

Laplacian mask 1 (threshold = 15)

```
mask = [0, 1, 0, 1, -4, 1, 0, 1, 0]
neighbor = [img[i-1][j-1][0], img[i-1][j][0], img[i-1][j+1][0],
            img[i][j-1][0], img[i][j][0], img[i][j+1][0],
            img[i+1][j-1][0], img[i+1][j][0], img[i+1][j+1][0]]
ipgm = sum([mask[tmp] * neighbor[tmp] for tmp in range(len(mask))])
```

- 使用的mask和neighbor如上面所示，根據計算出的input pixel gradient magnitude大於等於 threshold則在map上紀錄為1、小於等於 -threshold則紀錄為-1、其他紀錄為0。
- 記錄完整張map後，針對每個紀錄為1的pixel查看周圍是否有臨域紀錄為-1，如果有則輸出為0，否則跟其他pixel一樣都輸出為255。



Laplacian mask 2 (threshold = 15)

```
mask = [1, 1, 1, 1, -8, 1, 1, 1, 1]
neighbor = [img[i-1][j-1][0], img[i-1][j][0], img[i-1][j+1][0],
            img[i][j-1][0], img[i][j][0], img[i][j+1][0],
            img[i+1][j-1][0], img[i+1][j][0], img[i+1][j+1][0]]
ipgm = (1/3)*(sum([mask[tmp] * neighbor[tmp] for tmp in range(len(mask))]))
```

- 和Laplacian mask 1做法類似，只有使用的mask和計算input pixel gradient magnitude的方式略有不同



minimum-variance Laplacian (threshold = 20)

```
mask = [2, -1, 2, -1, -4, -1, 2, -1, 2]
neighbor = [img[i-1][j-1][0], img[i-1][j][0], img[i-1][j+1][0],
            img[i][j-1][0], img[i][j][0], img[i][j+1][0],
            img[i+1][j-1][0], img[i+1][j][0], img[i+1][j+1][0]]
ipgm = (1/3)*(sum([mask[tmp] * neighbor[tmp] for tmp in range(len(mask))]))
```

- 和Laplacian mask 1做法類似，只有使用的mask和計算input pixel gradient magnitude的方式略有不同



Laplacian of Gaussian (threshold = 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]

ipgm = sum([mask[tmp] * neighbor[tmp] for tmp in range(len(mask))])
```

- 和Laplacian mask 1做法類似，只有使用的mask和neighbor pixel略有不同



Difference of Gaussian (threshold = 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]

ipgm = sum([mask[tmp] * neighbor[tmp] for tmp in range(len(mask))])
```

- 和Laplacian mask 1做法類似，只有使用的mask和neighbor pixel略有不同

