

Answer**thinning**

本次作業使用 python, IDE 為 Spyder

Description

1. Method of Algorithms of thinning operator

I. Yokoi connectivity operator

It is same as HW6. Thus, I will not describe anymore.

II. Pair Relationship operator

From the previous Yokoi array.

Let state= (Yokoi number = '1' \wedge at least one of Yokoi number neighbor = '1')

If the state condition established, output 1, otherwise=2

III. Connected Shrink operator

for 4-connectivity

$$h(b, c, d, e) = \begin{cases} 1 & \text{if } b = c \wedge (b \neq d \vee b \neq e) \\ 0 & \text{otherwise} \end{cases}$$

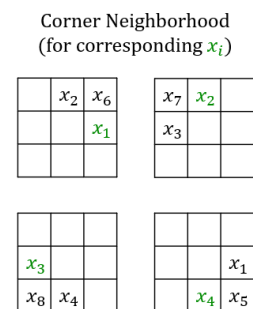
$$a_1 = h(x_0, x_1, x_6, x_2)$$

$$a_2 = h(x_0, x_2, x_7, x_3)$$

$$a_3 = h(x_0, x_3, x_8, x_4)$$

$$a_4 = h(x_0, x_4, x_5, x_1)$$

$$output = f(a_1, a_2, a_3, a_4, x_0) = \begin{cases} g & \text{if exactly one of } a_1, a_2, a_3, a_4 = 1 \\ x_0 & \text{otherwise} \end{cases}$$



Input : $A+B$

Let A=operating of $f(h())$ in every pixel of (original symbolic image)

Let B=output of pair relationship

If the pixel in $A = 1 \wedge B = 1$, then output pixel = 0, which means delete

Repeat step1, step2, step3 until the last output never changed

2. Source Code Description

主程式:

先做 binary , 再做 downsample , 再做 thinning(Yokoi+pairRelationship +connectedShrink)

```

173 if __name__ == '__main__':
174     from PIL import Image
175     import numpy as np
176
177     lena = Image.open("lena.bmp")
178     binary_lena = lena.point(lambda x: 0 if x < 128 else 255, '1')
179     downsampling_image = downsampling(binary_lena, 8)
180     downsampling_image.save('downsampling.bmp')
181     thinning_image = downsampling_image.copy()
182     check = True
183     iteration = 1
184     while check:
185         iteration += 1
186         yokoi_matrix = Yokoi(thinning_image)
187         paired_matrix = pairRelationship(yokoi_matrix)
188         thinning_image, check = connectedShrink(thinning_image, paired_matrix)
189         thinning_image.save('thinning.bmp')

```

Yokoi:上次作業做過，不再重新說明

pairRelationship:

如果yokoi matrix 的pixel 不等於' 1' , output2 , 如果yokoi =' 1' 則4-connected也必須等於1

才會output=1 , 否則一律標為2

```

96 def pairRelationship(matrix):
97     ## 10 11
98     type I0 numpy array
99     ## 10 11
100     # 1: p, 2: q
101     r, c = matrix.shape
102     paired_matrix = np.zeros(matrix.shape, dtype=int)
103     for i in range(r):
104         for j in range(c):
105             if matrix[i][j] != '1': # Yokoi number != 1
106                 paired_matrix[i][j] = 2 # Set to q
107             else: # Yokoi number == 1
108                 flag = True
109                 neighbor4 = [(1, 0), (0, -1), (-1, 0), (0, 1)]
110                 for m, n in neighbor4:
111                     if 0 <= i+m < r and 0 <= j+n < c:
112                         if matrix[i+m][j+n] == '1': # Exist a neighbor' Yokoi number = 1
113                             paired_matrix[i][j] = 1 # Set to p
114                             flag = False
115                             break
116                 if flag:
117                     paired_matrix[i][j] = 2
118
119     return paired_matrix

```

connectedShrink:

如果yokoi 是1，並且經過計算的f(h(原圖))也等於1則將output=0，並flag=1，繼續重複流程

```

122 def connectedShrink_h(b, c, d, e):
123     """
124     type of b,c,d,e: int return: int
125     """
126     if b == c and (b != d or b != e):
127         return 1
128     else:
129         return 0
130
131
132 def connectedShrink_f(a1, a2, a3, a4):
133     """
134     type of a1, a2, a3, a4: int return: int
135     """
136     return [a1, a2, a3, a4].count(1) == 1
137
138
139 def connectedShrink(originalImage, matrix):
140     """
141     type img: Image,matrix :numpy array return: Image ,Int
142     """
143     ImageArray = np.array(originalImage)
144     r, c = ImageArray.shape
145     flag = False
146     for i in range(r):
147         for j in range(c):
148             if ImageArray[i][j] == 1:
149                 x = [0 for i in range(9)]
150                 x[0] = ImageArray[i][j]
151                 index = 0
152                 neighbor8 = [(1, 0), (0, -1), (-1, 0), (0, 1),
153                             (1, 1), (1, -1), (-1, -1), (-1, 1)]
154                 for m, n in neighbor8:
155                     index += 1
156                     if 0 <= i+m < r and 0 <= j+n < c:
157                         x[index] = ImageArray[i+m][j+n]
158
159                 a1 = connectedShrink_h(x[0], x[1], x[6], x[2])
160                 a2 = connectedShrink_h(x[0], x[2], x[7], x[3])
161                 a3 = connectedShrink_h(x[0], x[3], x[8], x[4])
162                 a4 = connectedShrink_h(x[0], x[4], x[5], x[1])
163
164                 number = connectedShrink_f(a1, a2, a3, a4)
165                 # Yokoi number = 1 (edge) and pair relationship =1
166                 if number == 1 and matrix[i][j] == 1:
167                     ImageArray[i][j] = 0
168                     flag = True
169     img = Image.fromarray(ImageArray)
170     return img, flag

```