

Answer



Lena.bmp

Laplacian mask 1
Threshold =20Laplacian2
Thresholds:20Minimum variance Laplacian
Thresholds:15Laplacian of Gaussian
Thresholds:3000Difference of Gaussian
Thresholds:1

Algorithm

Each kernel of function

(a) Laplacian mask 1, Threshold =20

```
mask = [0, 1, 0, 1, -4, 1, 0, 1, 0]
```

(b) Laplacian2, Thresholds:20

```
mask = [1, 1, 1, 1, -8, 1, 1, 1, 1]
```

(c) Minimum variance Laplacian, Thresholds:15

```
mask = [2, -1, 2, -1, -4, -1, 2, -1, 2]
```

(d) Laplacian of Gaussian, Thresholds:3000

```
mask = [0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0,
        0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0,
        0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0,
        -1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1,
        -1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1,
        -2, -9, -23, -1, 103, 178, 103, -1, -23, -9, -2,
        -1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1,
        -1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1,
        0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0,
        0, 0, -2, -4, -8, -9, -8, -4, -2, 0, 0,
        0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0]
```

(e) Difference of Gaussian Thresholds:1

```
mask = [-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1,
        -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3,
        -4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4,
        -6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6,
        -7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7,
        -8, -13, -17, 15, 160, 283, 160, 15, -17, -13, -8,
        -7, -13, -17, 0, 85, 160, 85, 0, -17, -13, -7,
        -6, -11, -16, -16, 0, 15, 0, -16, -16, -11, -6,
        -4, -8, -12, -16, -17, -17, -17, -16, -12, -8, -4,
        -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3,
        -1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1]
```

Code description

本次作業使用 python3.7 · IDE 使用 Spyder

(a) Main function:

```
152 if __name__ == "__main__":
153     lena = cv2.imread('lena.bmp', cv2.IMREAD_GRAYSCALE)
154     ...
155     laplacian1 = laplacian_mask1(lena, 20)
156     laplacian2 = laplacian_mask2(lena, 20)
157     minimum_variance = minimum_variance(lena, 15)
158     laplacian_gaussian = laplacian_gaussian(lena, 3000)
159     difference_gaussian = difference_gaussian(lena, 1)
160
161     cv2.imwrite('Laplacian1.bmp', laplacian1)
162     cv2.imwrite('Laplacian2.bmp', laplacian2)
163     cv2.imwrite('Minimum Variance Laplacian.bmp', minimum_variance)
164     cv2.imwrite('Laplacian of Gaussian.bmp', laplacian_gaussian)
165     cv2.imwrite('Difference of Gaussian.bmp', difference_gaussian)
166     ...
```

沒什麼特別，輸入格別 function，再寫入 bmp 檔

(b) get_label:

```
10 def get_label(img, size, mask, threshold, coef):
11     r, c = img.shape
12     res = np.zeros(img.shape, dtype=np.uint8)
13     middle = size // 2
14
15     for i in range(r):
16         for j in range(c):
17             candidate = []
18             for m in range(-middle, middle+1):
19                 for n in range(-middle, middle+1):
20                     if 0 <= i+m < r and 0 <= j+n < c:
21                         candidate.append(img[i+m][j+n])
22             else:
23                 candidate.append(0)
24             value = 0
25             for k in range(len(candidate)):
26                 value += int(candidate[k]) * mask[k]
27             value *= coef
28
29             if value < threshold:
30                 res[i][j] = 0
31             elif value > threshold:
32                 res[i][j] = 1
33             else:
34                 res[i][j] = 2
35
36     return res
```

主要為判斷 threshold 以便往後做判定是否符合 Zero Crossing Edge 條件

```

38 def laplacian_mask1(img, threshold):
39     r, c = img.shape
40     res = np.zeros(img.shape, dtype=np.uint8)
41     mask = [0, 1, 0, 1, -4, -1, 0, 1, 0]
42     label = get_label(img, 3, mask, threshold, 1)
43     for i in range(r):
44         for j in range(c):
45             if label[i][j] == 1:
46                 for m in range(-1, 2):
47                     for n in range(-1, 2):
48                         if 0 <= i+m < r and 0 <= j+n < c:
49                             if label[i+m][j+n] == 0:
50                                 res[i][j] = 0
51             else:
52                 res[i][j] = 255
53     return res
54
55 def laplacian_mask2(img, threshold):
56     r, c = img.shape
57     res = np.zeros(img.shape, dtype=np.uint8)
58     mask = [1, 1, 1, 1, -8, 1, 1, 1, 1]
59     label = get_label(img, 3, mask, threshold, 1/3)
60     for i in range(r):
61         for j in range(c):
62             if label[i][j] == 1:
63                 for m in range(-1, 2):
64                     for n in range(-1, 2):
65                         if 0 <= i+m < r and 0 <= j+n < c:
66                             if label[i+m][j+n] == 0:
67                                 res[i][j] = 0
68             else:
69                 res[i][j] = 255
70     return res

```

```

118 def difference_gaussian(img, threshold):
119     r, c = img.shape
120     res = np.zeros(img.shape, dtype=np.uint8)
121     mask = [-1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1,
122            -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3,
123            -4, -8, -12, -16, -17, -17, -16, -12, -8, -4,
124            -6, -11, -16, -16, 0, 15, 0, 16, 16, 11, 6,
125            -7, -13, -17, 0, 85, 160, 85, 0, 17, 13, 7,
126            -8, -13, -17, 15, 160, 283, 160, 15, 17, 13, 8,
127            -7, -13, -17, 0, 85, 160, 85, 0, 17, 13, 7,
128            -6, -11, -16, -16, 0, 15, 0, 16, 16, 11, 6,
129            -4, -8, -12, -16, -17, -17, -16, -12, -8, -4,
130            -3, -5, -8, -11, -13, -13, -13, -11, -8, -5, -3,
131            -1, -3, -4, -6, -7, -8, -7, -6, -4, -3, -1]
132
133     label = get_label(img, 11, mask, threshold, 1)
134     for i in range(r):
135         for j in range(c):
136
137             if label[i][j] == 1:
138                 for m in range(-5, 6):
139                     for n in range(-5, 6):
140                         if 0 <= i+m < r and 0 <= j+n < c:
141                             if label[i+m][j+n] == 0:
142                                 res[i][j] = 0
143             else:
144                 res[i][j] = 255
145
146     for i in range(r):
147         for j in range(c):
148             res[i][j] = 255 - res[i][j]
149
150     return res

```

```

72 def minimum_variance(img, threshold):
73     r, c = img.shape
74     res = np.zeros(img.shape, dtype=np.uint8)
75     mask = [2, -1, 2, -1, -4, -1, 2, -1, 2]
76     label = get_label(img, 3, mask, threshold, 1/3)
77     for i in range(r):
78         for j in range(c):
79             if label[i][j] == 1:
80                 for m in range(-1, 2):
81                     for n in range(-1, 2):
82                         if 0 <= i+m < r and 0 <= j+n < c:
83                             if label[i+m][j+n] == 0:
84                                 res[i][j] = 0
85             else:
86                 res[i][j] = 255
87     return res
88
89 def laplacian_gaussian(img, threshold):
90     r, c = img.shape
91     res = np.zeros(img.shape, dtype=np.uint8)
92     mask = [0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0,
93            0, 0, -2, -4, -3, -9, -8, -4, -2, 0, 0,
94            0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0,
95            -1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1,
96            -1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1,
97            -2, -9, -23, -1, 103, 178, 103, -1, -23, -9, -2,
98            -1, -8, -22, -14, 52, 103, 52, -14, -22, -8, -1,
99            -1, -4, -15, -24, -14, -1, -14, -24, -15, -4, -1,
100            0, -2, -7, -15, -22, -23, -22, -15, -7, -2, 0,
101            0, 0, -2, -4, -3, -9, -8, -4, -2, 0, 0,
102            0, 0, 0, -1, -1, -2, -1, -1, 0, 0, 0]
103
104     label = get_label(img, 11, mask, threshold, 1)
105     for i in range(r):
106         for j in range(c):
107
108             if label[i][j] == 1:
109                 for m in range(-5, 6):
110                     for n in range(-5, 6):
111                         if 0 <= i+m < r and 0 <= j+n < c:
112                             if label[i+m][j+n] == 0:
113                                 res[i][j] = 0
114             else:
115                 res[i][j] = 255
116     return res

```

以上為五個 mask 來判斷不同的梯度，也沒特別技巧，都依照上課 PPT 上去做