Computer vision
Homework 10

B06902091 資工四 羅寶瑩

Zero Crossing Edge Detection

Description

Implement 2 Laplacian Mask, Minimum Variance Laplacian, Laplacian of Gaussian, and Difference of Gaussian(inhibitory sigma=3, excitatory sigma=1, kernel size 11x11).

1. *Some functions take homework 9 for reference. All of the kernel and threshold values use the reference in the lecture slide.*
2. *For the laplacian output pixel t, I have changed the value from -1 to 100.*

Input pixel gradient magnitude >= threshold (15)
→ Laplacian output pixel t = 1
Input pixel gradient magnitude <= -threshold (15)
→ Laplacian output pixel t = ~~-1~~ 100
Else → Laplacian output pixel t = 0

(a) Laplace Mask 1
(1, 1, 1, 1, -8, 1, 1, 1, 1): 15

(threshold = 15)

	1	
1	-4	1
	1	

(b) Laplace Mask 2
(0, 1, 0, 1, -4, 1, 0, 1, 0): 15

(threshold = 15)

$\frac{1}{3}$

1	1	1
1	-8	1
1	1	1

(c) Minimum variance Laplacian: 20

(threshold = 20)

$\frac{1}{3}$

2	-1	2
-1	-4	-1
2	-1	2

Laplacian output (section)

```
int result = 0;
for (int a = 0; a < 3; a++) {
    for (int b = 0; b < 3; b++) {
        result += array[a][b] * kernel[a][b];
    }
}
result /= constant;

if (result >= threshold) {
    tmp.at<uchar>(i, j) = 1;
}
else if (result <= (threshold * -1)) {
    tmp.at<uchar>(i, j) = 100;  //-1 -> 100
}
else {
    tmp.at<uchar>(i, j) = 0;
}
```

Zero-crossing (section)

```
if (tmp.at<uchar>(i, j) == 100 || tmp.at<uchar>(i, j) == 0) {
    output.at<uchar>(i, j) = 255;
    continue;
}
int flag = 0;
for (int a = 0; a < 3; a++) {
    for (int b = 0; b < 3; b++) {
        if (a == 1 && b == 1) {
            continue;
        }
        if (array[a][b] == 100) {
            flag = 1;
        }
    }
}
if (flag == 1) {
    output.at<uchar>(i, j) = 0;
}
else {
    output.at<uchar>(i, j) = 255;
}
```

(For more information, please refer to lines 14~171.)

(d) Laplace of Gaussian: 3000

(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

(e)Difference of Gaussian: 1

(threshold = 1)

-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

Laplacian output (section)

```
int result = 0;
int array[11][11];
for (int a = 0; a < 11; a++) {
    for (int b = 0; b < 11; b++) {
        result += ori.at<uchar>(i - 5 + a, j - 5 + b) * kernel[a][b];
    }
}

result /= constant;

if (result >= threshold) {
    tmp.at<uchar>(i, j) = 1;
}
else if (result <= (threshold * -1)) {
    tmp.at<uchar>(i, j) = 100;  //-1 -> 100
}
else {
    tmp.at<uchar>(i, j) = 0;
}
```

Zero-crossing (section)

```
if (tmp.at<uchar>(i, j) == 100 || tmp.at<uchar>(i, j) == 0) {
    output.at<uchar>(i, j) = 255;
    continue;
}
int flag = 0;
for (int a = 4; a < 7; a++) {
    for (int b = 4; b < 7; b++) {
        if (a == 5 && b == 5) {
            continue;
        }
        if (tmp.at<uchar>(i - 5 + a, j - 5 + b) == 100) {
            flag = 1;
        }
    }
}
if (flag == 1) {
    output.at<uchar>(i, j) = 0;
}
else {
    output.at<uchar>(i, j) = 255;
}
```

(For more information, please refer to lines 173~227.)

main and function

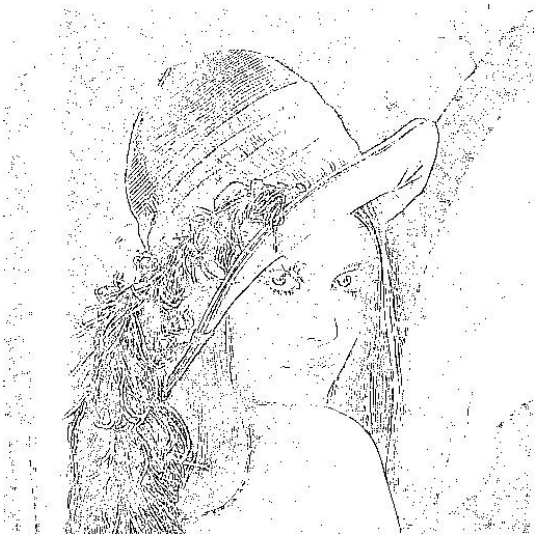
```
14   void laplacian(Mat ori, Mat output, int threshold, int kernel[3][3], int constant) { ... }
172
173  void gaussian(Mat ori, Mat output, int threshold, int kernel[11][11], int constant) { ... }
228
229  int main() {
230      Mat original;
231      original = imread("lena.bmp", CV_LOAD_IMAGE_GRAYSCALE);
232      img_rows = original.rows;
233      img_cols = original.cols;
234
235      int kernel1[3][3] = { 0,1,0,1,-4,1,0,1,0 };
236      Mat laplacian_o(img_rows, img_cols, CV_8UC1, Scalar(255));
237      laplacian(original, laplacian_o, 15, kernel1, 1);
238      imwrite("1_laplacian_1.jpg", laplacian_o);
239
240      int kernel2[3][3] = { 1,1,1,1,-8,1,1,1,1 };
241      Mat laplacian_2o(img_rows, img_cols, CV_8UC1, Scalar(255));
242      laplacian(original, laplacian_2o, 15, kernel2, 3);
243      imwrite("2_laplacian_2.jpg", laplacian_2o);
244
245      int kernel3[3][3] = { 2,-1,2,-1,-4,-1,2,-1,2 };
246      Mat laplacian_3o(img_rows, img_cols, CV_8UC1, Scalar(255));
247      laplacian(original, laplacian_3o, 20, kernel3, 3);
248      imwrite("3_laplacian_3.jpg", laplacian_3o);
```

```
250      int kernel4[11][11] = {
251          0,0,0,-1,-1,-2,-1,-1,0,0,0
252          ,0,0,-2,-4,-8,-9,-8,-4,-2,0,0
253          ,0,-2,-7,-15,-22,-23,-22,-15,-7,-2,0
254          ,-1,-4,-15,-24,-14,-1,-14,-24,-15,-4,-1
255          , -1,-8,-22,-14,52,103,52,-14,-22,-8,-1
256          ,-2,-9,-23,-1,103,178,103,-1,-23,-9,-2
257          ,-1,-8,-22,-14,52,103,52,-14,-22,-8,-1
258          ,-1,-4,-15,-24,-14,-1,-14,-24,-15,-4,-1
259          ,0,-2,-7,-15,-22,-23,-22,-15,-7,-2,0
260          ,0,0,-2,-4,-8,-9,-8,-4,-2,0,0
261          ,0,0,0,-1,-1,-2,-1,-1,0,0,0 };
262
263      Mat laplacian_4o(img_rows, img_cols, CV_8UC1, Scalar(255));
264      gaussian(original, laplacian_4o, 3000, kernel4, 1);
265      imwrite("4_laplacian_4.jpg", laplacian_4o);
266
267      int kernel5[11][11] = {
268          -1, -3,-4,-6,-7,-8,-7,-6,-4,-3,-1
269          ,-3,-5,-8,-11,-13,-13,-13,-11,-8,-5,-3
270          ,-4,-8,-12,-16,-17,-17,-17,-16,-12,-8,-4
271          ,-6,-11,-16,-16,0,15,0,-16,-16,-11,-6
272          ,-7,-13,-17,0,85,160,85,0,-17,-13,-7
273          ,-8,-13,-17,15,160,283,160,15,-17,-13,-8
274          ,-7,-13,-17,0,85,160,85,0,-17,-13,-7
275          ,-6,-11,-16,-16,0,15,0,-16,-16,-11,-6
276          ,-4,-8,-12,-16,-17,-17,-17,-16,-12,-8,-4
277          ,-3,-5,-8,-11,-13,-13,-13,-11,-8,-5,-3
278          ,-1,-3,-4,-6,-7,-8,-7,-6,-4,-3,-1};
279
280      Mat laplacian_5o(img_rows, img_cols, CV_8UC1, Scalar(255));
281      gaussian(original, laplacian_5o, 1, kernel5, 1);
282      imwrite("5_laplacian_5.jpg", laplacian_5o);
```

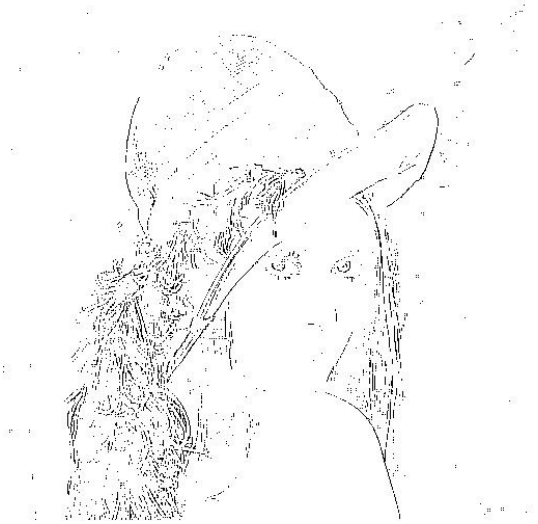
Result

(a) Laplace Mask 1 (0, 1, 0, 1, -4, 1, 0, 1, 0): 15

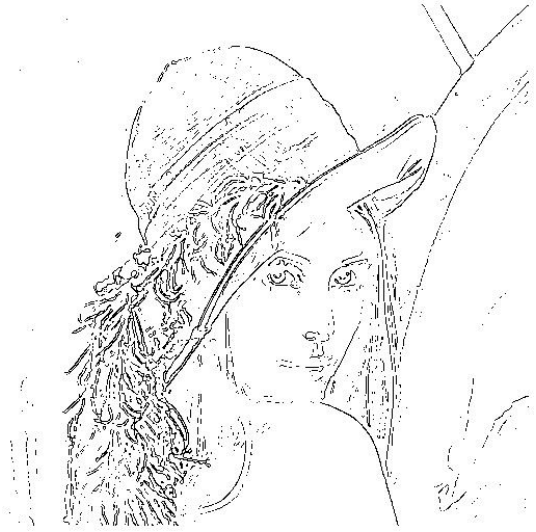
(b) Laplace Mask 2 (1, 1, 1, 1, -8, 1, 1, 1, 1): 15



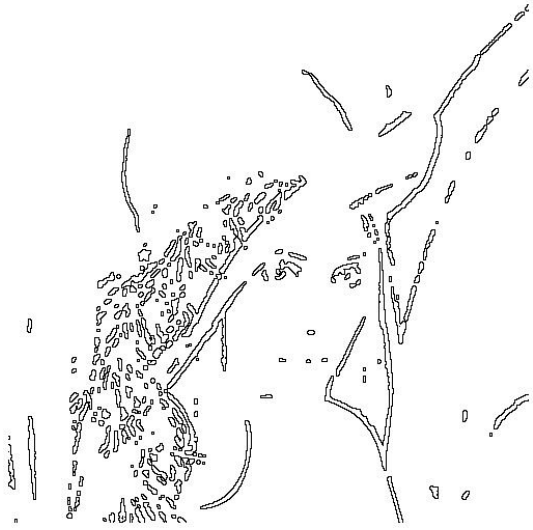
(c) Minimum variance Laplacian: 20



(d) Laplace of Gaussian: 3000



(e) Difference of Gaussian: 1



Reference:

1. *lecture slide*
   http://cv2.csie.ntu.edu.tw/CV/_material/CH7_HW9_10%E8%AC%9B%E8%A7%A3(v4).pdf