Computer and Robot Vision I

Chapter 7

Binary Machine Vision:

Conditioning And Labeling_

HW9 and HW10 講解(已更新圖片)

Presented by: 傅楸善 & 蔡婷瑀
E-mail: r08945026@ntu.edu.tw
手機: 0987233087
授課教授：傅楸善 博士

DC & CV Lab. CSIE NTU

Project due ~~Nov. 26~~ Dec. 3

Write the following programs

1. Generate additive white Gaussian noise
2. Generate salt-and-pepper noise
3. Run box filter *(3x3, 5x5)* on all noisy images
4. Run median filter *(3x3, 5x5)* on all noisy images
5. Run opening followed by closing and closing followed by opening.

Project due ~~Nov. 26~~ Dec. 3

Generate additive white Gaussian noise:

$$I(nim, i, j) = I(im, i, j) + amplitude * N(0, 1)$$

N(0, 1): Gaussian random variable with zero mean and st. dev. 1

amplitude determines signal-to-noise ratio, try 10 and 30

Generate salt-and-pepper noise:

$$I(nim, i, j) = 0 \text{ if } uniform(0, 1) < threshold$$

$$I(nim, i, j) = 255 \text{ if } uniform(0, 1) > 1 - threshold$$

$$I(nim, i, j) = I(im, i, j) \text{ otherwise}$$

uniform(0, 1): random variable uniformly distributed over [0, 1]

threshold determines signal-to-noise ratio, try both 0.05 and 0.10

Project Assignment Related

```python
from PIL import Image
import random
import numpy as np
```

```python
def GetGaussianNoise_Image_10(original_Image):
    gaussianNoise_Image_10 = original_Image.copy()
    for c in range(original_Image.size[0]):
        for r in range(original_Image.size[1]):
            noisePixel = int(original_Image.getpixel((c, r)) + 10 * random.gauss(0, 1))
            if noisePixel > 255:
                noisePixel = 255
            gaussianNoise_Image_10.putpixel((c, r), noisePixel)
    return gaussianNoise_Image_10
```

Generate additive white Gaussian noise with *amplitude = 10*

```python
def GetSaltAndPepper_Image_05(original_Image):
    SaltAndPepper_Image_05 = original_Image.copy()
    for c in range(original_Image.size[0]):
        for r in range(original_Image.size[1]):
            random_Value = random.uniform(0, 1)
            if (random_Value <= 0.05):
                SaltAndPepper_Image_05.putpixel((c, r), 0)
            elif (random_Value >= 1-0.05):
                SaltAndPepper_Image_05.putpixel((c, r), 255)
            else:
                SaltAndPepper_Image_05.putpixel((c, r), original_Image.getpixel((c, r)))
    return SaltAndPepper_Image_05
```

Generate salt-and-pepper noise with *threshold = 0.05*

169	153	145	137	112	98
130	95	120	130	115	128
124	162	45	87	75	101
124	177	176	136	150	137
141	38	54	155	132	57
87	156	161	180	99	79

Generate additive white Gaussian noise

173	147	135	145	100	105
133	100	106	143	130	132
132	165	46	96	60	132
125	169	163	144	144	143
130	36	34	144	140	47
91	146	169	178	94	66

Run box filter (*3x3*) on image.

138	132	129	126	125	116
141	126	120	106	115	109
137	126	125	114	124	123
126	111	110	107	116	111
116	118	131	134	122	105
100	101	117	126	111	86

I(0,0) = (173+147+133+100)/4=138 (向下取整數)

I(0,1) = (173+147+135+133+100+106)/6=132

I(1,0) = (173+147+133+100+132+165)/6=141

I(2,2) = (100+106+143+165+46+96+169+163+144)/9=125

169	153	145	137	112	98
130	95	120	130	115	128
124	162	45	87	75	101
124	177	176	136	150	137
141	38	54	155	132	57
87	156	161	180	99	79

Generate additive white Gaussian noise

157	154	136	138	139	88
160	84	113	119	100	127
123	176	46	93	75	85
123	169	172	130	149	154
138	35	70	148	116	71
77	153	165	187	90	87

Run median filter (*3x3*) on image.

157	154	136	136	127	127
157	136	119	113	100	100
160	123	119	113	119	127
138	123	130	116	116	116
138	138	153	148	130	116
138	138	153	148	116	90

O(0,0) = median(157, 154, 160, 84)=157 (向上取整數)

O(0,1) = median(157, 154, 136, 160, 84, 113)=154

O(1,0) = median(157, 154, 160, 84, 123, 176)=157

O(2,2) = median(84, 113, 119, 176, 46, 93, 169, 172, 130)=119

Project due Dec. ~~3~~ 10

Write programs to generate the following gradient magnitude images and choose proper thresholds to get the binary edge images:

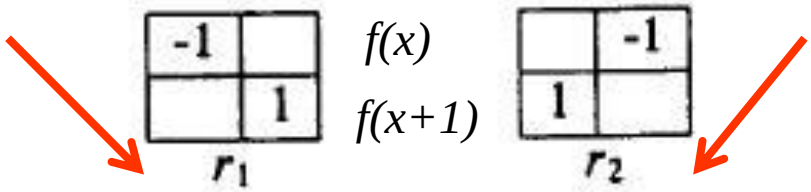
1. Roberts operator (threshold: 30)
2. Prewitt edge detector (threshold: 24)
3. Sobel edge detector (threshold: 38)
4. Frei and Chen gradient operator (threshold: 30)
5. Kirsch compass operator (threshold: 135)
6. Robinson compass operator (threshold: 43)
7. Nevatia-Babu 5×5 operator (threshold: 12500)

Roberts operator (ppt p.80)
Threshold=30

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	122	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	255	255	0	0	0	0	0
255	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	255	0	0	0	0	0	0
255	0	0	0	0	0	0	0
0	255	0	0	0	0	0	0
0	0	255	255	0	0	255	255

$f'(x) \approx f(x+1) - f(x)$



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

169	146
104	104

I(0, 0) =
r_1: 104-169=-65
r_2: 104-146=-42
gradient magnitude: $\sqrt{(-65)^2+(-42)^2}=77$
77 >= 30 (*threshold*)
O(0, 0) = 0 (black)

Prewitt operator (ppt p.82)

Threshold=24

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	122	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$f'(x) \approx f(x+1) - f(x-1)$

-1	-1	-1
1	1	1

p_1

$f(x-1)$

$f(x+1)$

-1		1
-1		1
-1		1

p_2

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

I(0, 0) =
p_1: (104+104+104)- (169+169+146)=-172
p_2: (146+146+104)-(169+169+104)=-46
gradient magnitude: 178 >= 24(*threshold*)
O(0, 0) = 0 (black)

169	169	146
169	169	146
104	104	104

I(0, 1) =
p1: (104+104+97)- (169+146+153)=-163
p2: (153+153+97)-(169+169+104)=-39
gradient magnitude: 167 >= 24 → O(0, 1) = 0 (black)

169	146	153
169	146	153
104	104	97

I(1, 0) =
p1: (130+130+120)- (169+169+146)=-104
p2: (146+104+120)-(169+104+130)=-33
gradient magnitude: 109 >= 24 → O(1, 0) = 0 (black)

169	169	146
104	104	104
130	130	120

Sobel operator (ppt p.84)

Threshold=38

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	120	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	255	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	255	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$f'(x) \approx f(x+1) - f(x-1)$

-1	-2	-1
1	2	1

s_1

$f(x-1)$

$f(x+1)$

-1		1
-2		2
-1		1

s_2

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

I(0, 0) =
s_1: (104+2×104+104)- (169+2×169+146)=-237
s_2: (146+2×146+104)-(169+2×169+104)=-69
gradient magnitude: 246 >= 38(*threshold*)
O(0, 0) = 0 (black)

169	169	146
169	169	146
104	104	104

I(0, 1) =
s_1: (104+2×104+97)- (169+2×146+153)=-205
s_2: (153+2×153+97)-(169+2×169+104)=-55
gradient magnitude: 212>= 38 → O(0, 1) = 0 (black)

169	146	153
169	146	153
104	104	97

I(1, 4) =
s_1 : (120+2×130+212)- (145+2×137+151)=22
s_2: (151+2×40+212)-(145+2×100+120)=--22
gradient magnitude: 31 < 38 → O(1, 4) = 255 (white)

145	137	151
100	115	40
120	130	212

Frei and Chen gradient operator (ppt p.87) Threshold=30

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	120	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	255	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

f(x-1)

f(x+1)

-1	$-\sqrt{2}$	-1
1	$\sqrt{2}$	1

f_1

-1		1
$-\sqrt{2}$		$\sqrt{2}$
-1		1

f_2

gradient magnitude: $\sqrt{f_1^2 + f_2^2}$ $\quad f'(x) \approx f(x+1) - f(x-1)$

I(0, 0) =
f_1: (104+$\sqrt{2}$×104+104)- (169+$\sqrt{2}$×169+146)=-198.92
f_2: (146+$\sqrt{2}$×146+104)-(169+$\sqrt{2}$×169+104)=-55.53
gradient magnitude: 206 >= 30(*threshold*)
O(0, 0) = 0 (black)

169	169	146
169	169	146
104	104	104

I(0, 1) =
f_1: (104+$\sqrt{2}$×104+97)- (169+$\sqrt{2}$×146+153)=-180.40
f_2: (153+$\sqrt{2}$×153+97)-(169+$\sqrt{2}$×169+104)=-45.63
gradient magnitude: 186 >= 30 → O(0, 1) = 0 (black)

169	146	153
169	146	153
104	104	97

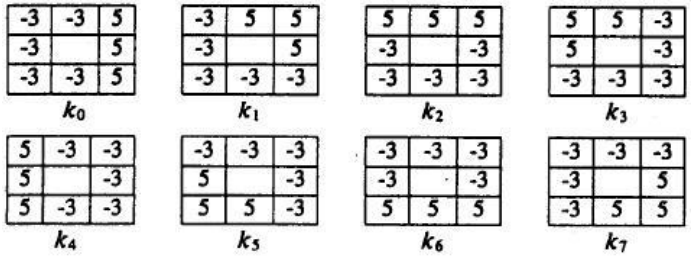
I(1, 4) =
f_1: (120+$\sqrt{2}$×130+212)- (145+$\sqrt{2}$×137+151)=26.10
f_2: (151+$\sqrt{2}$×40+212)-(145+$\sqrt{2}$×100+120)=13.15
gradient magnitude: 29 < 30 → O(1, 4) = 255 (white)

145	137	151
100	115	40
120	130	212

Kirsch compass operator (ppt p.91)
Threshold=135

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	120	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



gradient magnitude: $\max_{n,n=0,\ldots,7} k_n$

169	169	146
169	169	146
104	104	104

I(0, 0) =

k_0: -3x(169+169+169+104+104)+5x(146+146+104) = -165

k_1: -3x(169+169+104+104+104)+5x(169+146+146) = 355

k_2: -3x(169+104+104+104+146)+5x(169+169+146) = 539

k_3: -3x(146+146+104+104+104)+5x(169+169+169) = 723

k_4: -3x(169+146+146+104+104)+5x(169+169+104) = 203

k_5: -3x(169+169+146+146+104)+5x(169+104+104) = -317

k_6: -3x(169+169+169+146+146)+5x(104+104+104) = -837

k_7: -3x(104+169+169+169+146)+5x(146+104+104) = -501

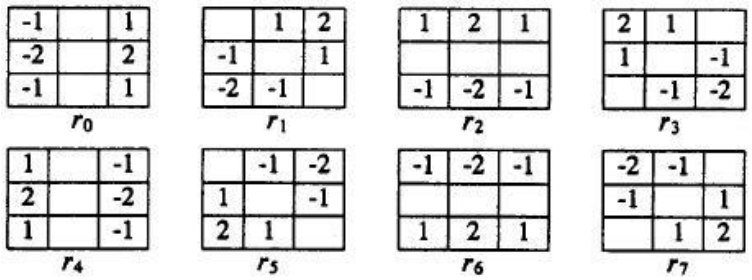
gradient magnitude: k_3=723 >= 135 (*threshold*)

O(0, 0) = 0 (black)

Robinson compass operator (ppt p.93)
Threshold=43

169	146	153	145	137	151	112	76
104	104	97	100	115	40	42	63
130	120	95	120	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



gradient magnitude: $\max_{n,n=0,\ldots,7} r_n$

169	169	146
169	169	146
104	104	104

I(0, 0) =
r_0: (146+2×146+104) - (169+2×169+104) = -69
r_1: (169+2×146+146) - (169+2×104+104) = 126
r_2: (169+2×169+146) - (104+2×104+104) = 237
r_3: (169+2×169+169) - (146+2×104+104) = 218
r_4: (169+2×169+104) - (146+2×146+104) = 69
r_5: (169+2×104+104) - (169+2×146+146) = -126
r_6: (104+2×104+104) - (169+2×169+146) = -237
r_7: (146+2×104+104) - (169+2×169+169) = -218

gradient magnitude: r_2=237 >= 43 (*threshold*)
O(0, 0) = 0 (black)

Nevatia-Babu 5×5 operator (1/2) (ppt p.95) Threshold=12500

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	120	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	255
255	255	255	0	0	0	255	255
0	255	255	0	0	0	0	0
255	0	0	0	0	0	0	255
0	0	0	0	255	255	255	0
0	0	0	255	255	255	0	0
0	0	0	0	0	255	0	0

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

0°

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

30°

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

60°

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

-90°

-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

-60°

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

-30°

gradient magnitude: $\max_{n,n=0,\ldots,5} N_n$

169	169	169	146	153
169	169	169	146	153
169	169	169	146	153
104	104	104	104	97
130	130	130	120	95

I(0, 0) =
N_0 = 49400, N_1 = 45724, N_2 = 29816,
N_3 = -16900, N_4 = 878, N_5 = 34750

gradient magnitude: N_0=49400 >= 12500 (*threshold*)
O(0, 0) = 0 (black)

Nevatia-Babu 5×5 operator (2/2) (ppt p.95) Threshold=12500

169	146	153	145	137	151	112	98
104	104	97	100	115	40	42	63
130	120	95	120	130	212	115	128
124	157	162	45	87	77	75	101
124	201	177	176	136	113	150	137
162	155	193	46	52	87	126	203
141	149	38	54	155	145	132	57
87	64	156	161	180	210	99	79

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	255
255	255	255	0	0	0	255	255
0	255	255	0	0	0	0	0
255	0	0	0	0	0	0	255
0	0	0	0	255	255	255	0
0	0	0	255	255	255	0	0
0	0	0	0	0	255	0	0

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

0°

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

30°

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

60°

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

-90°

-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

-60°

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

-30°

gradient magnitude: $\max_{n,n=0,\ldots,5} N_n$

169	146	153	145	137
104	104	97	100	115
130	120	95	120	130
124	157	162	45	87
124	201	177	176	136

I(2, 2) =
N_0 = -11900, N_1 = -9458, N_2 = 1774,
N_3 = -18800, N_4 = -29842, N_5 = -15442

gradient magnitude: N_2=1774 < 12500 (*threshold*)
O(2, 2) = 255 (white)

Project due Dec. ~~3~~ 10

Roberts operator with *threshold = 12*

Project due Dec. ~~3~~ 10

Roberts operator with *threshold = 30*

Project due Dec. ~~3~~ 10

Prewitt edge detector with *threshold = 24*

Project due Dec. ~~3~~ 10

Sobel edge detector with *threshold = 38*

Project due Dec. ~~3~~ 10

Frei and Chen gradient operator with *threshold = 30*

Project due Dec. ~~3~~ 10

Kirsch compass operator with *threshold = 135*

Project due Dec. ~~3~~ 10

Robinson compass operator with *threshold = 43*

Project due Dec. ~~3~~ 10

Nevatia-Babu 5×5 operator *threshold = 12500*

Project due Dec. ~~10~~ 17

Write the following programs to detect edge:

Zero-crossing on the following four types of images to get edge images (choose proper threshold t=1), p. 349

1. Laplacian(2張圖) (threshold = 15)
2. Minimum-variance Laplacian (threshold = 30)
3. Laplacian of Gaussian (threshold = 3000)
4. Difference of Gaussian, (use tk to generate D.O.G.)

   DoG (inhibitory σ = 1, excitatory σ = 3, kernel size=11)

   (threshold = 1)

Laplacian

(threshold = 15)

	1	
1	-4	1
	1	

(threshold = 15)

$\frac{1}{3}$

1	1	1
1	-8	1
1	1	1

minimum-variance Laplacian

(threshold = 20)

$\frac{1}{3}$

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

Laplacian of Gaussian

(threshold = 3000)

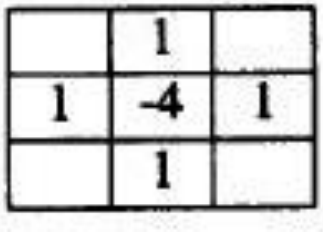
0	0	0	-1	-1	-2	-1	-1	0	0	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-2	-9	-23	-1	103	178	103	-1	-23	-9	-2
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
0	-2	-7	-15	-22	-23	-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

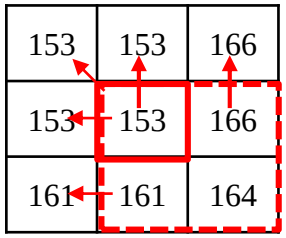
Difference of Gaussian (threshold = 1)
(inhibitory σ = 3, excitatory σ = 1, kernel size=11)

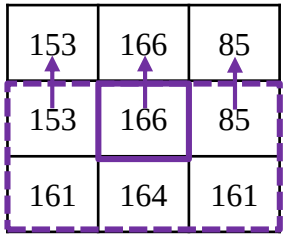
-1	-3	-4	-6	-7	-8	-7	-6	-4	-3	-1
-3	-5	-8	-11	-13	-13	-13	-11	-8	-5	-3
-4	-8	-12	-16	-17	-17	-17	-16	-12	-8	-4
-6	-11	-16	-16	0	15	0	-16	-16	-11	-6
-7	-13	-17	0	85	160	85	0	-17	-13	-7
-8	-13	-17	15	160	283	160	15	-17	-13	-8
-7	-13	-17	0	85	160	85	0	-17	-13	-7
-6	-11	-16	-16	0	15	0	-16	-16	-11	-6
-4	-8	-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

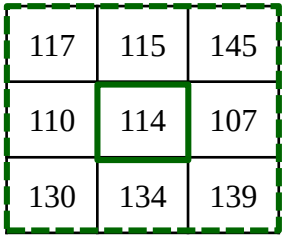
Laplacian mask 1 (1/4) (ppt p.104) Threshold=15

153	166	85	90	96	102	101	98	73	64	55	40	172	163	155
161	164	161	166	171	173	174	176	180	178	183	174	180	174	180
103	105	97	97	97	103	101	98	111	111	104	48	54	49	57
126	123	117	115	145	86	113	146	96	73	110	98	43	53	95
135	129	110	114	107	113	114	66	84	34	120	130	73	51	79
134	128	130	134	139	129	94	39	45	197	48	43	41	57	46
123	167	142	183	168	110	45	141	61	85	81	61	57	118	127
130	131	170	194	170	69	171	100	173	150	136	128	134	142	137
133	127	211	201	172	127	181	115	141	131	148	150	151	147	140
108	110	95	227	172	192	203	169	163	149	42	51	189	171	159
163	149	146	150	203	127	69	37	49	64	85	78	98	215	202
155	152	144	209	182	205	66	46	45	147	156	112	146	125	101
209	143	149	46	52	55	134	159	157	150	143	136	122	148	109
123	93	54	55	144	156	164	155	145	188	215	207	209	98	91
96	51	67	163	162	154	157	145	191	210	213	48	91	85	89









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

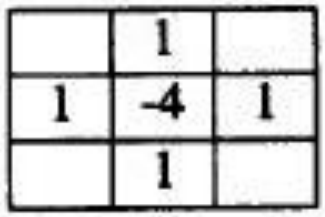
I(0, 0): gradient magnitude: 1×153+1×166+1×161+1×153-4×153 = 21 >= 15 (*threshold*)
L(2, 2) = 1

I(0, 1): gradient magnitude: 1×166+1×85+1×164+1×153-4×166 = -96 <= -15 (*-threshold*)
L(0, 1) = -1

I(4, 3): gradient magnitude: 1×115+1×107+1×134+1×110-4×114 = 10 < 15 (*threshold*) and > -15 (*-threshold*)
L(4, 3) = 0

Laplacian mask 1 (2/4)
(ppt p.104) Threshold=15

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



	1	
1	-4	1
	1	

Input image

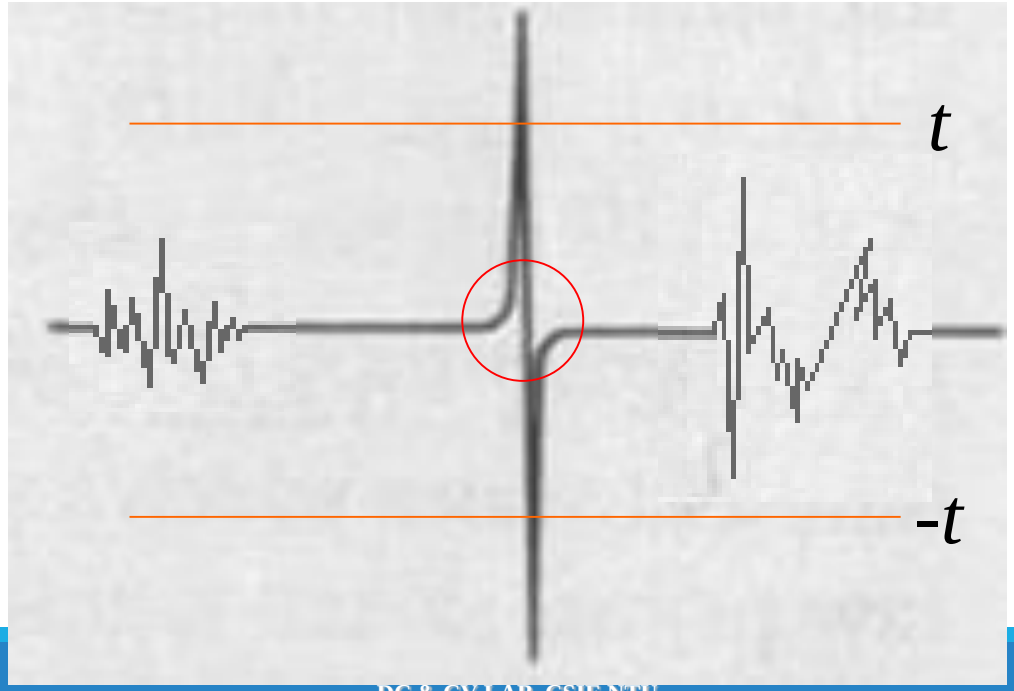
153	166	85	90	96	102	101	98	73	64	55	40	172	163	155
161	164	161	166	171	173	174	176	180	178	183	174	180	174	180
103	105	97	97	97	103	101	98	111	111	104	48	54	49	57
126	123	117	115	145	86	113	146	96	73	110	98	43	53	95
135	129	110	114	107	113	114	66	84	34	120	130	73	51	79
134	128	130	134	139	129	94	39	45	197	48	43	41	57	46
123	167	142	183	168	110	45	141	61	85	81	61	57	118	127
130	131	170	194	170	69	171	100	173	150	136	128	134	142	137
133	127	211	201	172	127	181	115	141	131	148	150	151	147	140
108	110	95	227	172	192	203	169	163	149	42	51	189	171	159
163	149	146	150	203	127	69	37	49	64	85	78	98	215	202
155	152	144	209	182	205	66	46	45	147	156	112	146	125	101
209	143	149	46	52	55	134	159	157	150	143	136	122	148	109
123	93	54	55	144	156	164	155	145	188	215	207	209	98	91
96	51	67	163	162	154	157	145	191	210	213	48	91	85	89

Laplacian output

1	-1	1	1	1	1	1	1	1	1	1	1	-1	0	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-1	-1	-1	0	-1	1	0	-1	1	1	-1	-1	1	1	-1
-1	-1	1	0	1	-1	-1	1	-1	1	-1	-1	-1	1	-1
-1	1	0	1	-1	-1	-1	1	1	-1	1	1	1	1	1
1	-1	1	-1	-1	-1	1	-1	1	1	0	1	1	-1	-1
0	1	0	-1	-1	1	-1	1	-1	-1	-1	-1	-1	-1	0
-1	1	-1	0	-1	1	-1	1	1	1	-1	-1	1	1	1
1	1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	1	1
-1	-1	-1	1	-1	1	1	1	1	1	0	1	1	-1	-1
1	-1	1	-1	-1	-1	1	1	1	-1	-1	1	-1	1	1
-1	1	-1	1	1	1	-1	-1	-1	1	1	1	1	-1	0
1	0	1	1	-1	-1	-1	0	1	-1	-1	-1	-1	1	1
-1	1	1	-1	-1	0	0	1	-1	-1	-1	1	1	1	0

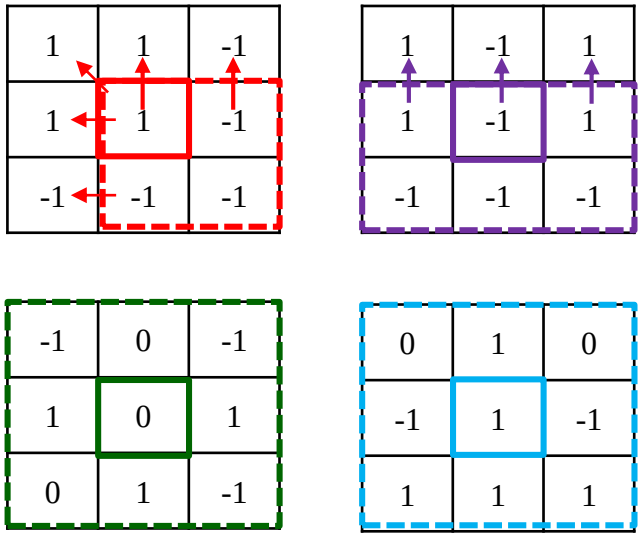
Zero-Crossing Edge Detectors

A pixel is declared to have a zero crossing if it is less than $-t$ and one of its eight neighbors is greater than t, or if it is greater than t and one of its eight neighbors is less than $-t$ for some fixed threshold t



Zero-crossing on Laplacian mask 1 (3/4) (ppt p.104, 119)

1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-1	-1	-1	0	-1	1	0	-1	1	1	-1	-1	1	1	-1
-1	-1	1	0	1	-1	-1	1	-1	1	-1	-1	-1	1	-1
-1	1	0	1	-1	-1	-1	1	1	-1	1	1	1	1	1
1	-1	1	-1	-1	-1	1	-1	1	1	0	1	1	-1	-1
0	1	0	-1	-1	1	-1	1	-1	-1	-1	-1	-1	-1	0
-1	1	-1	0	-1	1	-1	1	1	1	-1	-1	1	1	1
1	1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	1	1
-1	-1	-1	1	-1	1	1	1	1	1	0	1	1	-1	-1
1	-1	1	-1	-1	-1	1	1	1	-1	-1	1	-1	1	1
-1	1	-1	1	1	1	-1	-1	-1	1	1	1	1	-1	0
1	0	1	1	-1	-1	-1	0	1	-1	-1	-1	-1	1	1
-1	1	1	-1	-1	0	0	1	-1	-1	-1	1	1	1	0



L(0, 0): 1, and zero crossing in L(-1, 1), L(0, 1), L(1, -1), L(1, 0), L(1, 1) = -1
O(0, 0) = 0 (black)

L(0, 1): -1 → O(0, 1) : 255 (white)

L(4, 3): 0 → O(4, 3) = 255(white)

L(8, 1): 1, and zero crossing in L(8, 0), L(8, 2) = -1
O(8, 1) = 0 (black)

做完Laplacian運算子後，
如果mask pixel >= t (t=1)，並且它的8個鄰域像素之一 <= –t (-1)，
則代表該pixel過零(zero crossing)，為邊(黑色)。

Zero-crossing on Laplacian mask 1 (4/4) (ppt p.104, 119)

Laplacian output

1	-1	1	1	1	1	1	1	1	1	1	1	-1	0	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-1	-1	-1	0	-1	1	0	-1	1	1	-1	-1	1	1	-1
-1	-1	1	0	1	-1	-1	1	-1	1	-1	-1	-1	1	-1
-1	1	0	1	-1	-1	-1	1	1	-1	1	1	1	1	1
1	-1	1	-1	-1	-1	1	-1	1	1	0	1	1	-1	-1
0	1	0	-1	-1	1	-1	1	-1	-1	-1	-1	-1	-1	0
-1	1	-1	0	-1	1	-1	1	1	1	-1	-1	1	1	1
1	1	1	-1	1	-1	-1	-1	-1	-1	1	1	-1	1	1
-1	-1	-1	1	-1	1	1	1	1	1	0	1	1	-1	-1
1	-1	1	-1	-1	-1	1	1	1	-1	-1	1	-1	1	1
-1	1	-1	1	1	1	-1	-1	-1	1	1	1	1	-1	0
1	0	1	1	-1	-1	-1	0	1	-1	-1	-1	-1	1	1
-1	1	1	-1	-1	0	0	1	-1	-1	-1	1	1	1	0

Zero-crossing on Laplacian output (Output image)

0	255	0	0	0	0	0	0	0	0	0	0	255	255	0
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
255	255	255	255	255	0	255	255	0	0	255	255	0	0	255
255	255	0	255	0	255	255	0	255	0	255	255	255	0	255
255	0	255	0	255	255	255	0	0	255	0	0	0	0	0
0	255	0	255	255	255	0	255	0	0	255	0	0	255	255
255	0	255	255	255	0	255	0	255	255	255	255	255	255	255
255	0	255	255	255	0	255	0	0	0	255	255	0	0	0
0	0	0	255	0	255	255	255	255	255	0	0	255	0	0
255	255	255	0	255	0	0	0	0	0	255	0	0	255	255
0	255	0	255	255	255	0	0	0	255	255	0	255	0	0
255	0	255	0	0	0	255	255	255	0	0	0	0	255	255
0	255	0	0	255	255	255	255	0	255	255	255	255	0	0
255	0	0	255	255	255	255	0	255	255	255	0	0	0	255