tags: `DIP`

Principles and Applications of Digital Image Processing

> Fall, 2021
>
> for better reading: https://hackmd.io/@tohow06/BJsMY6nIY

Homework 3

B06611008 賴乙豪

Part 1:

3.22

(a)

$$\omega = vw^T = \begin{bmatrix}1\\2\\1\end{bmatrix}\begin{bmatrix}2 & 1 & 1 & 3\end{bmatrix} = \begin{bmatrix}2 & 1 & 1 & 3\\4 & 2 & 2 & 6\\2 & 1 & 1 & 3\end{bmatrix}$$

Yes, it is separable. It can be separated to two column vectors' (v and w) outer product.

(b)

$$\begin{bmatrix}1 & 3 & 1\\2 & 6 & 2\end{bmatrix} = \begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}1 & 3 & 1\end{bmatrix}$$

$$w_1 = \begin{bmatrix}1\\2\end{bmatrix} \quad w_2 = \begin{bmatrix}1\\3\\1\end{bmatrix}$$

3.27

(a)

According (3-38), (3-39) in textbook, the single kernel size is
$W_v = W_h = 4 \times (3-1) + 3 = 11$
kernel size is 11x11.

(b)

According Table 3.6, standard deviation of convolution of two Gaussian functions is
$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$
So the standard deviation of the single kernel will be
$\sigma = \sqrt{1^2 + 1^2 + 1^2 + 1^2} = 2$

3.38

If the order is reversed, the detail and the noise will be enhanced first by Laplacian kernel, then the smoothing kernel can only blur all the detail and noise. With the first order we can get clean and fine detailed image, however, with the reversed order, we get blurred image with somewhat mess.

4.3

(a)

Convolution in the frequency domain is analogous to multiplication in the spatial domain, so I first transform two functions

$$\begin{aligned} \Im\left\{\delta(t)\right\} &= 1 \\ \Im\left\{\delta(t - t_0)\right\} &= e^{-j2\pi\mu t_0} \end{aligned}$$

Because the first one equals 1, the result of convolution is $\delta(t - t_0)$.

(b)

Same process with (a)

$$\begin{aligned} \Im\left\{\delta(t + t_0)\right\} &= e^{j2\pi\mu t_0} \\ \Im\left\{\delta(t - t_0)\right\} &= e^{-j2\pi\mu t_0} \\ e^{j2\pi\mu t_0} \times e^{-j2\pi\mu t_0} &= 1 \\ \Im^{-1}\left\{1\right\} &= \delta(t) \end{aligned}$$

The result of convolution is $\delta(t)$.

4.32

(a)

Origin sequence of array is even. Insert 0's in center of array become
$\{a, b, c, 0, 0, 0, 0, 0, 0, 0, 0, 0, c, b\}$
The new array still has evenness.

(b)

Origin sequence of array is odd. Insert 0's in center of array become
$\{0, -b, -c, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, c, b\}$
The new array still has oddness.

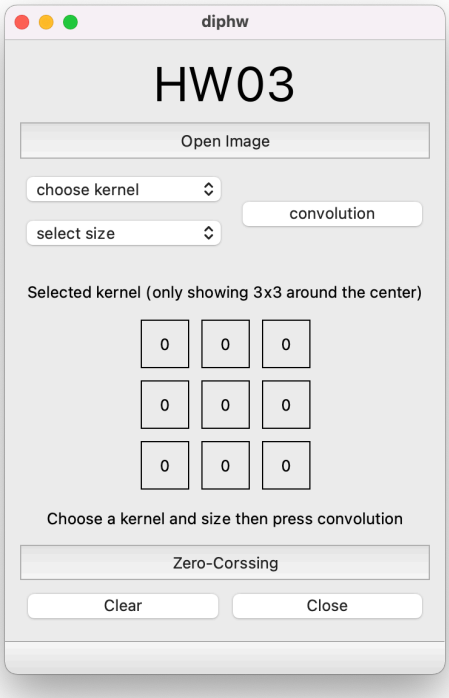
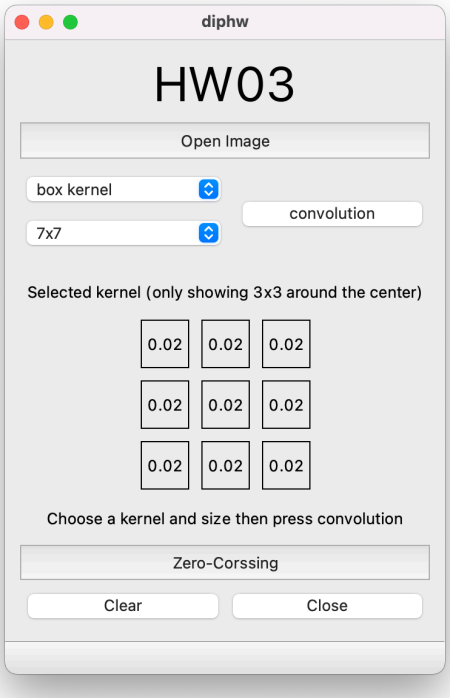
(c)

Origin sequence of array is even. Insert 0's become
$\{a, b, c, 0, 0, 0, 0, 0, d, 0, 0, 0, 0, 0, c, b\}$
The new array still has evenness.

(d)

Origin sequence of array is odd. Insert 0's in center of array become
$\{0, -b, -c, 0, 0, 0, 0, 0, 0, 0, 0, 0, c, b\}$
The new array still has oddness.

Program Usage

default	after choose 7x7 box kernel
	

1. **open image**
2. choose kernel type
3. choose kernel size
4. **convolution**
    - execution time will show in third-to-last line

5. **Clear** to close image
6. **Close** to exit the program

> I do the preprocess to map rpg image to gray scale and add suitable padding before convolution

Part 2:

result

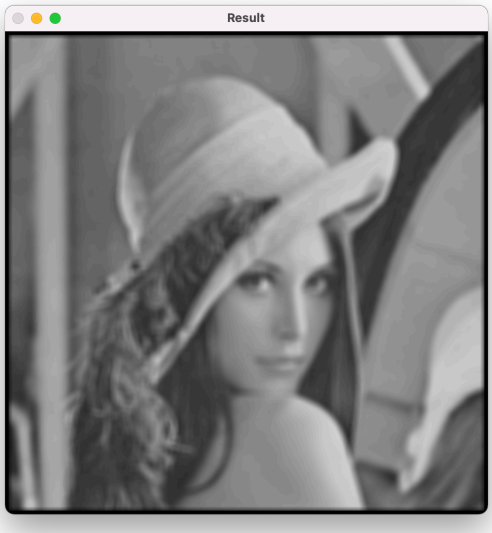
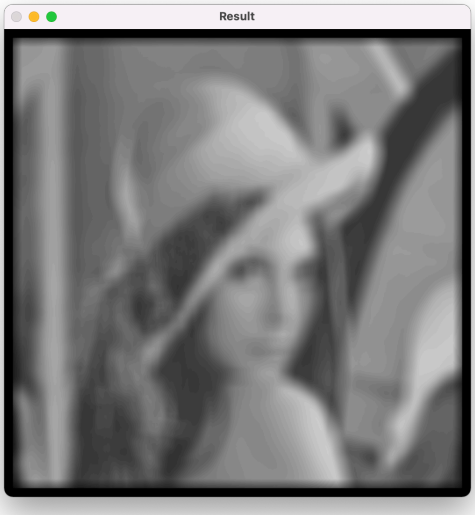
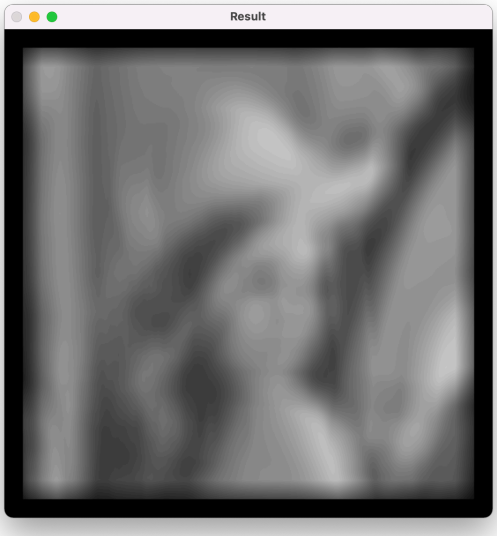
have various types of masks to use

- box
- Gaussian
- LoG
- Marr-Hildreth
- Sobel
- max
- median

and can use with several sizes

- 3, 5, 7, 9, 11, 21, 25, 43, 85

Take box filter for example

3x3 box execution time: 40ms	9x9 box execution time: 256ms
	
21x21 box excution time: 1390ms	43x43 box execution time: 5750ms
	

Discuss

when mask's size bigger

1. the filter is more powerful
    - although it's not obvious when using Gaussian
2. the execution time will be longer

Part 3:

method

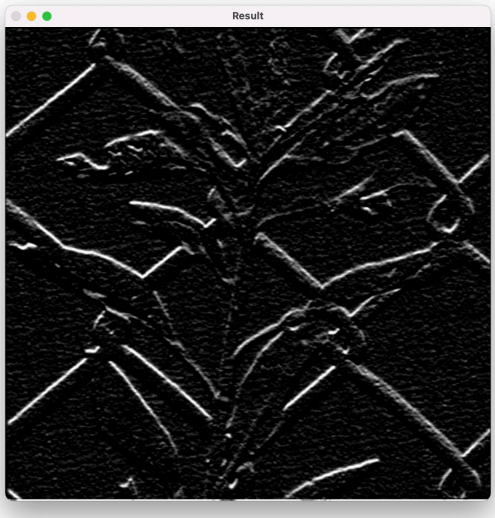
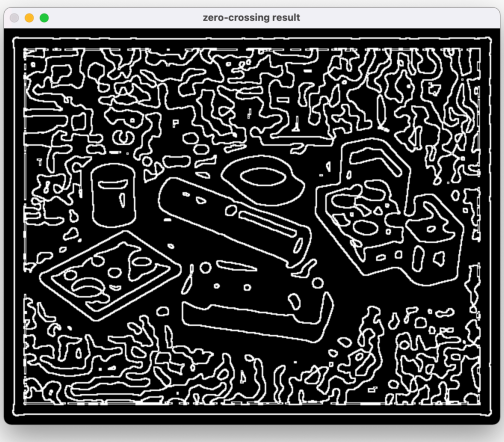
Marr-Hildreth

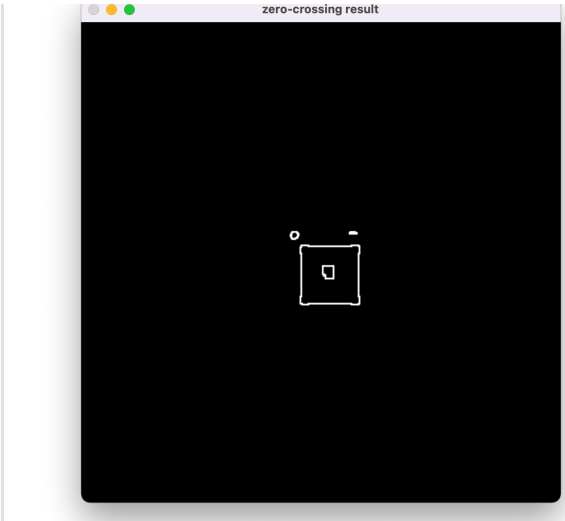
1. covert to grayscale
2. scale to [0 1]
3. conv2 25x25 Gaussian and Laplacian(1 1 1 1 -8 1 1 1)
4. scale to [0 255]
5. zero-crossing
    - find 3x3 neightbors which have different sign and difference > threshold

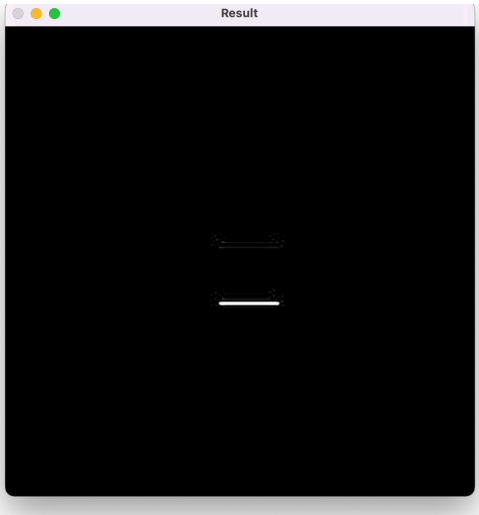
result

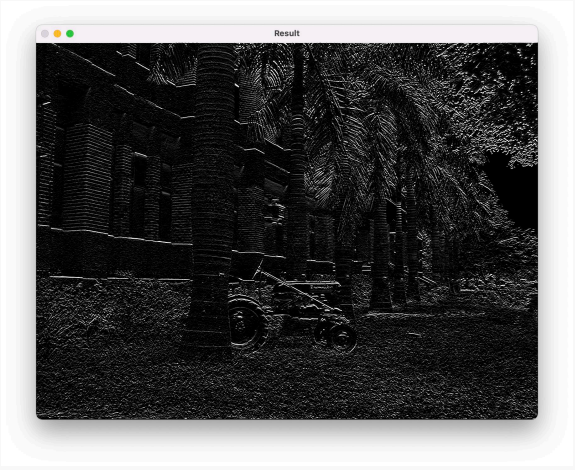
Marr-Hildreth Sobel comparison

> zero-crossing threshold = 0

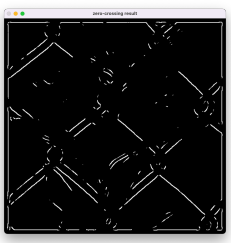
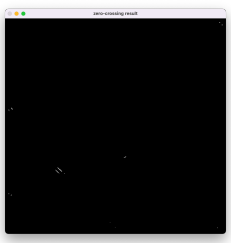
Marr-Hildreth	Sobel
	
	







zero-crossing threshold with Marr-Hildreth comparison

thres = 0% of max value	25%	50%	75%
			

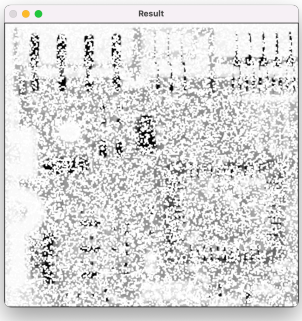
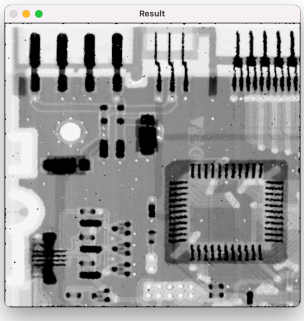
Discussion

With 0% threahold, all edges can be detected, however, some noise will also show in image.
In this example, with 25%, some noise we don't want will be removed, and show a nice result.
When therhold more the 50%, all most everything were removed.

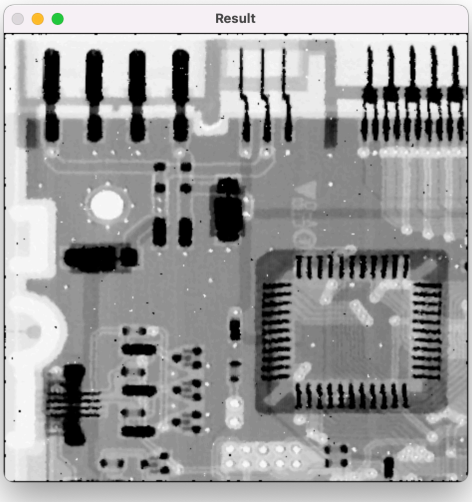
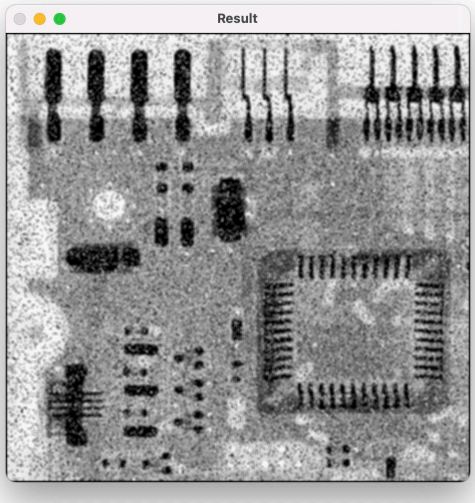
Part 4:

result

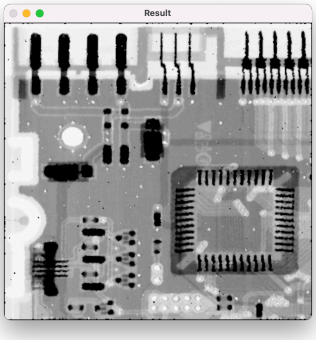
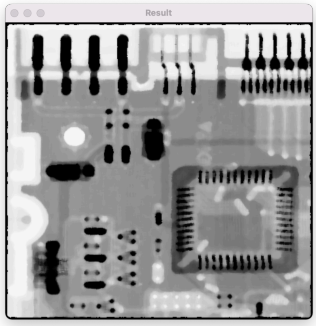
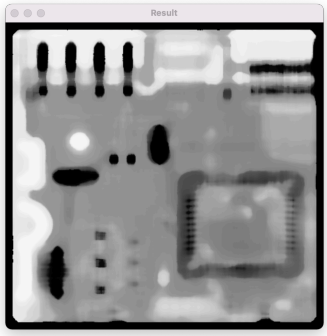
max median min filter comparion

3x3 max	3x3 median	3x3 min
		

median Gaussian filter comparion

3x3 median	3x3 Gaussian sigma=1
	

different size median filter comparion

3x3 median	7x7 median	21x21 median
		

Discuss

1. Comparing median to Gaussian filter can find out that median filter can remove the noise points which spread on monotonous blocks of the image, however Gaussian filter smoothed the entire image.
2. Comparing the filter size, again, the larger filter will cause stronger effect. If we want to reduce the noise, 21x21 median may be too large.