

HW10 report

姓名:徐有慶 學號:R05922162

1. 作業要求：

Zero Crossing Edge Detection

- You are to implement Laplacian, Minimum Variance Laplacian, Laplacian of Gaussian, and Difference of Gaussian(inhibitory sigma=1, excitatory sigma=3, kernel size 11x11 [1][1])
- Threshold Values listed below are for reference:

(僅供參考，同學可自己找出 Edge Image 品質最佳的門檻值 threshold value)

- Laplace Mask (0, 1, 0, 1, -4, 1, 0, 1, 0): 15
- Minimum variance Laplacian: 20
- Laplace of Gaussian: 3000

程式語言：Java

執行環境：Eclipse

2. 程式設計:

- Laplacian:

Kernel 如下，threshold = 15:

	1	
1	-4	1
	1	

- Minimum Variance Laplacian:

Kernel 如下，threshold = 20:

$$\frac{1}{3} \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

- Laplacian of Gaussian:

Kernel 如下，threshold = 3000:

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

- Difference of Gaussian:

利用下列公式建立出 kernel，分別計算出 $G1(\sigma = 1)$, $G2(\sigma = 3)$ ，再算出 $G1-G2$ 當作 mask，

$$Gaussian: \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}(\frac{r^2+c^2}{\sigma^2})}$$

其中，kernel size = 11。

Kernel 如下，threshold = 1:

```
-0.00239325325350624 -0.0023932529277181716 -0.006505530247119942 -0.0065054044941294945 -0.006505177093437326 -0.006504558954268024 -0.006505177093437326 -0.0065054044941294945 -0.006505530247119942 -0.0023932529277181716 -0.00239325325350624
-0.0023932529277181716 -0.0065055189255061345 -0.006504558954268024 -0.006498311212797738 -0.017630492030312868 -0.017630492030312868 -0.017630492030312868 -0.006498311212797738 -0.006504558954268024 -0.0065055189255061345 -0.0023932529277181716
-0.006505530247119942 -0.006504558954268024 -0.006485895555689065 -0.017289376904149903 -0.016611504994570495 -0.014768858100737957 -0.016611504994570495 -0.017289376904149903 -0.006485895555689065 -0.006504558954268024 -0.006505530247119942
-0.0065054044941294945 -0.006498311212797738 -0.017289376904149903 -0.014768858100737957 -0.00385539673608248 -0.00385539673608248 -0.00385539673608248 -0.014768858100737957 -0.017289376904149903 -0.006498311212797738 -0.0065054044941294945
-0.006505177093437326 -0.017630492030312868 -0.016611504994570495 -0.00385539673608248 -0.040865948958553014 -0.1414710605261292 -0.040865948958553014 -0.00385539673608248 -0.016611504994570495 -0.017630492030312868 -0.006505177093437326
-0.006504558954268024 -0.017630492030312868 -0.014768858100737957 -0.00385539673608248 -0.1414710605261292 -0.1414710605261292 -0.00385539673608248 -0.014768858100737957 -0.017630492030312868 -0.006504558954268024
-0.006505177093437326 -0.017630492030312868 -0.016611504994570495 -0.00385539673608248 -0.040865948958553014 -0.1414710605261292 -0.040865948958553014 -0.00385539673608248 -0.016611504994570495 -0.017630492030312868 -0.006505177093437326
-0.0065054044941294945 -0.006498311212797738 -0.017289376904149903 -0.014768858100737957 -0.00385539673608248 -0.00385539673608248 -0.00385539673608248 -0.014768858100737957 -0.017289376904149903 -0.006498311212797738 -0.0065054044941294945
-0.006505530247119942 -0.006504558954268024 -0.006485895555689065 -0.017289376904149903 -0.016611504994570495 -0.014768858100737957 -0.016611504994570495 -0.017289376904149903 -0.006485895555689065 -0.006504558954268024 -0.006505530247119942
-0.0023932529277181716 -0.0065055189255061345 -0.006504558954268024 -0.006498311212797738 -0.017630492030312868 -0.017630492030312868 -0.017630492030312868 -0.006498311212797738 -0.006504558954268024 -0.0065055189255061345 -0.0023932529277181716
-0.00239325325350624 -0.0023932529277181716 -0.006505530247119942 -0.0065054044941294945 -0.006505177093437326 -0.006504558954268024 -0.006505177093437326 -0.0065054044941294945 -0.006505530247119942 -0.0023932529277181716 -0.00239325325350624
```

3. 主要程式：

Main(/Source Code/src/hw10/MainZeroCrossing.java):

```
public class MainZeroCrossing {
    public static void main(String[] args) throws IOException {
        int[][] img = FileProcess.inputImg(new File("lena.bmp"));
        int[][] outputImg = null;
        List<Point> mask = new ArrayList<>();

        mask = Mask.getLaplacian();
        outputImg = ZeroCrossingEdgeDectection.operator(img, mask);
        outputImg = ImageProcess.binaryImage(outputImg, 15);
        FileProcess.outputImg(outputImg, "laplacian_lena.bmp");

        mask = Mask.getMinimunVarianceLaplacian();
        outputImg = ZeroCrossingEdgeDectection.operator(img, mask);
        outputImg = ImageProcess.binaryImage(outputImg, 20);
        FileProcess.outputImg(outputImg, "variance_lena.bmp");

        mask = Mask.getLOG();
        outputImg = ZeroCrossingEdgeDectection.operator(img, mask);
        outputImg = ImageProcess.binaryImage(outputImg, 3000);
        FileProcess.outputImg(outputImg, "log_lena.bmp");

        mask = Mask.getDOG();
        outputImg = ZeroCrossingEdgeDectection.operator(img, mask);
        outputImg = ImageProcess.binaryImage(outputImg, 1);
        FileProcess.outputImg(outputImg, "dog_lena.bmp");
    }
}
```

Zero crossing edge detection(./Source Code/src/hw10/ZeroCrossingEdgeDection.java):

```
public class ZeroCrossingEdgeDection {
    public static int[][] operator(int[][] inputImg, List<Point> mask) {
        int height = inputImg.length;
        int width = inputImg[0].length;
        int[][] outputImg = new int[height][width];
        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                double newPixel = 0;
                for (Point point : mask) {
                    int x = point.getX();
                    int y = point.getY();
                    double value = point.getValue();
                    int pixel = i + x < height && j + y < width && i + x > 0 && j + y > 0 ? inputImg[i + x][j + y] : 0;
                    newPixel = newPixel + value * pixel;
                }
                outputImg[i][j] = (int) Math.round(newPixel);
            }
        }
        return outputImg;
    }
}
```

4. 執行結果：

Laplacian (Threshold: 15), Minimum Variance Laplacian (20), Laplacian of Gaussian (3000),

Difference of Gaussian (1):



5. 如何執行

執行./Source Code/ hw10.jar 即可產生作業要求的兩種圖片(lena.bmp 需要和 hw10.jar 在同一個資料夾底下)

程式進入點為 ./Source Code/src/hw10/ MainZeroCrossing.java