執行環境: python3, PIL, numpy

檔案:

hw10.py laplace1.bmp laplace2.bmp min_var_laplace.bmp

laplace_gauss.bmp difference_gauss.bmp

說明(hw9.py):

mask 的表示方法(以 difference_G 為例):

是一個三元 tuple，第一格表示$\left\lfloor \frac{size}{2} \right\rfloor$，第二格是數值放大大小，第三格則是放大後的數值(以 numpy array 儲存)

```
Gauss1 = np.array(\
[[    0,    0,    0,    0,    1,    1,    1,    0,    0,    0,    0],\
[     0,    0,    1,   14,   55,   88,   55,   14,    1,    0,    0],\
[     0,    1,   36,  362, 1445,  2289, 1445,  362,   36,    1,    0],\
[     0,   14,  362, 3672,14648, 23204,14648, 3672,  362,   14,    0],\
[     1,   55, 1445,14648,58433, 92564,58433,14648, 1445,   55,    1],\
[     1,   88, 2289,23204,92564,146632,92564,23204, 2289,   88,    1],\
[     1,   55, 1445,14648,58433, 92564,58433,14648, 1445,   55,    1],\
[     0,   14,  362, 3672,14648, 23204,14648, 3672,  362,   14,    0],\
[     0,    1,   36,  362, 1445,  2289, 1445,  362,   36,    1,    0],\
[     0,    0,    1,   14,   55,   88,   55,   14,    1,    0,    0],\
[     0,    0,    0,    0,    1,    1,    1,    0,    0,    0,    0]])


Gauss3 = np.array(\
[[  1283,  2106,  3096,  4077,  4809,  5081,  4809,  4077,  3096,  2106,  1283],\
[   2106,  3456,  5081,  6691,  7892,  8339,  7892,  6691,  5081,  3456,  2106],\
[   3096,  5081,  7469,  9836, 11602, 12258, 11602,  9836,  7469,  5081,  3096],\
[   4077,  6691,  9836, 12952, 15277, 16142, 15277, 12952,  9836,  6691,  4077],\
[   4809,  7892, 11602, 15277, 18020, 19040, 18020, 15277, 11602,  7892,  4809],\
[   5081,  8339, 12258, 16142, 19040, 20117, 19040, 16142, 12258,  8339,  5081],\
[   4809,  7892, 11602, 15277, 18020, 19040, 18020, 15277, 11602,  7892,  4809],\
[   4077,  6691,  9836, 12952, 15277, 16142, 15277, 12952,  9836,  6691,  4077],\
[   3096,  5081,  7469,  9836, 11602, 12258, 11602,  9836,  7469,  5081,  3096],\
[   2106,  3456,  5081,  6691,  7892,  8339,  7892,  6691,  5081,  3456,  2106],\
[  1283,  2106,  3096,  4077,  4809,  5081,  4809,  4077,  3096,  2106,  1283]])


difference_G = (5, 1000000, np.subtract(Gauss3, Gauss1))
```

setWhite1 和 setWhite2 是兩種二值化方法，皆接收一陣列 value、位置(r,c)、閾值 threshold。前者直接判斷位置(r,c)處的 value 值是否小於等於閾值，後者則判斷位置(r,c)處的 value 值與其中一個 4-connected neighbors 的 value 值是否有差超過閾值。

```
def setWhite1(value, r, c, threshold):
    return True if value[r][c]<=threshold else False

"""
def setWhite2(value, r, c, threshold):
    if r>0 and np.absolute(value[r][c]-value[r-1][c])>threshold:
        return False
    if r<(R-1) and np.absolute(value[r][c]-value[r+1][c])>threshold:
        return False
    if c>0 and np.absolute(value[r][c]-value[r][c-1])>threshold:
        return False
    if c<(C-1) and np.absolute(value[r][c]-value[r][c+1])>threshold:
        return False
    return True
"""
```

generatePicture 接收四個參數: 輸出圖檔名 name、ker 是 mask、閾值 threshold、二值化方法 setWhite。

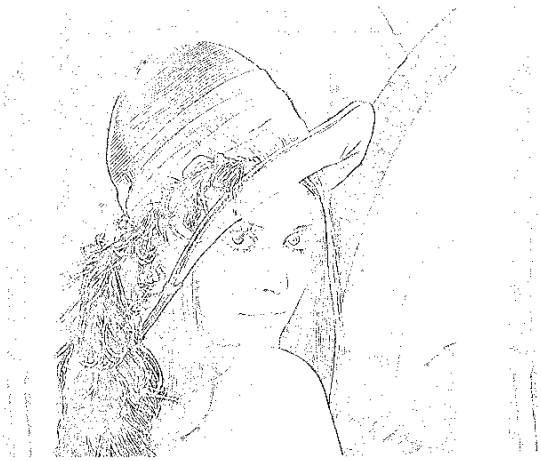
```
91  def generatePicture(name, ker, threshold, setWhite):
92      value = np.zeros((R,C), dtype=np.float32)
93      ret = np.zeros((R,C), dtype=np.uint8)
94      for r in range(5, 5+R):
95          for c in range(5, 5+C):
96              value[r-5][c-5] = sum((arr[r-ker[0]:r+ker[0]+1, c-ker[0]:c+ker[0]+1] * ker[2]).flatten()) / ker[1]
97      for r in range(R):
98          for c in range(C):
99              ret[r][c] = 255 if setWhite(value, r, c, threshold) else 0
100     Image.fromarray(ret).save(name)
```

最後選擇的二值化方法是 setWhite1，也就是只看自己是否大於閾值。

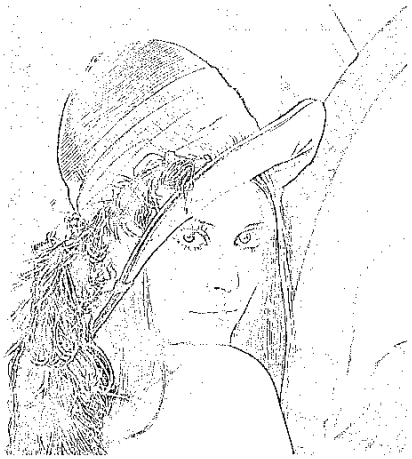
```
102 generatePicture('laplace1.bmp', laplace1, 30, setWhite1)
103 generatePicture('laplace2.bmp', laplace2, 20, setWhite1)
104 generatePicture('min_var_laplace.bmp', mv_laplace, 20, setWhite1)
105 generatePicture('laplace_gauss.bmp', laplace_G, 6000, setWhite1)
106 generatePicture('difference_gauss.bmp', difference_G, 6, setWhite1)
```

結果:

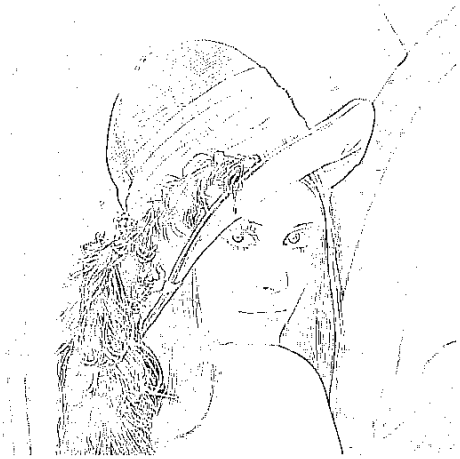
laplace1.bmp



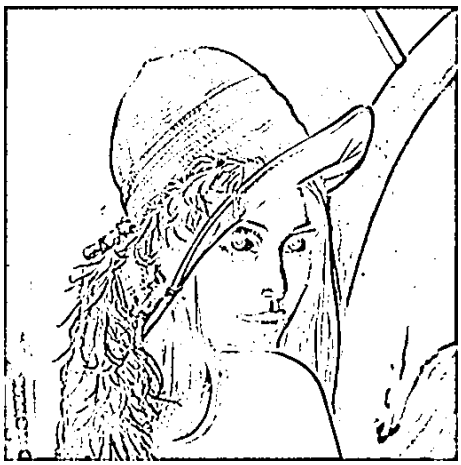
laplace2.bmp



min_var_laplace.bmp



laplace_gauss.bmp



difference_gauss.bmp