Image Processing Homework #4

Implementations of Sobel Operator and Laplacian of a Gaussian Operator

Advisor : Jin-Jang Leou (柳金章)
Student : Chih-Chia Lo (羅治嘉)
Registration Number : 610410002
Date Due : 2022/01/07 23:59
Date Handed In : 2021/12/23 17:00

Outline

I. Technical Description

A. Sobel operators

Sobel operators are the two equations use a weight of 2 in the center coefficient:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3), \tag{1}$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7), \tag{2}$$

where G_x denotes the horizontal gradient, G_y denotes the vertical gradient, and $z_{i\in(1,2,...,9)}$ denotes a 3x3 region of image with gray-level values. Then, combine G_x and G_y to obtain the Sobel image, the equation is as follow,

$$\nabla f = \sqrt{{G_x}^2 + {G_y}^2}\,. \tag{3}$$

In summary, take input image and the two Sobel operators into convolution one-by-one. After that, calculate equation (3) to acquire edge detection image. The source code is in the following figure.

```
5   def my_sobel(img):
6       hs = np.array([[-1.0, -2.0, -1.0], [0.0, 0.0, 0.0], [1.0, 2.0, 1.0]]) #Horizontal sobel
7       vs = np.array([[-1.0, 0.0, 1.0], [-2.0, 0.0, 2.0], [-1.0, 0.0, 1.0]]) #Vertical sobel
8       [rows, cols] = np.shape(img)
9       sobel_result = np.zeros(shape = [rows, cols])
10      for i in range(1, rows - 1):
11          for j in range(1, cols - 1):
12              Gx = (hs * img[i-1:i+2, j-1:j+2]).sum()
13              Gy = (vs * img[i-1:i+2, j-1:j+2]).sum()
14              sobel_result[i,j] = np.sqrt(Gx ** 2 + Gy ** 2)
15              if sobel_result[i,j] > 255: #Avoid overflow
16                  sobel_result[i,j] = 255
17              elif sobel_result[i,j] < 0:
18                  sobel_result[i,j] = 0
19      return sobel_result
```

Fig. 1: Handcrafted function of Sobel operators.

B. Laplacian of a Gaussian (LoG) operator

The Laplacian is combined with smoothing as a precursor to finding edges via zero-crossings. Consider the smoothing function:

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}, \tag{4}$$

where r^2 denotes the summation of x^2 and y^2, and σ denotes the standard deviation. The Laplacian of h (the second derivative of h with respect to r) is defined as:

$$\nabla^2 h(r) = -[\frac{r^2 - \sigma^2}{\sigma^4}]e^{-\frac{r^2}{2\sigma^2}}. \quad (5)$$

This function is commonly referred to as the LoG because $h(r)$ is in the form of a Gaussian function. [1] The following figure shows that a 5x5 mask approximates $\nabla^2 h$.

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Fig. 2: LoG 5x5 mask.

The mask is also called the Mexican hat function due to its shape. In summary, take input image and the LoG operator into convolution to acquire edge detection image. The source code is in the following figure.

```
21  def my_LoG(img):
22      LoG = np.array([[0.0, 0.0, -1.0, 0.0, 0.0],
23                      [0.0, -1.0, -2.0, -1.0, 0.0],
24                      [-1.0, -2.0, 16.0, -2.0, -1.0],
25                      [0.0, -1.0, -2.0, -1.0, 0.0],
26                      [0.0, 0.0, -1.0, 0.0, 0.0]])
27      [rows, cols] = np.shape(img)
28      LoG_result = np.zeros(shape = [rows, cols])
29      for i in range(2, rows - 2):
30          for j in range(2, cols - 2):
31              LoG_result[i,j] = (LoG * img[i-2:i+3, j-2:j+3]).sum()
32              if LoG_result[i,j] > 255: #Avoid overflow
33                  LoG_result[i,j] = 255
34              elif LoG_result[i,j] < 0:
35                  LoG_result[i,j] = 0
36      return LoG_result
```

Fig. 3: Handcrafted function of the LoG operator.

II. Experimental Results

A. Datasets and Environments

This project contains a dataset of HW4_test_image from ecourse2, which three gray-level images are given in this dataset. The programming language of this project is Python 3.9.6 64-bit.

B. Edge Detection

The following figure is the result of using Sobel operator and the LoG operator to do edge detection:

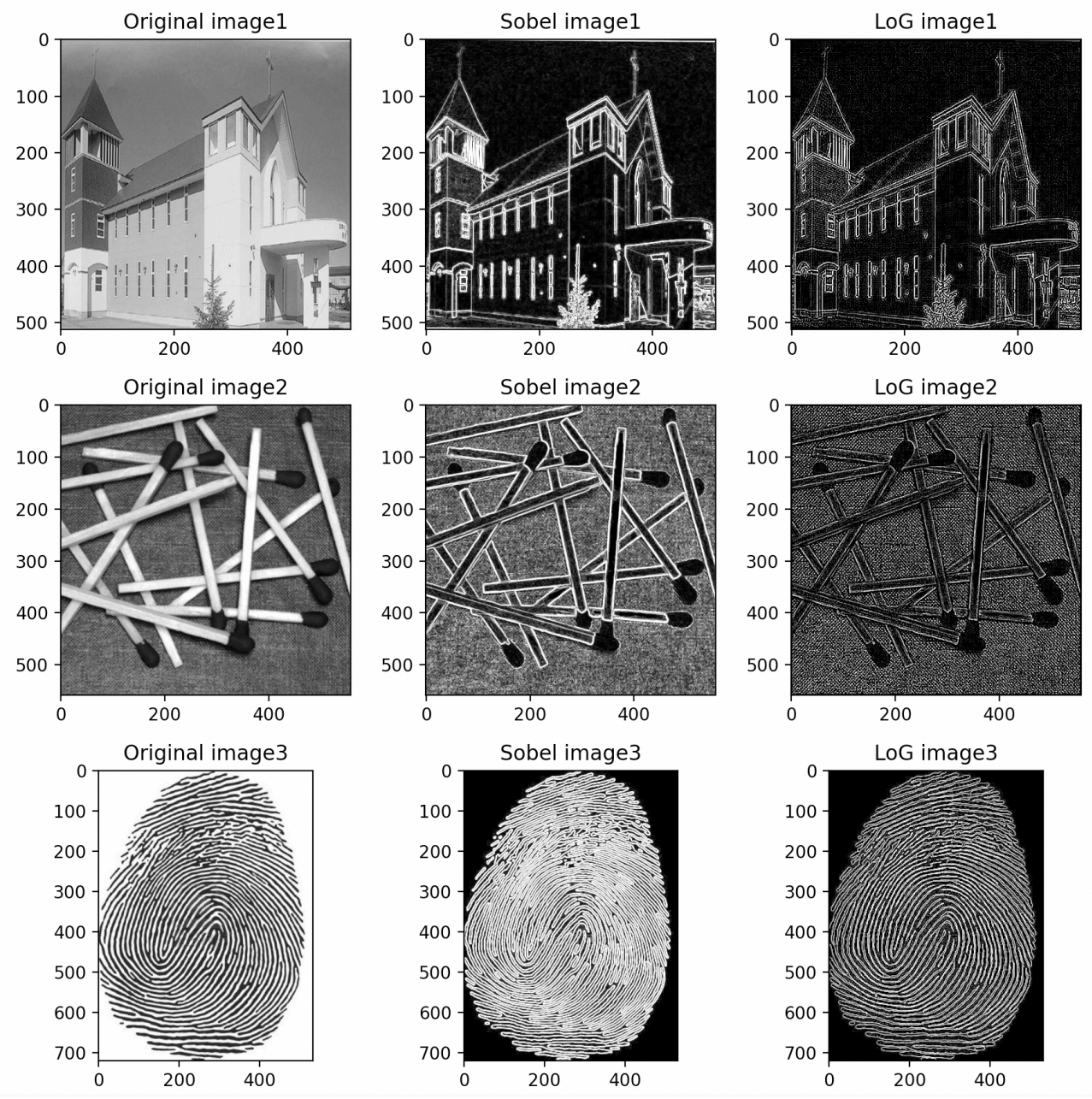

Fig. 4: Edge detection result.

where the first column displays original image, the second column displays Sobel edge detection image, and the third column displays the LoG edge detection image.

III. Discussions

A. Edge Detection

In my opinion, Sobel edge detection has better performances in both image1 and image2 because the edges are deeply illustrated and clear, while the LoG edge detection in image1 is a little bit vague, and in image2, the edges are severely vague. However, the LoG edge detection is better in image3 because the edges in Sobel edge detection image3 are too rough, basically has the most image covered with edges. It seems that the LoG edge detection can handle better in image that contains dense groups of edges. Therefore, when it comes to choosing the best edge detection approaches, it depends on the situation of the image.

IV. References and Appendix

[1] R. C. Gonzalez and R. E. Woods, “Digital Image Processing, 4th Ed., Pearson, New York, NY, 2018.” in Chapter 10 Color Image Segmentation LoG operator.