



【小林的OpenCV基础课 12】霍夫变换见实习生



小林同学

中度二次元 非职业代码手

16 人赞了该文章

新学期、学生はどんな計画もありますか？

这一话的结构如下：

- 霍夫线变换API与Demo
- 霍夫圆变换API与Demo

霍夫变换的由来

“霍夫变换于1962年由Paul Hough首次提出，最初的Hough变换是设计用来检测直线和曲线，起初的方法要求知道物体边界线的解析方程，但不需要有关区域位置的先验知识。后于1972年由Richard Duda & Peter Hart推广使用。”

其实，霍夫变换的中心思想就是通过坐标变换来检测直线，后来经过改进，就可以检测椭圆等。

关于霍夫线变换和圆变换的原理，同学们可以参考[【小林的OpenCV基础课 番外】霍夫变换原理](#)

霍夫线变换API与Demo

OpenCV支持三种不同

▲ 赞同 16 ▼

● 4 条评论

➤ 分享

★ 收藏

...

【小林的Op

- 累计概率霍夫变换(Progressive Probabilistic Hough Transform , PPHT), 是SHT的改进版

API:

标准霍夫变换SHT与多尺度霍夫变换MSHT:

```
lines=cv2.HoughLines(image, rho, theta, threshold[, lines[, srn[, stn[, min_theta[, ma
```

- lines: 三维的直线集合, 第一维为某条直线, 第二维和第三维为 对应的 ρ 和 θ
- rho: 以像素为单位的步进精度
- theta: 以角度为单位的旋转精度
- threshold: 阈值, 先将属于同一条直线的像素点个数累加, 累加值大于阈值threshold的线段才可以被检测通过并返回到结果中
- srn: 默认值为0, 此时为标准霍夫变换SHT, 步进精度等于参数rho。不为0时为多尺度霍夫变换MSHT, 步进精度为rho/srn
- stn: 默认值为0, 此时为标准霍夫变换SHT, 旋转精度等于参数theta。不为0时为多尺度霍夫变换MSHT, 旋转精度为theta/stn
- min_theta和max_theta有各自的默认值, 可以无视。

注意了: 当参数srn和stn同时为0时为标准霍夫变换, 否则为多尺度霍夫变换。

累计概率霍夫变换PPHT:

这个方法通过随机采样计算, 可以降低运算量, 提高处理速度, 但有降低检测精度的可能, 在实际应用中应加以权衡。

```
lines=cv2.HoughLinesP(image, rho, theta, threshold[, lines[, minLineLength[, maxLineGa
```

- image: 8为单通道二值化的原图像
- lines: 检测到的线段, 每个line有四个参数组成, 即 (x_1, y_1, x_2, y_2) , 分别对应线段的起始点坐标和终止点坐标
- minLineLength: 最短线段长度, 只有长度大于该参数的线段才会被检测到, 默认值为0
- maxLineGap: 允许将同一行点与点之间连接起来的最大的距离, 默认值为0

Demo: 下面是标准霍夫变换SHT的Demo, 小林将解析直线和画直线封装到了drawfindLines_Hough函数里。

```
import cv2
import numpy as np
```

```
for rho, theta in i:
    a = np.cos(theta)
    b = np.sin(theta)
# 变换到x-y坐标系
    x0 = a * rho
    y0 = b * rho
# 用点和斜率得到直线上的两个点
    x1 = int(x0 + 1000 * (-b))
    y1 = int(y0 + 1000 * (a))
    x2 = int(x0 - 1000 * (-b))
    y2 = int(y0 - 1000 * (a))
    cv2.line(img, (x1, y1), (x2, y2), (0, 0, 255), 1)

img = cv2.imread('Road1.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray, 50, 150, apertureSize = 3)

lines = cv2.HoughLines(edges, 1, np.pi/180, 100)
drawfindLines_Hough(img, lines)

cv2.imshow('Line', img)
cv2.waitKey()
cv2.destroyAllWindows()
```



原图像



效果图

如果使用累计概率霍夫变换PPHT，则可通过下面的代码解析得到起始点和终止点的坐标：

```
# lines为调用HoughLinesP函数得到的结果
linesP = lines[:, 0, :]
for x1, y1, x2, y2 in linesP[:]:
    cv2.line(img, (x1, y1), (x2, y2), (0, 0, 255), 1)
```

霍夫圆变换API与Demo

API:

```
circles=cv2.HoughCircles(image, method, dp, minDist[, circles[, param1[, param2[, minR
```

- image: 原图像，需为8位单通道图像
- circles: 检测到的圆的列表，每个圆都由矢量 (x, y, r) 构成，即圆心和半径
- method: 检测圆时使用的算法，目前只有霍夫梯度法可以使用（文档里也是这么说的），对应cv2.HOUGH_GRADIENT
- dp: 用来检测圆心的累加器图像的分辨率于输入图像之比的倒数。例如，如果dp=1时，累加器和输入图像具有相同的分辨率；如果dp=2，累加器便有输入图像一半那么大的宽度和高度
- minDist: 检测到的圆中，圆心之间的最小距离
- param1: 有默认值100。它是第三个参数method设置的霍夫梯度法对应的参数，它表示传递给canny边缘检测算子的高阈值，而低阈值为高阈值的一半

通过检测的圆就更加接近完美的圆形了

- minRadius和maxRadius：有默认值0，表示圆半径的最小值和最大值

讲真，这个函数并不是很好用，它能较好的检测出圆心，但没法得出满意的半径，因此：

- 修改minRadius和maxRadius得出合适的半径
- 只得出圆心，半径通过其他算法得到

Demo：检测OpenCV的logo中存在的圆：

```
import cv2
import numpy as np

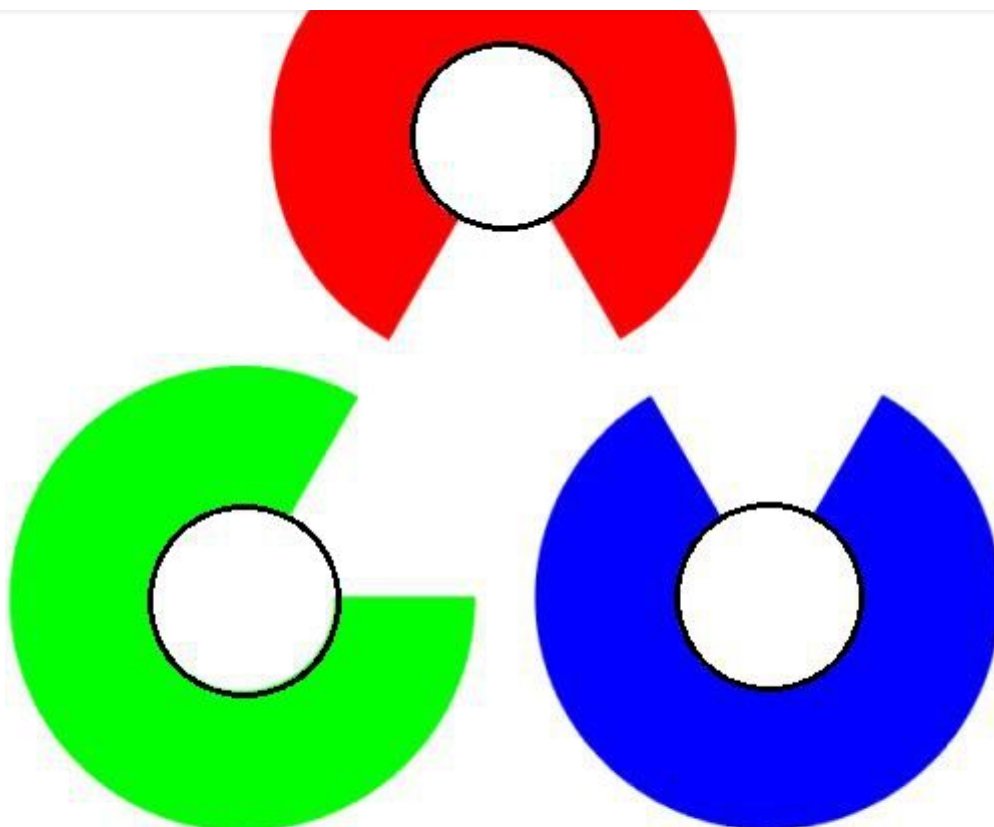
img = cv2.imread('logo2.jpg')
grayImg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

circle = cv2.HoughCircles(grayImg, cv2.HOUGH_GRADIENT, 2, 50)

for c in circle:
    for x,y,r in c:
        cv2.circle(img, (x,y), r, (0, 0, 0), 2)

cv2.imshow('Circle', img)

cv2.waitKey()
cv2.destroyAllWindows()
```



OpenCV

dp = 2

$dp = 1$

下面再安利一张图，同学们可以修改参数来达到满意的效果：

这一话的示例代码已经同步到了[Github](#)，分别对应Class 4 Image Processing文件夹下的C4 Hough Lines.py和C4 Hough Circles.py。

最后的最后

[OpenCV](#)[Python](#)[计算机视觉](#)

文章被以下专栏收录



小林的CV视觉工坊

这里是小林的CV视觉工坊。每周末定时推送OpenCV的入门教程和相关文章，快来跟...

[进入专栏](#)

推荐阅读



【小林的OpenCV基础课 10】Canny边缘检测

小林同学

发表于小林的CV...



【小林的OpenCV基础课 9】非线性滤波与模糊

小林同学

发表于小林的CV...



【小林的OpenCV基础课 8】画轮廓的新人

小林同学

发表于

4 条评论

[⇌ 切换为时间排序](#)

写下你的评论...



小七

加油

👍 赞

8 个月前



Yonaldo

你好。有几个问题：

7 个月前


```
x2 = int(x0 - 1000 * (-b))
```

```
y2 = int(y0 - 1000 * (a))
```

这几步的数学原理是什么，没太理解

(2) 您`lines = cv2.HoughLines(edges, 1, np.pi/180, 100)`设置的`threshold`100似乎并不适用，180为宜

(3) 您定义的函数`drawfindLines_Hough`只接受两个参数，但在调用时输入参数为三个 (`img, lines, 0`)

👍 赞



小林同学 (作者) 回复 Yonaldo

7 个月前

1.是在`x0`和`y0`的基础上在斜率方向扩展得到直线上两点的坐标 2.阈值问题 图像复杂程度会影响阈值效果 不同的需求要设置不同的阈值 3.三个参数的函数版本是我写demo时候新加的功能 后来不需要就删掉了 蟹蟹指出错误 ($\geq \nabla \leq$)

👍 赞



rainbOw 回复 小林同学 (作者)

13 天前

请问1000是什么意思

👍 赞