HW9 Report

R10922067 林云雲

Introduction

This program does edge detection to an image with *Robert's Operator, Prewitt's Edge Detector, Sobel's Edge Detector, Frei-and-Chen's Gradient Operator, Kirsch's Compass Operator, Robinson's Compass Operator, Nevatia-Babu 5x5 Operator*. Each of these methods defines its masks to calculate gradient magnitude for every pixel to generate the output image.

Usage

Place the source image and main.py under the same directory. Run the following command in the terminal.

```
python3 main.py -s <source> -m <method> -t <threshold>
```

Parameters

`-s <source>` : the file path of source image, default = `lena.bmp`
`-m <method>` : the method for edge detection, default = `Robert`
*Options:
-`Robert`
-`Prewitt`
-`Sobel`
-`FreiAndChen`
-`Kirsch`
-`Robinson`
-`Nevatia-Babu`
`-t <threshold>` : the threshold for edge detection, default = `12`

Robert's Operator

1. Masks

r_1

-1	
	1

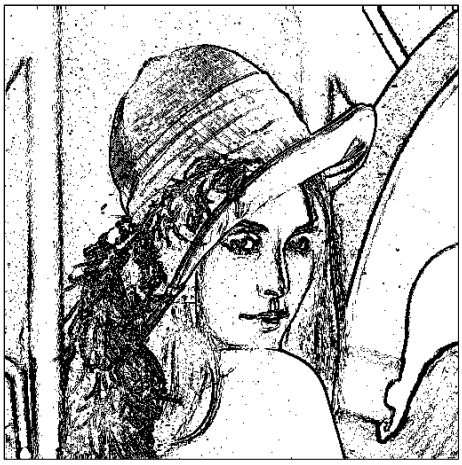
r_2

	-1
1	

2. Gradient Magnitude

$gradient\ magnitude = \sqrt{{r_1}^2 + {r_2}^2}$

3. The result (threshold=12 for example) is saved as `robert.png` as shown on the right.



Prewitt's Edge Detector

1. Masks

p_1

-1		1
-1		1
-1		1

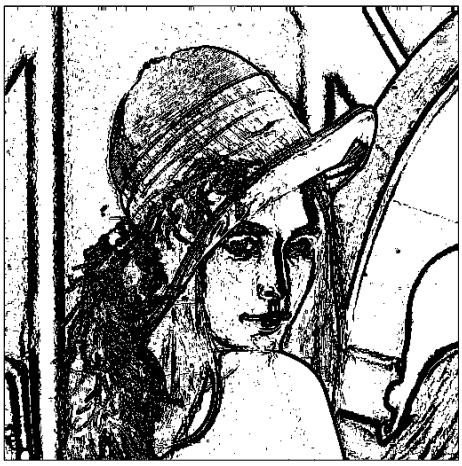
p_2

-1	-1	-1
1	1	1

2. Gradient Magnitude

$gradient\ magnitude = \sqrt{{p_1}^2 + {p_2}^2}$

3. The result (threshold=24 for example) is saved as `prewitt.png` as shown on the right.



Sobel's Edge Detector

1. Masks

s_1

-1		1
-2		2
-1		1

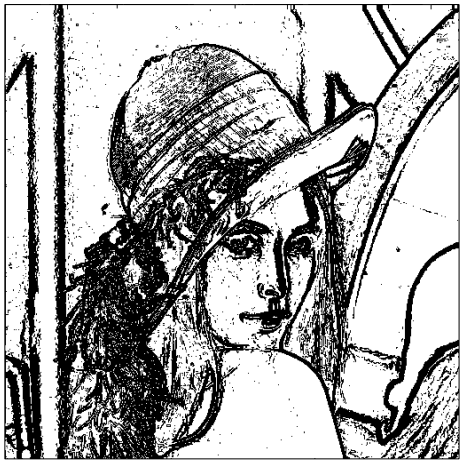
s_2

-1	-2	-1
1	2	1

2. Gradient Magnitude

$gradient\ magnitude = \sqrt{{s_1}^2 + {s_2}^2}$

3. The result (threshold=38 for example) is saved as `sobel.png` as shown on the right.



Frei-and-Chen's Gradient Operator

1. Masks

f_1

-1		1
-√2		√2
-1		1

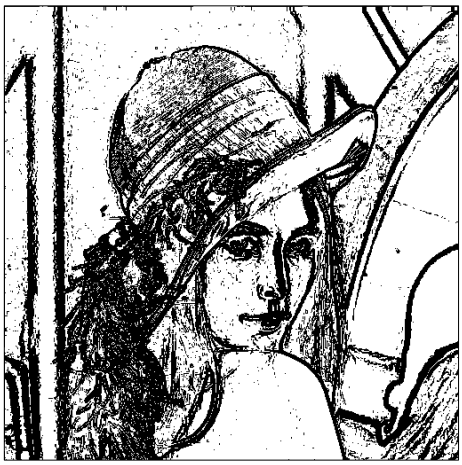
f_2

-1	-√2	-1
1	√2	1

2. Gradient Magnitude

$gradient\ magnitude = \sqrt{{f_1}^2 + {f_2}^2}$

3. The result (threshold=30 for example) is saved as `frei_and_chen.png` as shown on the right.



Kirsch's Compass Operator

1. Masks

k_1

-3	-3	5
-3		5
-3	-3	5

k_2

-3	-3	-3
-3		5
-3	5	5

k_3

-3	-3	-3
-3		-3
5	5	5

k_4

-3	-3	-3
5		-3
5	5	-3

k_5

5	-3	-3
5		-3
5	-3	-3

k_6

5	5	-3
5		-3
-3	-3	-3

k_7

5	5	5
-3		-3
-3	-3	-3

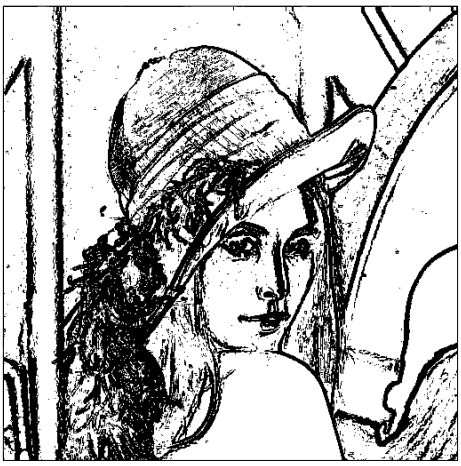
k_8

-3	5	5
-3		5
-3	-3	-3

2. Gradient Magnitude

$gradient\ magnitude = max\ k_n \quad (n = 1, \dots, 8)$

3. The result (threshold=135 for example) is saved as `kirsch.png` as shown below.



Robinson's Compass Operator

1. Masks

r_1

-1		1
-2		2
-1		1

r_2

	1	2
-1		1
-2	-1	

r_3

1	2	1
-1	-2	-1

r_4

2	1	
1		-1
	-1	-2

r_5

1		-1
2		-2
1		-1

r_6

	-2	-1
1		-1
2	1	

r_7

-1	-2	-1
1	2	1

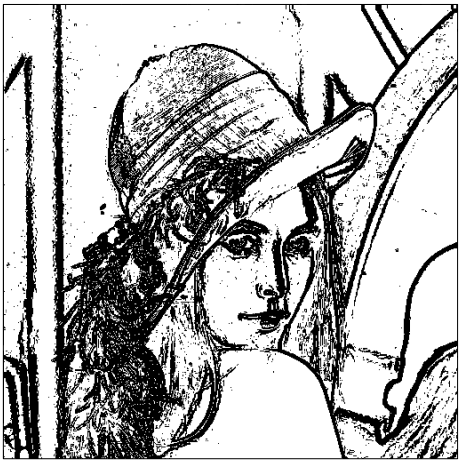
r_8

-2	-1	
-1		1
	1	2

2. Gradient Magnitude

$gradient\ magnitude = max\ r_n \quad (n = 1, \dots, 8)$

3. The result (threshold=43 for example) is saved as `robinson.png` as shown below.



Nevatia-Babu 5x5 Operator

1. Masks

N_1

100	100	100	100	100
100	100	100	100	100
0	0	0	0	0
-100	-100	-100	-100	-100
-100	-100	-100	-100	-100

N_2

100	100	100	100	100
100	100	100	78	-32
100	92	0	-92	-100
32	-78	-100	-100	-100
-100	-100	-100	-100	-100

N_3

100	100	100	100	100
-32	78	100	100	100
-100	-92	0	92	100
-100	-100	-100	-78	32
-100	-100	-100	-100	-100

N_4

-100	-100	-100	-100	-100
-100	-100	-100	-100	-100
0	0	0	0	0
100	100	100	100	100
100	100	100	100	100

N_5

100	100	100	32	-100
100	100	92	-78	-100
100	100	0	-100	-100
100	78	-92	-100	-100
100	-32	-100	-100	-100

N_6

-100	32	100	100	100
-100	-78	92	100	100
-100	-100	0	100	100
-100	-100	-92	78	100
-100	-100	-100	-32	100

2. Gradient Magnitude

$$gradient\ magnitude = max\ N_n \quad (n = 1, \dots, 6)$$

3. The result (threshold=12500 for example) is saved as `nevatia_babu.png` as shown below.