形态学运算

电子信息工程系

袁羽

目录

CONTENTS

一 形态学运算

腐蚀和膨胀是形态学的基础操作，除了开运算和闭运算以外，形态学中其他高级运算也都是由腐蚀和膨胀组合而来的。

一 形态学运算

OpenCV提供了一个morphologyEx()形态学方法，包含所有常用的运算，其语法如下：

dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)

参数说明：

src：原始图像

op：操作类型

kernel：操作过程中使用的核。

anchor：可选参数，核的锚点位置。

iterations：可选参数，迭代次数，默认值为1。

borderType：可选参数，边界样式，建议默认。

borderValue：可选参数，边界值，建议默认。

返回值说明：dst：操作之后得到的图像。

一 形态学运算

操作类型:

参数值	含义
MORPH_ERODE	腐蚀，跟函数效果一样
MORPH_DILATE	膨胀，跟函数效果一样
MORPH_OPEN	开运算，先腐蚀后膨胀
MORPH_CLOSE	闭运算，先膨胀后腐蚀
MORPH_GRADIENT	剃度运算，原图的膨胀减去原图腐蚀
MORPH_TOPHAT	顶帽运算，原图减去原图的开运算
MORPH_BLACKHAT	黑帽运算，原图的闭运算减去原图

二 腐蚀和膨胀

1.腐蚀

dst = cv2.erode(src, kernel, anchor, iterations, borderType, borderValue)

等价于:

op = cv2.MORPH_ERODE

dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)

2.膨胀

dst = cv2.dilate(src, kernel, anchor, iterations, borderType, borderValue)

等价于:

op = cv2.MORPH_DILATE

dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)

二 腐蚀和膨胀

1.腐蚀

```
import cv2
import numpy as np
img = cv2.imread("cactus.jpg") # 读取原图
k = np.ones((3, 3), np.uint8) # 创建3*3的数组作为核
cv2.imshow("img", img) # 显示原图
# dst = cv2.erode(img, k) # 腐蚀操作
dst = cv2.morphologyEx(img, cv2.MORPH_ERODE, k) # 腐蚀操作
cv2.imshow("dst", dst) # 显示腐蚀效果
cv2.waitKey() # 按下任何键盘按键后
cv2.destroyAllWindows() # 释放所有窗体
```

二 腐蚀和膨胀

2.膨胀

```
import cv2
import numpy as np
img = cv2.imread("cactus.jpg") # 读取原图
k = np.ones((3, 3), np.uint8) # 创建3*3的数组作为核
cv2.imshow("img", img) # 显示原图
# dst = cv2.dilate(img, k) # 膨胀操作
dst = cv2.morphologyEx(img, cv2.MORPH_DILATE, k) # 膨胀操作
cv2.imshow("dst", dst) # 显示腐蚀效果
cv2.waitKey() # 按下任何键盘按键后
cv2.destroyAllWindows() # 释放所有窗体
```

三 开运算和闭运算

1.开运算：

```
dst = cv2.erode(img, k)  # 腐蚀操作
dst = cv2.dilate(dst, k)  # 膨胀操作
```

等价于：

```
op = cv2.MORPH_OPEN
dst = cv2.morphologyEx(img, op, k)
```

2.闭运算

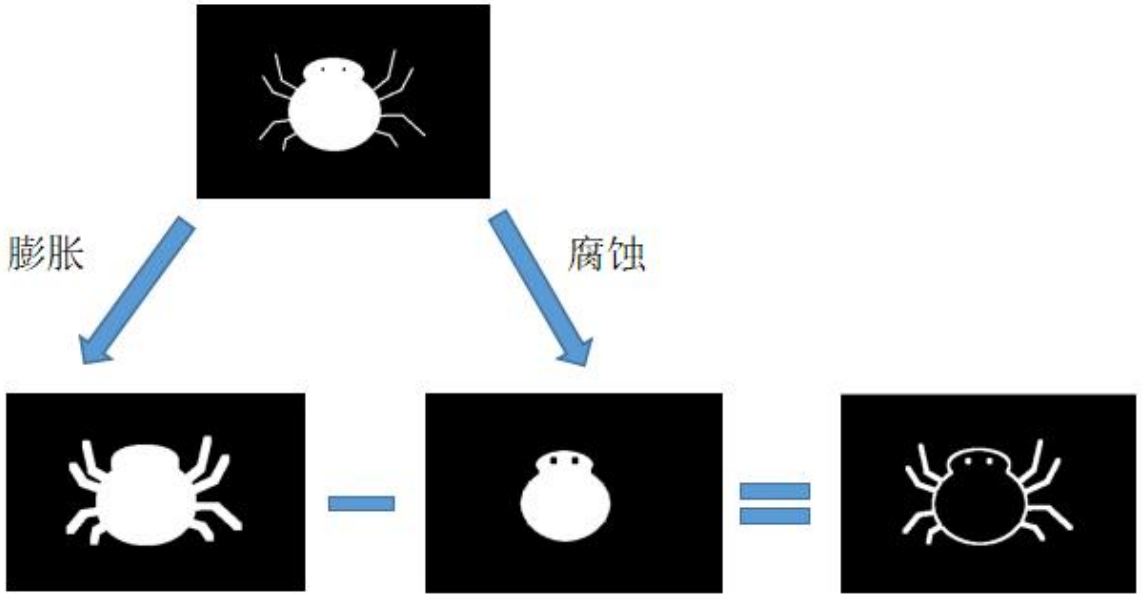
```
dst = cv2.dilate(img, k)  # 膨胀操作
dst = cv2.erode(dst, k)  # 腐蚀操作
```

等价于：

```
op = cv2.MORPH_CLOSE
dst = cv2.morphologyEx(img, op, k)
```

四 梯度运算

梯度运算的运算过程是让原图的膨胀图减原图的腐蚀图。因为膨胀图比原图大，腐蚀图比原图小，利用腐蚀图将膨胀图掏空，就得到了原图的轮廓图（梯度运算中得到的轮廓图只是一个大概轮廓，不精准）。



四 梯度运算

OpenCV中借助morphologyEx()形态学方法可实现梯度运算。

dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)

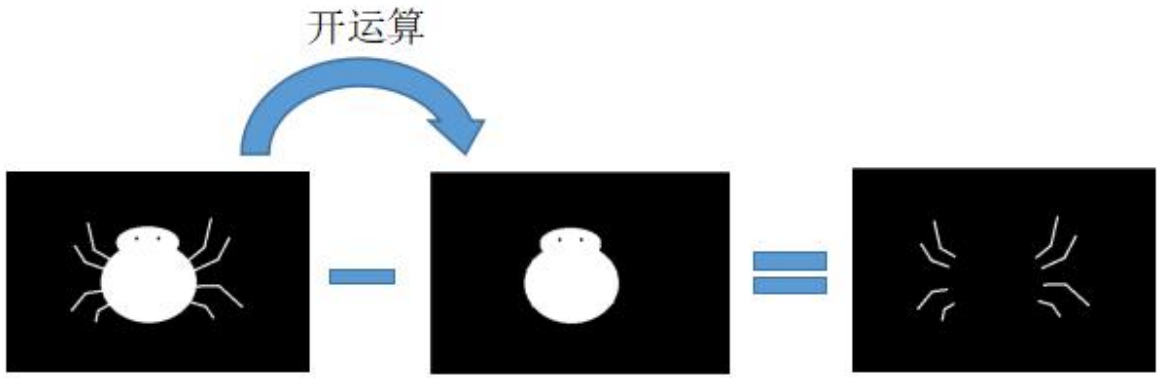
梯度运算的参数为cv2.MORPH_GRADIENT

例：通过梯度运算画出小蜘蛛的轮廓。

```
import cv2
import numpy as np
img = cv2.imread("spider.png") # 读取原图
k = np.ones((5,5), np.uint8) # 创建5*5的数组作为核
cv2.imshow("img", img) # 显示原图
dst = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, k) # 进行梯度运算
cv2.imshow("dst", dst) # 显示梯度运算结果
cv2.waitKey() # 按下任何键盘按键后
cv2.destroyAllWindows() # 释放所有窗体
```

五 顶帽运算

图像顶帽运算是原始图像减去图像开运算的结果，得到图像的噪声。因为开运算抹除图像的外部细节，“有外部细节”的图像减去“无外部细节”的图像，得到的结果就只剩外部细节了。



五 顶帽运算

OpenCV中借助morphologyEx()形态学方法可实现顶帽运算。

dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)

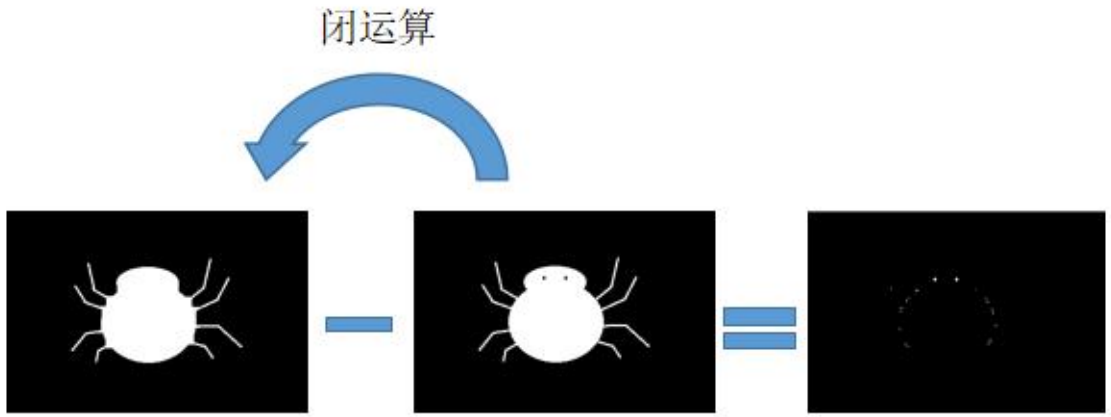
顶帽运算的参数为cv2.MORPH_TOPHAT

例 通过顶帽运算画出小蜘蛛的腿。

```
import cv2
import numpy as np
img = cv2.imread("spider.png") # 读取原图
k = np.ones((5, 5), np.uint8) # 创建5*5的数组作为核
cv2.imshow("img", img) # 显示原图
dst = cv2.morphologyEx(img, cv2.MORPH_TOPHAT, k) # 进行顶帽运算
cv2.imshow("dst", dst) # 显示顶帽运算结果
cv2.waitKey() # 按下任何键盘按键后
cv2.destroyAllWindows() # 释放所有窗体
```

六 黑帽运算

黑帽运算的运算过程是让原图的闭运算图减去原图。因为闭运算抹除图像的内部细节，“无内部细节”的图像减去“有内部细节”的图像，得到的结果就只剩内部细节了。



六 黑帽运算

OpenCV中借助morphologyEx()形态学方法可实现顶帽运算。

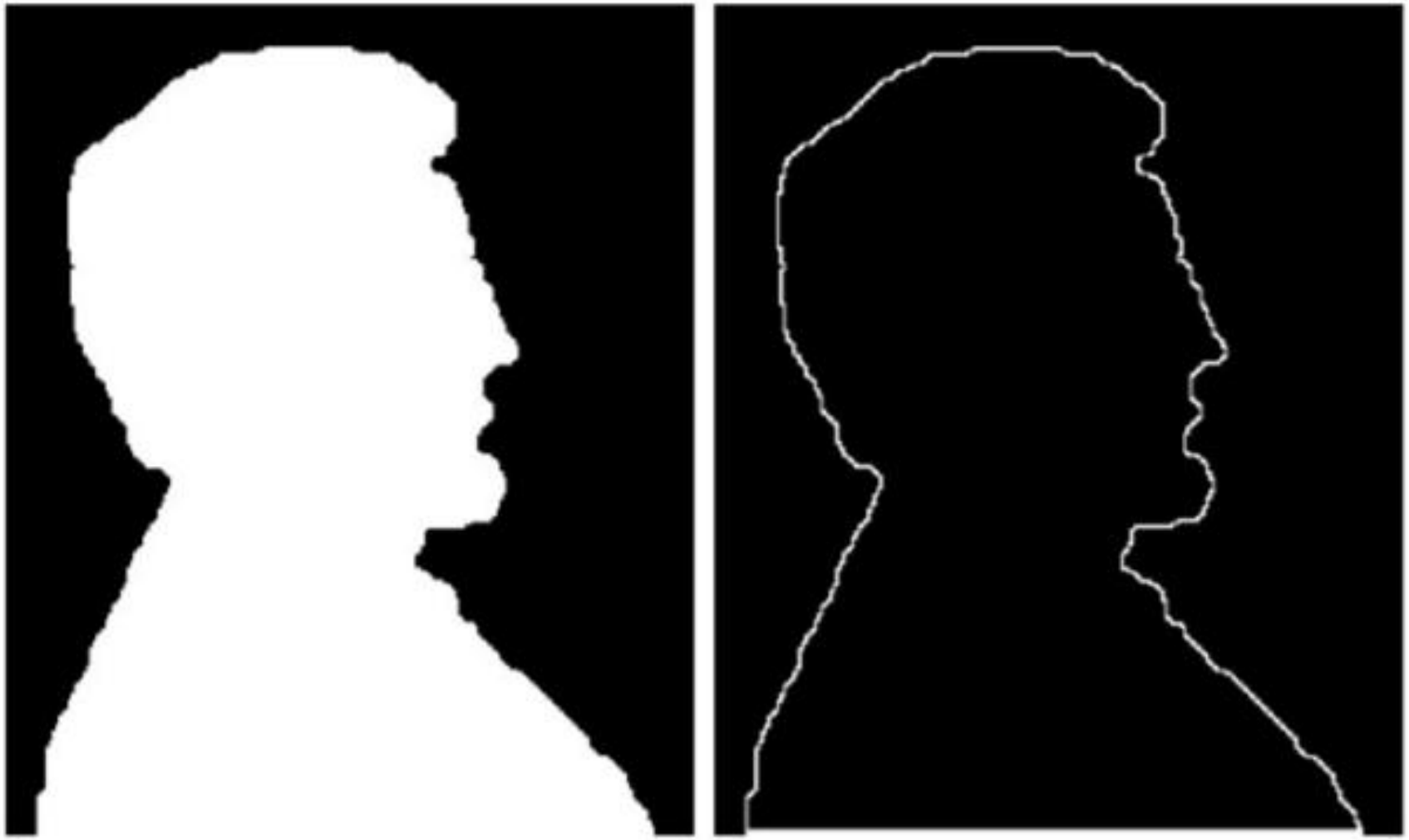
dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)

梯度运算的参数为cv2.MORPH_BLACKHAT

例 通过黑帽运算画出小蜘蛛内部细节。

```
import cv2
import numpy as np
img = cv2.imread("spider.png") # 读取原图
k = np.ones((5, 5), np.uint8) # 创建5*5的数组作为核
cv2.imshow("img", img) # 显示原图
dst = cv2.morphologyEx(img, cv2.MORPH_BLACKHAT, k) # 进行黑帽运算
cv2.imshow("dst", dst) # 显示黑帽运算结果
cv2.waitKey() # 按下任何键盘按键后
cv2.destroyAllWindows() # 释放所有窗体
```

例：获取图像轮廓。



例　对指纹图像进行去噪处理。

。

小结

1.形态学运算

2 腐蚀与膨胀

3.开运算与闭运算

4.梯度运算

5.顶帽运算

6.黑帽运算

THANKS