



周围的人都比你厉害，你才会慢慢变强

公告

昵称： 山上有风景
园龄： 1年11个月
粉丝： 170
关注： 19
+加关注

< 2019年12月 >

日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- STL(18)
- SDN(15)
- ThinkPHP3.2(1)

积分与排名

积分 - 211835
排名 - 1998

随笔分类

- C/C++(74)
- Html(2)
- Java(33)
- Javascript(19)
- OpenCV(29)
- PHP(2)
- Python(155)
- STL泛型编程(18)
- 单片机笔记（复习用）(3)
- 计算机网络(32)
- 其他知识(14)

OpenCV---对象测量

目录

- 一：获取图像的外接矩形boundingRect和几何距moments
- 二：多边形逼近approxPolyDP
- 相关知识补充
 - (一) 相关函数方法
 - (二) approxPolyDP轮廓逼近方法
- 返回值

弧长与面积

- 轮廓发现
- 计算每个轮廓的弧长与面积, 像素单位

多边形拟合

- 获取轮廓的多边形拟合结果
- approxPolyDP
 - contour
 - epsilon 越小 越折线越逼近真实形状
 - close - 是否为闭合区域

2 0

关注 | 顶部 | 评论

设计模式(27)
数据结构(57)
数据库(8)
算法习题(43)
算法训练营
随笔所想(4)
图形界面编程
正则表达式(2)
转载推文(4)

随笔档案

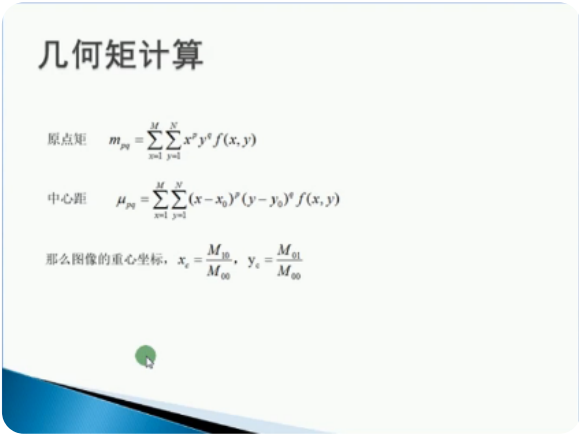
2019年11月(5)
2019年10月(32)
2019年9月(21)
2019年7月(10)
2019年5月(8)
2019年4月(25)
2019年3月(8)
2019年2月(1)
2019年1月(12)
2018年12月(19)
2018年9月(5)
2018年8月(95)
2018年7月(78)
2018年6月(26)
2018年5月(17)
2018年4月(22)
2018年3月(111)

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，麻烦问一下如何解决呢？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

1. python---websocket的使用(17253)



一：获取图像的外接矩形boundingRect和几何距moments

```
import cv2 as cv
import numpy as np

def measure_object(image):
    gray = cv.cvtColor(image,cv.COLOR_BGR2GRAY) #转灰度图像
    ret, binary = cv.threshold(gray,0,255,cv.THRESH_OTSU|cv.THRESH_BINARY_INV) #获取二值化图像
    print("thresold value:",ret)
    cv.imshow("binary image",binary)
    outImage,contours,hireachy = cv.findContours(binary,cv.RETR_EXTERNAL,cv.CHAIN_APPROX_SIMPLE)
    for i,contour in enumerate(contours):
        area = cv.contourArea(contour) #获取每个轮廓面积
        x,y,w,h = cv.boundingRect(contour) #获取轮廓的外接矩形
        rate = min(w,h)/max(w,h) #获取外接矩形宽高比，可以起到一定的筛选作用
        print("rectangle rate:%s"%rate)
        mm = cv.moments(contour) #求取轮廓的几何距
        print(type(mm))
        print(mm)
        cx = mm['m10']/mm['m00']
        cy = mm['m01']/mm['m00']
        cv.circle(image,(np.int(cx),np.int(cy)),2,(0,255,255),-1) #根据几何距获取的中心点，画出中心圆
        cv.rectangle(image,(x,y),(x+w,y+h),(0,0,255),2) #根据轮廓外接矩形返回数据，画出外接矩形
        print("contour area:",area)
    cv.imshow("measure_object",image)

src = cv.imread("./dg.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

measure_object(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
```

```
mm-->{'m02': 3301.25, 'm12': 233530.03333333333, 'm30': 6934377.600000001, 'mu11': -10.721153846156085,
'm21': 1265395.2, 'nu02': 0.08336283484212434, 'm01': 252.5, 'mu30': -12.458842864260077, 'nu21':
0.003451665836783194, 'mu12': 11.46577909273492, 'nu12': 0.006828369938345303, 'mu02': 31.698717948717785,
'mu21': 5.795825115486707, 'm03': 43556.350000000006, 'nu30': -0.007419782588937784, 'm20':
97840.416666666666, 'm00': 19.5, 'm10': 1381.0, 'nu20': 0.09806301522275791, 'mu03': -11.52064431294275,
'm11': 17871.458333333332, 'nu03': -0.0068610445623109115, 'mu20': 37.2884615384537,
-0.028195013402119877}
```

20

关注 | 顶部 | 评论

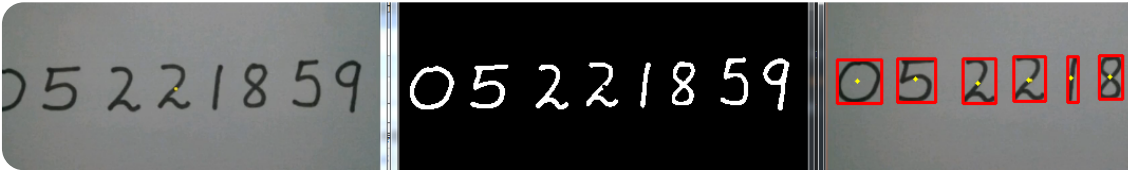
- 2. OpenCV---图像二值化(12513)
- 3. OpenCV---模板匹配matchTemplate(11534)
- 4. OpenCV---直线检测(8208)
- 5. python---基础知识回顾 (九) 图形用户界面-----WxPython(7986)

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)



二：多边形逼近approxPolyDP

```
import cv2 as cv
import numpy as np

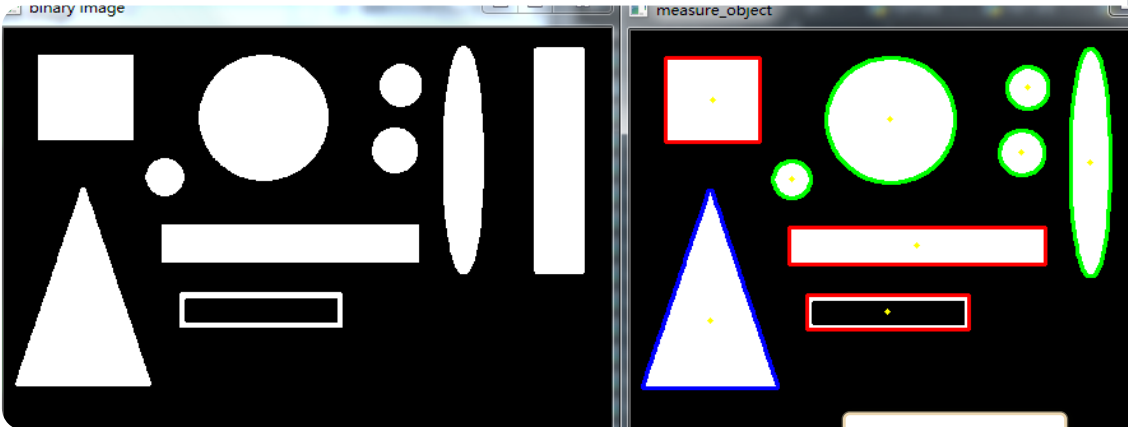
def measure_object(image):
    gray = cv.cvtColor(image,cv.COLOR_BGR2GRAY) #转灰度图像
    ret, binary = cv.threshold(gray,0,255,cv.THRESH_OTSU|cv.THRESH_BINARY) #获取二值化图像
    cv.imshow("binary image",binary)
    dst = cv.cvtColor(binary,cv.COLOR_GRAY2BGR)
    outImage,contours,hierarchy = cv.findContours(binary,cv.RETR_EXTERNAL,cv.CHAIN_APPROX_SIMPLE)
    for i,contour in enumerate(contours):
        mm = cv.moments(contour) #求取轮廓的几何距
        cx = mm['m10']/mm['m00']
        cy = mm['m01']/mm['m00']
        cv.circle(dst, (np.int(cx),np.int(cy)), 2, (0,255,255), -1) #画出中心点
        approxCurve = cv.approxPolyDP(contour,4,True) #4是与阈值的间隔大小，越小越易找出，True是否找闭合图像
        print(approxCurve.shape)
        if approxCurve.shape[0] >= 7:
            cv.drawContours(dst,contours,i, (0,255,0), 2) #画出轮廓
        elif approxCurve.shape[0] == 4:
            cv.drawContours(dst,contours,i, (0,0,255), 2)
        else:
            cv.drawContours(dst,contours,i, (255,0,0), 2)

    cv.imshow("measure_object",dst)

src = cv.imread("./1k.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

measure_object(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) 相关函数方法

20

关注 | 顶部 | 评论

```
cv.contourArea(contour)      #获取每个轮廓面积
cv.boundingRect(contour)     #获取轮廓的外接矩形
cv.moments(contour)          #求取轮廓的几何距
cv.arcLength(contour,True)   #求取轮廓的周长, 指定闭合
```

(二) approxPolyDP轮廓逼近方法

```
approxCurve = cv.approxPolyDP(contour,4,True)    #4是与阈值的间隔大小, 越小越易找出, True是是否找闭合图像
```

```
def approxPolyDP(curve, epsilon, closed, approxCurve=None): # real signature unknown; restored from __doc__
```

第一个参数curve: 输入的点集, 直接使用轮廓点集contour
第二个参数epsilon: 指定的精度, 也即是原始曲线与近似曲线之间的最大距离。
第三个参数closed: 若为true, 则说明近似曲线是闭合的, 反之, 若为false, 则断开。
第四个参数approxCurve: 输出的点集, 当前点集是能最小包容指定点集的。画出来即是一个多边形;

返回值

```
approxCurve: 输出的点集, 当前点集是能最小包容指定点集的。画出来即是一个多边形;
```

```
print(approxCurve)      #打印每个轮廓的特征点
print(approxCurve.shape) #打印该点集的shape, 第一个数是代表了点的个数, 也就是边长连接逼近数
```

```
[[[138 208]]

 [[138 234]]

 [[265 234]]

 [[264 207]]]
(4, 1, 2)    #矩形, 四个点逼近图像
[[[124 154]]

 [[124 183]]

 [[325 183]]

 [[325 154]]]
(4, 1, 2)
[[[ 61 125]]

 [[ 9 279]]

 [[114 280]]]
(3, 1, 2)    #3是三角形
[[[123 102]]

 [[111 118]]

 [[117 128]]

 [[129 131]]

 [[139 124]]

 [[141 114]]

 [[134 104]]]
(7, 1, 2)    #7, 8, 9都是圆形类
[[[303 78]]

 [[291 87]]

 [[289 99]]

 [[298 111]]
```

2 0

关注 | 顶部 | 评论

```
[[310 113]]

[[322 105]]

[[324 91]]

[[314 79]]]
(8, 1, 2)
[[[308 28]]

[[298 35]]

[[295 48]]

[[303 59]]

[[314 61]]

[[325 54]]

[[327 39]]]
(7, 1, 2)
[[[198 21]]

[[165 38]]

[[153 75]]

[[170 107]]

[[185 116]]

[[210 119]]

[[244 100]]

[[254 63]]

[[235 31]]]
(9, 1, 2)
[[[ 27 21]]

[[ 27 87]]

[[101 87]]

[[101 21]]]
(4, 1, 2)
[[[418 15]]

[[416 190]]

[[454 192]]

[[455 16]]]
(4, 1, 2)
[[[359 14]]

[[346 58]]

[[345 138]]

[[356 190]]

[[363 192]]

[[374 156]]
```

20

关注 | 顶部 | 评论

```
[[376 82]]

[[369 24]]]
(8, 1, 2)
```

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---轮廓发现
» 下一篇： OpenCV---膨胀与腐蚀

posted @ 2018-07-07 11:14 山上有风景 阅读(741) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)， [访问](#) 网站首页。

Copyright © 2019 山上有风景

Powered by .NET Core 3.1.0 on Linux

2 0

[关注](#) | [顶部](#) | [评论](#)