OpenCV-python 图像梯度

OpenCV提供了三种不同的梯度滤波器（高通滤波器）：

`Sobel`，`Scharr`,`Laplacian`

`Sobel`&`Scharr`:求一阶导数or求二阶导数

`Scharr`是对`Sobel`（使用小卷积核求解梯度角度）的优化

`Laplacian`：求二阶导数

1. Sobel算子与Scharr算子

`Sobel`算子是高斯平滑与微分操作的结合体，抗降噪能力很好。

可以设定求导的方向（`xorder`或`yorder`）。还可以设定使用的卷积核大小(`ksize`)，若`ksize=-1`

，会使用3×3的`Scharr`滤波器，效果会更好，若速度相同，在使用3x3滤波器时尽量使用`Scharr`。

3×3的`Scharr`滤波器卷积核如下：

X方向

-3	**0**	**3**
-10	0	10
-3	0	3

Y方向

-3	**-10**	**-3**
0	0	0
3	10	3

2. Laplacian算子

拉普拉斯算子可以使用二阶导数的形式定义，可假设其离散实现类似于二阶`Sobel`导数，事实上OpenCV在计算拉普拉斯算子时直接调用`Sobel`算子。
拉普拉斯滤波器使用的卷积核：

$$\text{kernel} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

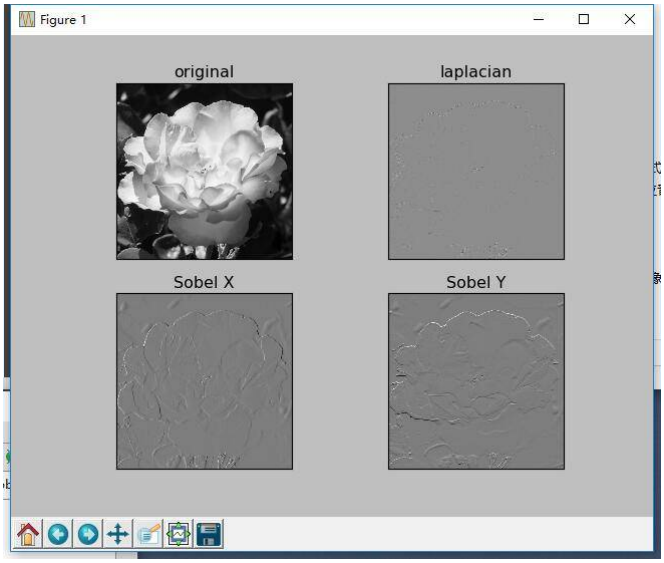
代码实现：

```
import cv2
import numpy
from matplotlib import pyplot as plt

img = cv2.imread('1024.jpg',0)
laplacian = cv2.Laplacian(img,cv2.CV_64F)  # 使用laplacian算子
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)  # 使用sobel算子
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)  # 使用sobel算子

plt.subplot(2,2,1),plt.imshow(img,cmap='gray')
plt.title('original'),plt.xticks([]),plt.yticks([])
plt.subplot(2,2,2),plt.imshow(laplacian,cmap='gray')
plt.title('laplacian'),plt.xticks([]),plt.yticks([])
plt.subplot(2,2,3),plt.imshow(sobelx,cmap='gray')
plt.title('Sobel X'),plt.xticks([]),plt.yticks([])
plt.subplot(2,2,4),plt.imshow(sobely,cmap='gray')
plt.title('Sobel Y'),plt.xticks([]),plt.yticks([])

plt.show()
```



> 注：
>
> 当我们可以通过参数-1来设定输出图像的深度（数据类型）与原图像保持一致，但是我们在代码中使用的却是`cv2.CV_64F`。这是为什么？想象一下一个从黑到白的边界的导数是正数，而一个从白到黑的边界的导数却是负数。如果原图像的深度是`np.int8`时，所有的负值都会被截断变成0。换句话就是把边

> 界丢失掉。
> 所以如果这两种边界你都想检测到，最好的办法就是将输出的数据类型设置的更高，比如`cv2.CV_16S`等，取绝对值然后再把它转回到`cv2.CV_8U`。