B06502149 資工三 張琦琛

Description:

1. Binarize lena.bmp with threshold 128
2. Down sampling lena.bmp from 512x512 to 64x64, and using 8x8 blocks as a unit, take the topmost-left pixel as the downsampled data.
3. Count the Yokoi connectivity number on a downsampled lena using 4-connected.

Algorithm:

Yokoi h function for 4-connectivity

$$h(b,c,d,e) = \begin{cases} q & if\ b = c\ and\ (d \neq b\ \vee e \neq b) \\ r & if\ b = c\ and\ (d = b \wedge e = b) \\ s & if\ b \neq c\ and\ (d = b \wedge e = b) \end{cases}$$

Yokoi f function for 4-connectivity

$$f(a_1,a_2,a_3,a_4) = \begin{cases} 5, if\ a_1 = a_2 = a_3 = a_4 = r \\ n,\ where\ \ n = \ \ numberof\{\#a_k | a_k = q\},\ otherwise \end{cases}$$

Code fragment:

```
def Yokoi(img):
    r = img.shape[0]
    c = img.shape[1]
    dr = [[0, -1, -1], [-1, -1, 0], [0, 1, 1], [1, 1, 0]]
    dc = [[1, 1, 0], [0, -1, -1], [-1, -1, 0], [0, 1, 1]]
    with open('Yokoi.txt', 'w') as f:
        for i in range(r):
            for j in range(c):
                if (img[i][j] == 255):
                    R, q = 0, 0
                    for k in range(4):
                        cnt = 0
                        for l in range(3):
                            pr, pc = i+dr[k][l], j+dc[k][l]
                            if l == 0:
                                if ((pr < 0) or (pr >= r) or (pc < 0) or (pc >= c) or (img[pr][pc] != 255)):
                                    break
                                else:
                                    cnt = 1
                            else:
                                if ((pr >= 0 and pr < r and pc >= 0 and pc < c) and (img[pr][pc] == 255)):
                                    cnt += 1
                        if cnt == 3:
                            R += 1
                        elif 0 < cnt and cnt < 3:
                            q += 1
                    if R == 4:
                        f.write('5')
                    else:
                        f.write(str(q))
                else:
                    f.write(' ')
            f.write('\n')
```

```python
def downSample(img):
    new_img = np.zeros((64,64), dtype=int)
    r = img.shape[0]
    c = img.shape[1]
    for i in range(0, r, 8):
        for j in range(0, c, 8):
            new_img[i//8][j//8] = img[i][j]
    return new_img
```

Result: