

Computer Vision Homework #6

Student-ID: r10944020; Name: 林顯倫; Department: GINM

[Results]

11111111	12111111111122322221	111111111111	0 0
15555551	11555555511 2 11 11	115555555511	0
15555551	1 2115555112 21112221	15555555551	21
15555551	1 2 155112 22221511	155555555511	1
15555551	22 2112 22 121 0 0	1555555555511	0
15555551	1 2 21 2 1 1	1555555555551	0
15555551	12 1 121111 1321	15555555555511	
15111551	1322 1155551111	15555555555551	
111 1551	1 12155555511	15555555555511	
11 1551	21155555511	1551115555511	
21 1551	2 15555555111	1551 1155511	
1 1551	2 155555555511	1551 115551	1
1551	112115555555551	1551 15511	12
1551	1555555555555511	1551 1111	111
1551	1 22211555555555511	1151 11	1151
1551	2 22 1 155555555555511	151 11111	1551
1551	2 1 1155555555555551	151 115551	11551
1551	2 115555555555555111511155511		115551
1551	12 11555555555555555555555551		155551
1551	11 0 2215555555555555555555555112		115551
1551	111 22 155555555555555555555551 1		155551
1551	1511 1 12511211111211155555555111		1155551
1551	15521 1 121 1 11 1 15555555111 0		1555551
1551	1151 132 2 1155555111 0		11555551
1551	151 0 322 115555111 121		15555551
1551	1221 2 155551 131		11555551
1551	2 0 1 11555511 1		115555551
1551	2 0 0 115555551 0		1 15555551
1551	2 1155555551		2115555551
1551	1 0 11555555551		1555555551
1551	1 11511115555521 1		11555555551
1551	1 1 1111 1155511 2		15555555551
1551	131 11 15111 2		15555555551
1551	121 0 1121 1 111 1 2		115555555551
1551	11 111 1 221 11 1 2		155555555551
1551	12 0 1 21 121 11 1111 2		155555555551
1551	1 12 22 151111111551 2		1155555555551
1551	1 2 1555551115511 1		1555555555551
1551	2 0 0 22 12555551 15551 1		1555555555551
1551	1 1 1555511 11511 2		11555555555551
1551	0 0 21 155551 1 151 2		15555555555551
1551	2 15555112 151 2		15555555555551
1551	1 1 1 1155555511111 2		155555555555551
1551	2 22 11151111212 2115555555555551		2115555555555551
1551	0 1 12 151 2 1 1555555111555551		1555555511155551
1551	0 0 0 1111 121 155555551 155551		155555551 155551
1551	0 11111111 155555551 155551		155555551 155551
1551	0 115551 155555551 1555511		211111111 155511
1551	15551 211111111 155511		2 11 115511
11521	1 12 122155511 2		11 15511
1 151 0	1 1 155555111 2111		15511
22 1511	1 15555555111 155111		1511
22 1511	1 15555555551 155551		1151
2 151	0 1 11155555555511 155511		1511
2 1521	0 1 155555555555511 15551		12151
2 151	121 15555555555551 155511		1551
2 1511	0 155555555555551 115551		1511
21 1511	11 155555555555551 111111151		
11 151	0 11555555555555511 111511		
11 151	15555555555555551 151		
11 151	0 11555555555555551 211		
11 151	115555555555555511 1		
11 151	0 155555555555555551		
11 111	0 1211111111111111111		

[Code Fragment & Explanation]

p.s. Please refer to the complete code in assignment folder. In this section, I just put the important functions I used.

Step1) Downsample original source image and binarize it.

```
4 def downsample(img, sampleSize=(64,64)):
5     downsample_img = Image.new('L', sampleSize)
6     downsample_img_pixel = downsample_img.load()
7     for x in range(sampleSize[0]):
8         for y in range(sampleSize[1]):
9             downsample_img_pixel[x, y] = img.getpixel((x*8, y*8))
10    downsample_img.save('./downsampled_lena.bmp')
11    return downsample_img
12
13 def binarize(img):
14     imageW, imageH = img.width, img.height
15     new_img = img.copy()
16     new_img_pixel = new_img.load()
17     for x in range(imageW):
18         for y in range(imageH):
19             new_img_pixel[x, y] = 255 if img.getpixel((x,y)) > 127 else 0
20     new_img.save('./binarize_lena.bmp')
21     return new_img
```

Step2) Compute Yokoi Connectivity Number for each pixel whose value is not zero. I just follow the notation in lecture slides to design the functions.

```
23 def pixel_val(x, y):
24     if (x >= 0 and x < 64 and y >= 0 and y < 64): return img.getpixel((x,y))
25     return 0
26
27 def neighborhood(img, x, y):
28     return [
29         pixel_val(x,y)      , # x0
30         pixel_val(x+1,y)    , # x1
31         pixel_val(x,y-1)    , # x2
32         pixel_val(x-1,y)    , # x3
33         pixel_val(x,y+1)    , # x4
34         pixel_val(x+1,y+1)  , # x5
35         pixel_val(x+1,y-1)  , # x6
36         pixel_val(x-1,y-1)  , # x7
37         pixel_val(x-1,y+1)  , # x8
38     ]
39
40 def hFunction(b, c, d, e):
41     if b == c and ( b != d or b != e ): return 'q'
42     elif b == c and ( b == d or b == e ): return 'r'
43     elif b != c: return 's'
44     return ' '
45
46 def fFunction(a1, a2, a3, a4):
47     if a1 == 'r' and a2 == 'r' and a3 == 'r' and a4 == 'r': return 5
48     numberOfQ, records = 0, [a1, a2, a3, a4]
49     for r in records: numberOfQ = numberOfQ + (1 if r == 'q' else 0)
50     return numberOfQ
51
52 def YokoiConnectivityNumber(x):
53     return fFunction(
54         hFunction(x[0], x[1], x[6], x[2]),
55         hFunction(x[0], x[2], x[7], x[3]),
56         hFunction(x[0], x[3], x[8], x[4]),
57         hFunction(x[0], x[4], x[5], x[1])
58     )
59
60 def writeYokoiResult(yokoi_res, sampleSize=(64,64)):
61     f = open("yokoi.txt", "w+")
62     for x in range(sampleSize[0]):
63         for y in range(sampleSize[1]):
64             f.write(str(yokoi_res[y][x]))
65             f.write('\n')
66     f.close()
```