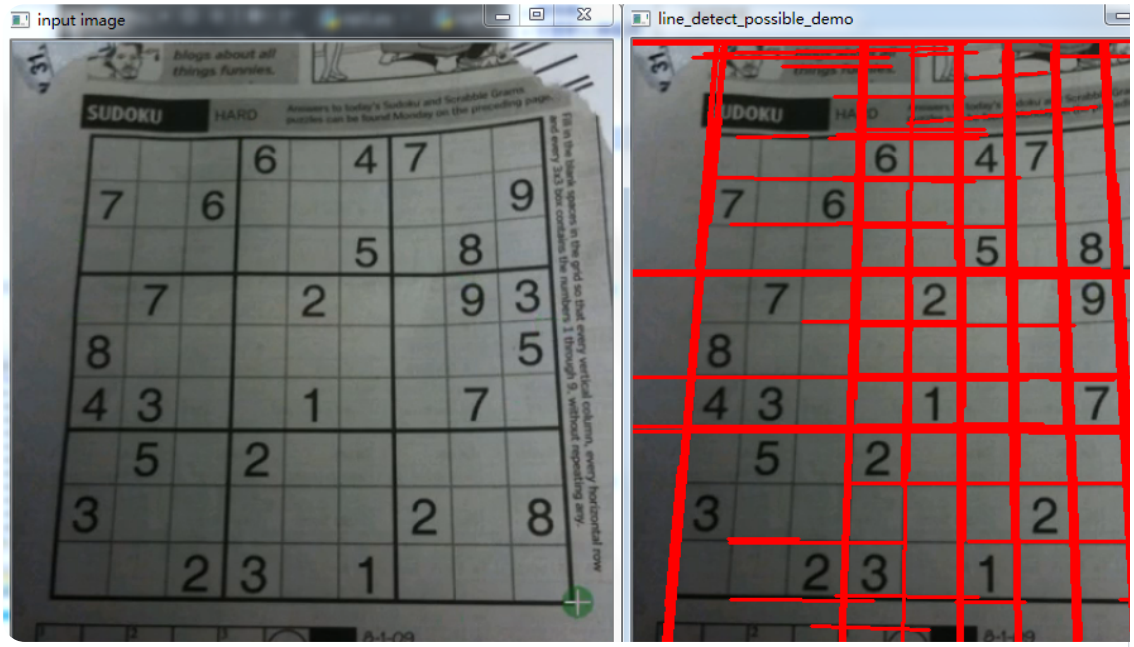
第二个参数rho和第三个参数theta：对应直线搜索的步长。在本例中，函数将通过步长为1的半径和步长为π/180的角来搜索所有可能的直线。

最后一个参数threshold：是经过某一点曲线的数量的阈值，超过这个阈值，就表示这个交点所代表的参数对(rho, theta)在原图像中为一条直线

观察前面的例子得到的结果图片，其中Hough变换看起来就像在图像中查找对齐的边界像素点集合。
但这样会在一些情况下导致虚假检测，如像素偶然对齐或多条直线穿过同样的对齐像素造成的多重检测。

二：HoughLinesP概率霍夫变换（是加强版）使用简单，效果更好，检测图像中分段的直线（而不是贯穿整个图像的直线）

```
def line_detect_possible_demo(image):
    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    edges = cv.Canny(gray, 50, 150, apertureSize=3)  # apertureSize是sobel算子大小，只能为1,3,5, 7
    lines = cv.HoughLinesP(edges, 1, np.pi / 180, 100, minLineLength=50,maxLineGap=10)  #函数将通过步长为1的半径和步长为π/180的角来搜索所有可能的直线
    for line in lines:
        print(type(line))   #多维数组
        x1,y1,x2,y2 = line[0]
        cv.line(image,(x1,y1),(x2,y2),(0,0,255),2)
    cv.imshow("line_detect_possible_demo",image)
```



相关知识补充：

（一）HoughLinesP方法

```
def HoughLinesP(image, rho, theta, threshold, lines=None, minLineLength=None, maxLineGap=None): # real signature unknown; restored from __doc__
```

```
cv.HoughLinesP(edges, 1, np.pi / 180, 100, minLineLength=50,maxLineGap=10)
```

第一个参数是需要处理的原图像，该图像必须为cannay边缘检测后的图像；

第二和第三参数：步长为1的半径和步长为π/180的角来搜索所有可能的直线

第四个参数是阈值，概念同霍夫变换

第五个参数：minLineLength-线的最短长度，比这个线短的都会被忽略。

第六个参数：maxLineGap-两条线之间的最大间隔，如果小于此值，这两条线就会被看成一条线。

这个函数的返回值就是直线的起点和终点。

作者：山上有风景

欢迎任何形式的转载，但请务必注明出处。

限于本人水平，如果文章和代码有表述不当之处，还请不吝赐教。

分类： OpenCV

好文要顶 关注我 收藏该文

山上有风景
关注 - 19
粉丝 - 170
+加关注

« 上一篇： OpenCV---Canny边缘提取
» 下一篇： OpenCV---圆检测

1 0
关注 | 顶部 | 评论

posted @ 2018-07-06 22:48 山上有风景 | 收藏