

OpenCV-python Canny边缘检测

Canny边缘检测: `cv2.Canny()`

1. 原理

噪音去除

第一步噪音去除使用 5×5 的高斯滤波器去除噪音。

计算图像梯度

对平滑后的图像使用 `Sobel` 算子计算水平方向和竖直方向的图像梯度(G_x 和 G_y)。根据得到的这两幅梯度图，找到边界的梯度和方向

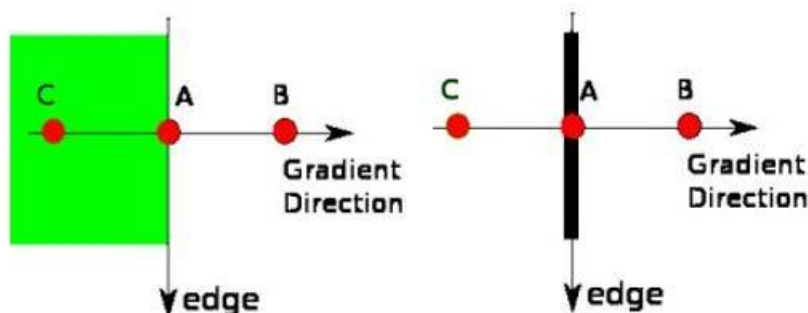
$$\text{Edge_Gradient } (G) = \sqrt{G_x^2 + G_y^2}$$
$$\text{Angle } (\theta) = \tan^{-1} \left(\frac{G_x}{G_y} \right)$$

梯度的方向一般总是与边界垂直。梯度方向被归为四类：垂直，水平，和两条对角线。

非极大值抑制

获得梯度的方向和大小后，对整幅图像进行扫描，去除非边界上的点。

对每个像素进行检查，观察这个点的梯度是否为周围具有相同梯度方向的点中最大的（数值）。



然后得到一个包含“窄边界”的二值图像。

滞后阈值

确定选定的边界为真实的边界，需要设置两个阈值： $minVal$ 和 $maxVal$

当图像的灰度梯度高于 $maxVal$ 时被认为是真的边界，低于 $minVal$ 认为不是真的边界。

如果介于两者之间的话，就要看这个点是否与某个被确定为真正边界点相连，如果是，就认为它也是边界点，如果不是就抛弃。

2. OpenCV中的Canny边界检测

使用函数：`cv2.Canny()`

第一个参数是输入图像，第二个为 $minVal$ 和 $maxVal$ ，第三个参数用来设置Sobel卷积核的大小，默认大小为3，第四个参数为`L2gradient`，可以用来设定求梯度大小的方程，若为`True`，就使用上述方程，否则使用：

$$\text{Edge_Gradient}(G) = |G_x^2| + |G_y^2|$$

即：L2范数

默认值为`False`。

代码实现：

```
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('1024.jpg',0)
6 edges = cv2.Canny(img,100,200)
7
8 plt.subplot(121),plt.imshow(img,cmap='gray')
9 plt.title('original'),plt.xticks([],plt.yticks([]))
10 plt.subplot(122),plt.imshow(edges,cmap='gray')
11 plt.title('edge'),plt.xticks([],plt.yticks([]))
12
13 plt.show()
```

original



edge

