

# 電腦視覺 HW7 report

D10922012 資工所 王傳啟

Write a program which does thinning on a downsampled image (lena.bmp).

Step1: 二值化( HW6, 略)

Step2: 8-block, 並以最左上進行採樣 (HW6, 略)

Step3: thinning operator, 按照作業網頁的 pdf 建構流程, 迴圈反覆執行直到不相同中止。

```
# step3: Thinning Operator as hw requirement
iteration = 1
isChanged = True
while isChanged:
    print("iteration: ", iteration)
    np_ori = np.copy(np_thin)
    isChanged = False
    np_yokoi = mark_yokoi(np_thin) # step 3.1
    np_pair = mark_pair(np_yokoi) # step 3.2
    np_thin = mark_shrink(np_pair) # step 3.3
    img_tmp = Image.fromarray(np.uint8(np_thin))
    img_tmp.save('result/iteration{}.bmp'.format(iteration))
    # check the output is changed from the origin
    for i in range(thin_row):
        for j in range(thin_col):
            if np_ori[i][j] != np_thin[i][j]:
                isChanged = True
                break
    iteration += 1
```

Step3.1: 以 4 連通的方式, 構築 Yokoi connectivity number, 此處幾乎與 HW6 相同, 唯一不同之處在於 背景 與 孤立點需要被明確區別, 此處賦予背景的 Yokoi number 為 7 (以 3-bit 結束的 7 為代表), 其他部分相同故省略。

Step3.2: 按照 pdf 給予的公式, 撰寫對應程式碼即可, 除背景都需做對應判斷。

```
def mark_pair(img, m = 1):
    # [input]: m = 1 (Yokoi edge)
    row = img.shape[0]
    col = img.shape[1]
    pair_list = list()
    h = (lambda a, m: 1 if a==m else 0)
```

```

for i in range(row):
    for j in range(col):
        x0 = img[i][j]
        if x0 != 7: # if current point is not background
            x1 = 0 if j == col - 1 else img[i][j + 1]
            x2 = 0 if i == 0 else img[i - 1][j]
            x3 = 0 if j == 0 else img[i][j - 1]
            x4 = 0 if i == row - 1 else img[i + 1][j]
            sum_ = h(x1, m) + h(x2, m) + h(x3, m) + h(x4, m)
            pair_list.append('p' if (sum_ >= 1 and x0 == m) else 'q')
        else:
            pair_list.append('g')
np_pair = np.array(pair_list).reshape((row, col))
return np_pair

```

Step3.3: 收縮因子的部分，則對 yokoi 的 h 函數修改即可。簡單來說，即為 b 為 p 且 c 有值時，d 與 e 任一點必須為背景。

```

def mark_shrink(np_pair):
    row, col = np_pair.shape[0], col = np_pair.shape[1]
    output = np.zeros((row, col), np.int)
    h = (lambda _, c, d, e: 1 if ((c!='g') and ((d=='g') or (e=='g'))))
else 0)
    f = (lambda a1, a2, a3, a4, x: 'g' if (a1+a2+a3+a4) == 1 else x)
    for i in range(row):
        for j in range(col):
            x0 = np_pair[i][j]
            if x0 == 'p':
                x1 = 'g' if j == col - 1 else np_pair[i][j + 1]
                ...
                a1 = h(x0, x1, x6, x2)
                a2 = h(x0, x2, x7, x3)
                a3 = h(x0, x3, x8, x4)
                a4 = h(x0, x4, x5, x1)
                np_pair[i][j] = f(a1, a2, a3, a4, x0)
    for i in range(row):
        for j in range(col):
            if np_pair[i][j] != 'g':
                output[i][j] = 255
    return output

```

輸出: 與題目要求一樣, 共作 7 次迭代



iteration1.bmp



iteration2.bmp



iteration3.bmp



iteration4.bmp



iteration5.bmp



iteration6.bmp



iteration7.bmp

最後輸出(詳見附檔):

