作業3:Morphology Boundary extraction and Region Filling

班級: 碩專班

學生:5108056016張仲威

教授: 吳俊霖 教授

目錄

[一. 主要演算法 2](#_Toc59979427)

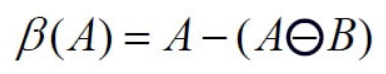
[二. 程式片斷 2](#_Toc59979428)

[三. 測試資料 5](#_Toc59979429)

[四. 結果呈現 5](#_Toc59979430)

[五. 結論 6](#_Toc59979431)

# 主要演算法



第一個邊界抽取 Boundary Extraction概念比較簡單, 用 **A 跟 B 做 Erosion** 會得到一個比較小的A，再用**原來的A 減掉比較小的A**，那麼多出來的地方就剛好是邊界了



第二個Region Filling比較複雜, Xk 為你要填充區域裡面的Element，B為要拿來做**Dilation** 的Structures Element，配合後面**跟A 的補集做 AND 的條件**，然後就會從 k = 1、2、3、... 開始做，做到被 A 包住的整個範圍都做完以後結束

# 程式片斷

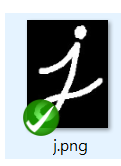
1. Boundary Extraction:
2. **import** cv2
3. **import** numpy as np
5. img = cv2.imread('./img/j.png',0) # 原圖
6. kernel = np.ones((5,5),np.uint8) # Structures Element
7. erosion = cv2.erode(img,kernel,iterations = 1) # 原圖和Structures Element erosion結果
9. boundary = cv2.subtract(img,erosion) # 原圖-erosion結果

2. Region Filling:

1. **import** numpy as np
2. **import** cv2 as cv
3. **from** matplotlib **import** pyplot as plt
4. %matplotlib inline
5. img = cv.imread("./img/regionfilling.jpg")
7. # 二值化
8. imgray = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)
9. imgray[imgray < 100] = 0
10. imgray[imgray >= 100] = 255
12. # 原圖取補
13. mask = 255 - imgray
15. # 定義Marker圖
16. marker = np.zeros\_like(imgray)
17. marker[0, :] = 255
18. marker[-1, :] = 255
19. marker[:, 0] = 255
20. marker[:, -1] = 255
21. marker\_0 = marker.copy()
23. # 形態學重建
24. SE = cv.getStructuringElement(shape=cv.MORPH\_CROSS, ksize=(3, 3))
25. # 無窮迴圈,做到XK不在改變
26. **while** True:
27. marker\_pre = marker
28. dilation = cv.dilate(marker, kernel=SE)
29. marker = np.min((dilation, mask), axis=0)
30. **if** (marker\_pre == marker).all():
31. **break**
32. dst = 255 - marker
33. filling = dst - imgray
35. # 畫圖
36. plt.figure(figsize=(12, 6))  # width \* height
37. plt.subplot(2, 3, 1), plt.imshow(imgray, cmap='gray'), plt.title('Original image'), plt.axis("off")
38. plt.subplot(2, 3, 2), plt.imshow(mask, cmap='gray'), plt.title('Original image ^'), plt.axis("off")
39. plt.subplot(2, 3, 3), plt.imshow(marker\_0, cmap='gray'), plt.title('Marker 0'), plt.axis("off")
40. plt.subplot(2, 3, 4), plt.imshow(marker, cmap='gray'), plt.title('Marker'), plt.axis("off")
41. plt.subplot(2, 3, 5), plt.imshow(dst, cmap='gray'), plt.title('dst'), plt.axis("off")
42. plt.subplot(2, 3, 6), plt.imshow(filling, cmap='gray'), plt.title('Region Filling result'), plt.axis("off")
43. plt.show()

# 測試資料

1.Boundary Extraction:



2. Region Filling:

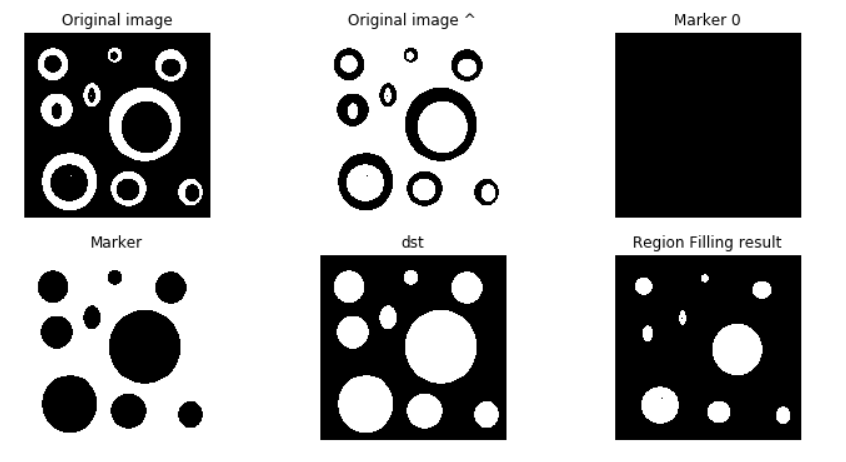


# 結果呈現

1.Boundary Extraction:



1. Region Filling



# 結論

此作業讓我了解簡單的影像形態學操作Dilation或Erosion,能應用於邊界抽取和區域填充,覺得這樣的作法也是可以讓結果做得很好