

20180730-20180805 周报

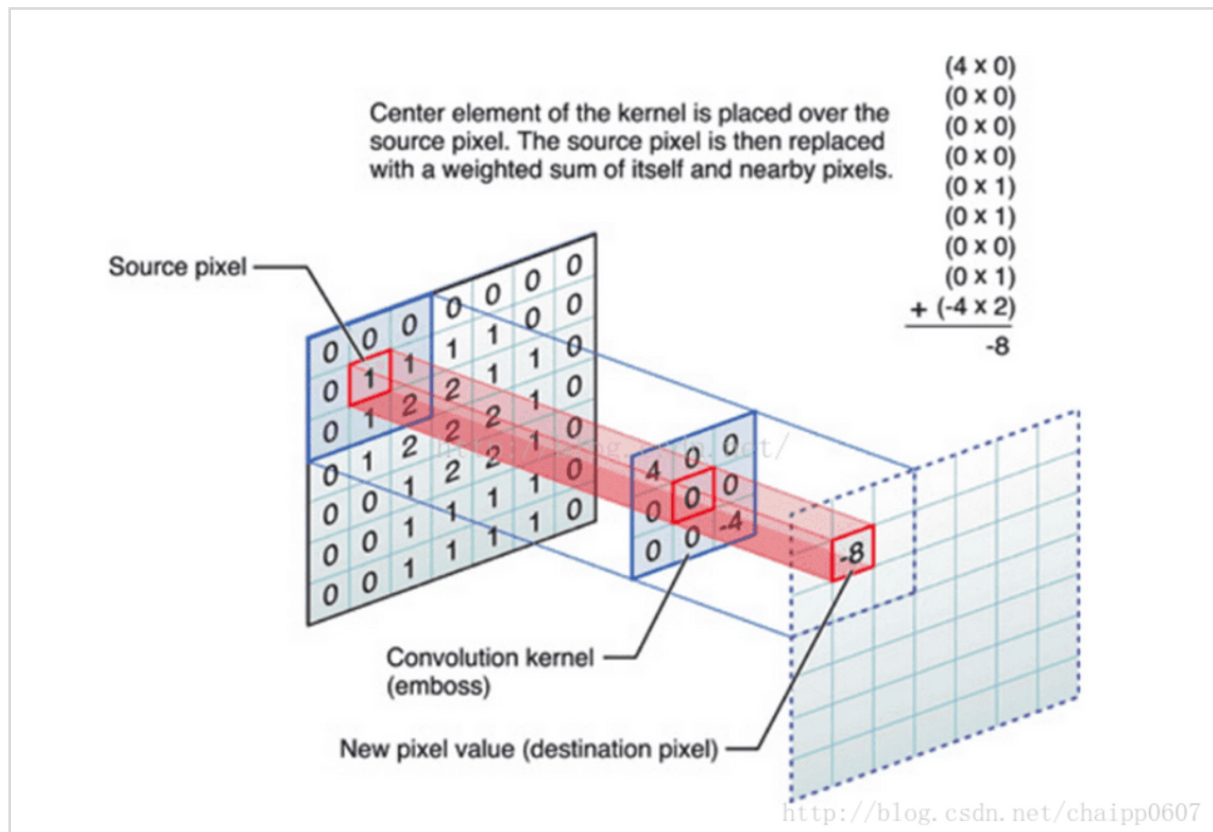
#CV@yang

1. Opencv

因为要参加巡线智能车比赛，继续学习opencv，做点笔记：

Image processing - Morphological Transformations

图像卷积的概念



erosion 腐蚀

```
kernel = np.ones((5,5),np.uint8)
erosion = cv.erode(img,kernel,iterations = 1)
```





dilation 膨胀

```
dilation = cv.dilate(img, kernel, iterations = 1)
```



两种运算一般是分别以形态核(shape kernel)在每个源像素周围的指定区域(核的形状, 以像素为中心)取最小/最大值, 扩展/收缩亮度大的区域。

opening 开运算

```
opening = cv.morphologyEx(img, cv.MORPH_OPEN, kernel)
```



先腐蚀再膨胀, 可以去除独立的亮点。

closing 闭运算

```
closing = cv.morphologyEx(img, cv.MORPH_CLOSE, kernel)
```



-
- 闭运算
- 先膨胀再腐蚀，可以去除独立的暗点。

gradient 梯度

```
gradient = cv.morphologyEx(img, cv.MORPH_GRADIENT, kernel)
```

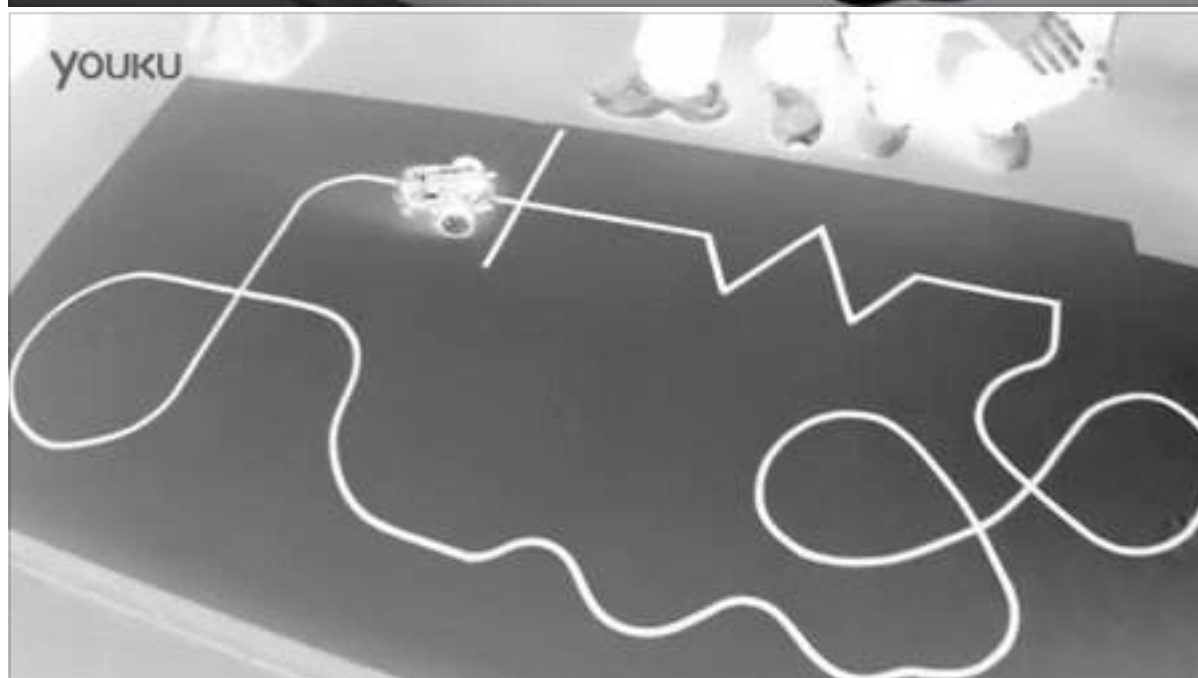
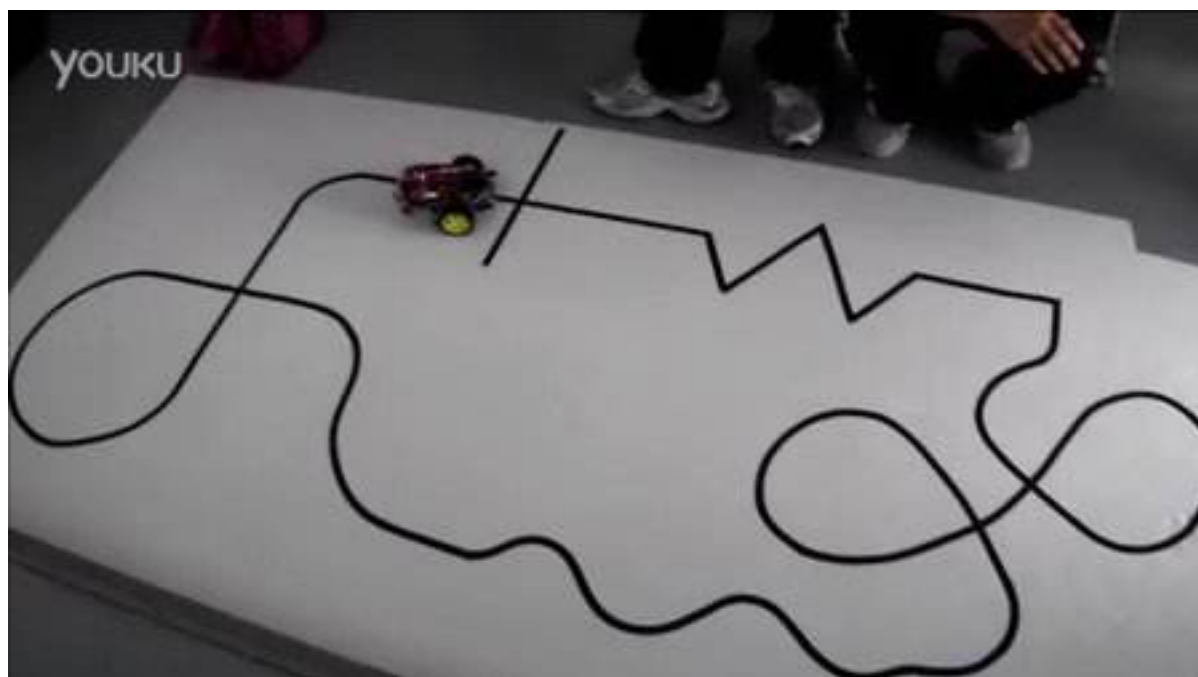
- 原理很简单，就是膨胀与腐蚀的差模运算。



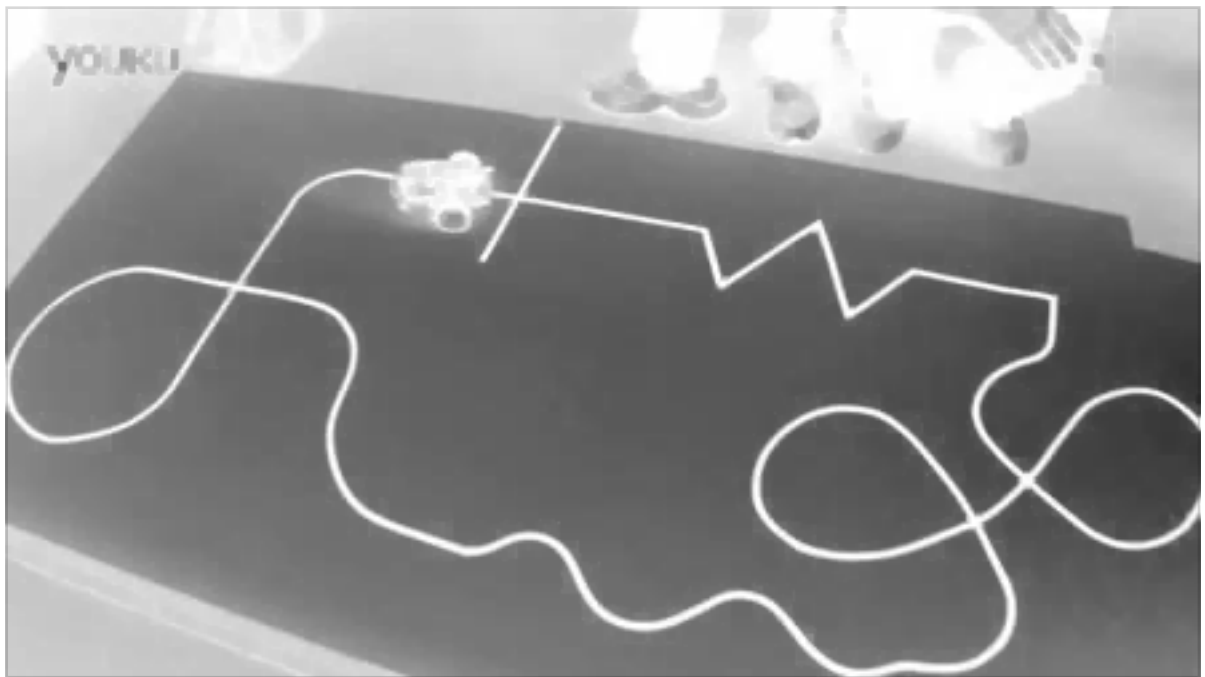
-
- 显然，这种操作可以得到物体的大致轮廓。

Practice

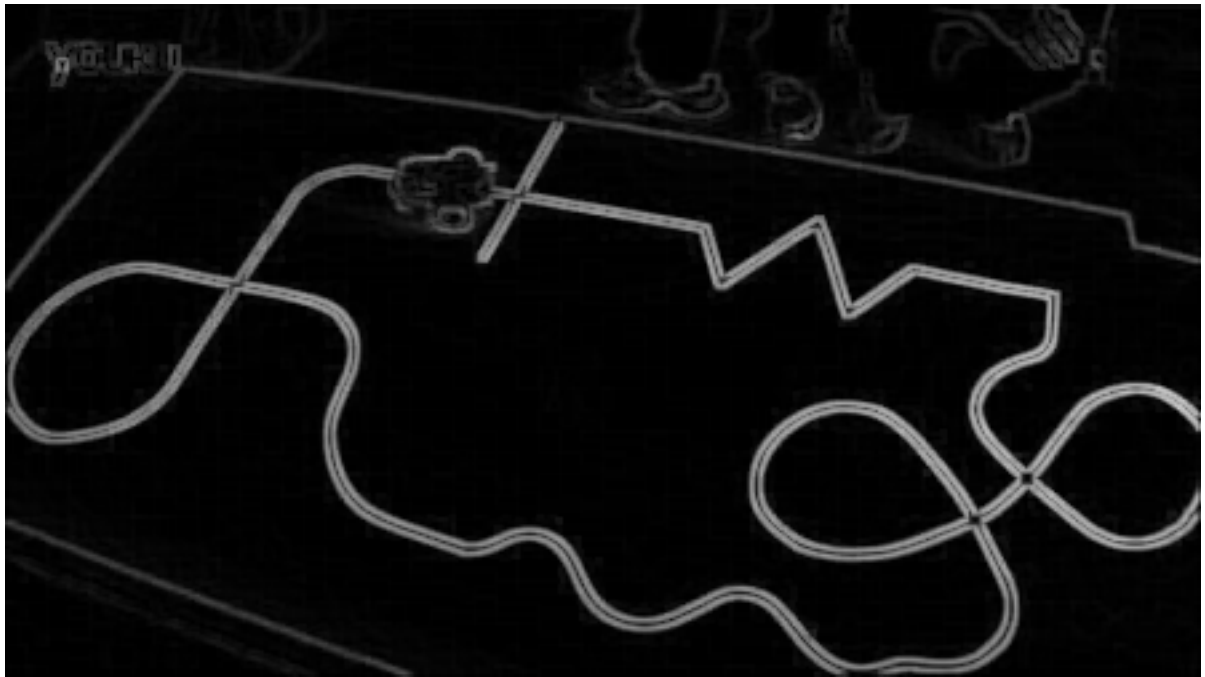
- 从网上搜了一张图，照着代码尝试对巡线小车场地做了如上操作，发现核的大小对比较细的线条的梯度运算结果差异比较大，用3x3方形核比5x5的效果好得多。（为了简便代码运行，预先将图片反相处理）



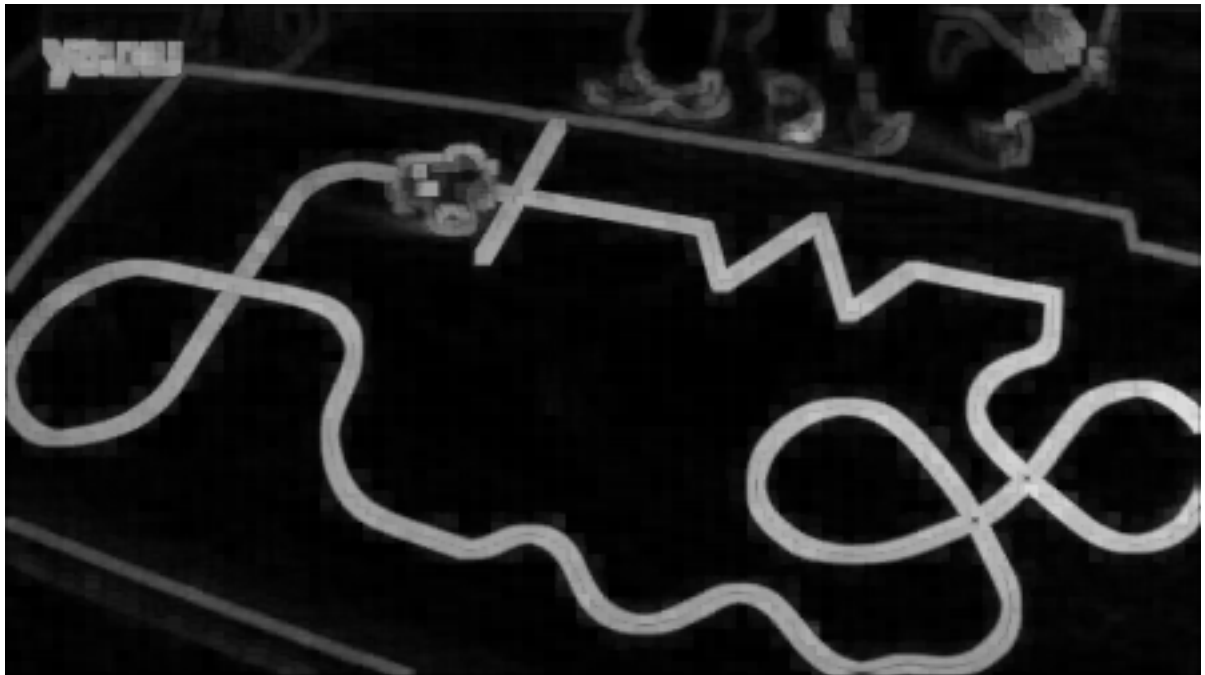
原图的反相图



闭运算 (3x3核)



梯度运算 (3x3)



梯度运算 (5x5)

- 很明显3x3核的效果很好，几乎可以不用后续处理就可以拿来做下一步的算法的输入了。