

컴퓨터 비전

[HW08]

학번	201203393
분반	00
이름	김현겸
과제번호	08

제출일 : 2016년 5월 15일 일요일

1. (MATLAB) Exercise 7, 10, 13

Exercise 7

```
function exercise7()

im = imread('circles.tif');
[x,y] = meshgrid(1:256, 1:256);
im2 = double(im).*((x+y)/2+64) + x + y;
im3 = uint8(256*mat2gray(im2));

thresh = @(param) im2bw(param, graythresh(param));

out1 = blkproc(im3, [32, 32], thresh);
out2 = blkproc(im3, [51, 51], thresh);
out3 = blkproc(im3, [64, 64], thresh);
out4 = blkproc(im3, [85, 85], thresh);
out5 = blkproc(im3, [128, 128], thresh);

figure,
subplot(2,3,1), imshow(out1), title('32x32')
subplot(2,3,2), imshow(out2), title('51x51')
subplot(2,3,3), imshow(out3), title('64x64')
subplot(2,3,4), imshow(out4), title('85x85')
subplot(2,3,5), imshow(out5), title('128x128')
```

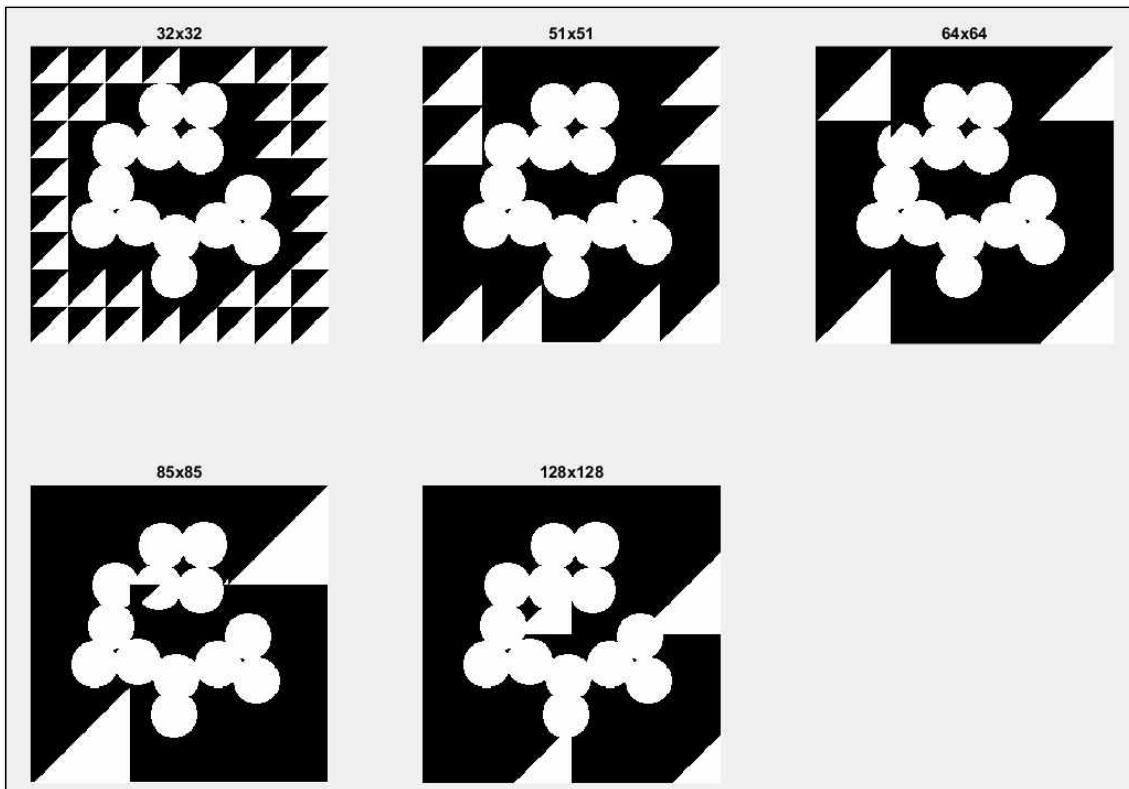
주어진 코드대로 circle에 대한 매트릭스를 생성하고 이에 threshold를 적용하여 원과 배경을 구분해 내는 블록을 찾는 문제이다. 책에서 나온 예시에서는 [256, 64]를 사용하여 x축만 4개의 블록으로 나누었지만 좀 더 세분화해서 구분해 내기 위해서는 좀 더 세분화 된 블록을 적용해야 한다.

우선 inline 함수는 더 이상 쓰이지 않으므로 anonymous 함수를 사용하여 im2bw와 graythresh 함수를 한 번에 적용하도록 만든다.

```
thresh = @(param) im2bw(param, graythresh(param))
```

그리고 이렇게 만든 함수에 blkproc을 사용하여 이미지 블록마다 thresh 함수를 적용해본다.

이렇게 해서 얻어 낼 수 있는 결과물은 다음과 같다.



순서대로 32x32, 51x51, 64x64, 85x85, 128x128 블록으로 쪼개서 threshold를 적용한 결과이다. 블록의 크기가 좀 더 세분화 될수록 circle들을 구분하는 것이 가장 좋을 수 있다. 또한 너무 크기가 작을 시에는 주변 이미지(배경들)의 색이 너무 많이 바뀌어 배경이 격자무늬로 바뀌어버리므로 적당한 값으로 얻어 내는 것이 중요하다. 만약 이미지가 circle보다 복잡하고, 분리해낼 대상이 좀 더 크다면 32보다 세분화된 블록이 더 좋을 수 있지만, 위와 같은 결과를 보았을 때는 $256/8 = 32$ 인 32x32 블록이나 $256/5 = 51$ 인 51x51 크기의 블록으로 threshold를 적용하는 것이 더 좋은 결과를 낼 수 있다고 할 수 있다.

Exercise 10

```
function exercise10(im, title_name)

detect1 = edge(im, 'Roberts');
detect2 = edge(im, 'Prewitt');
detect3 = edge(im, 'Sobel');

l = fspecial('laplacian', 0);

detect4 = mat2gray(filter2(l, im));

detect5 = edge(im, 'zerocross', 1);

detect6 = edge(im, 'log');
detect7 = edge(im, 'Canny');

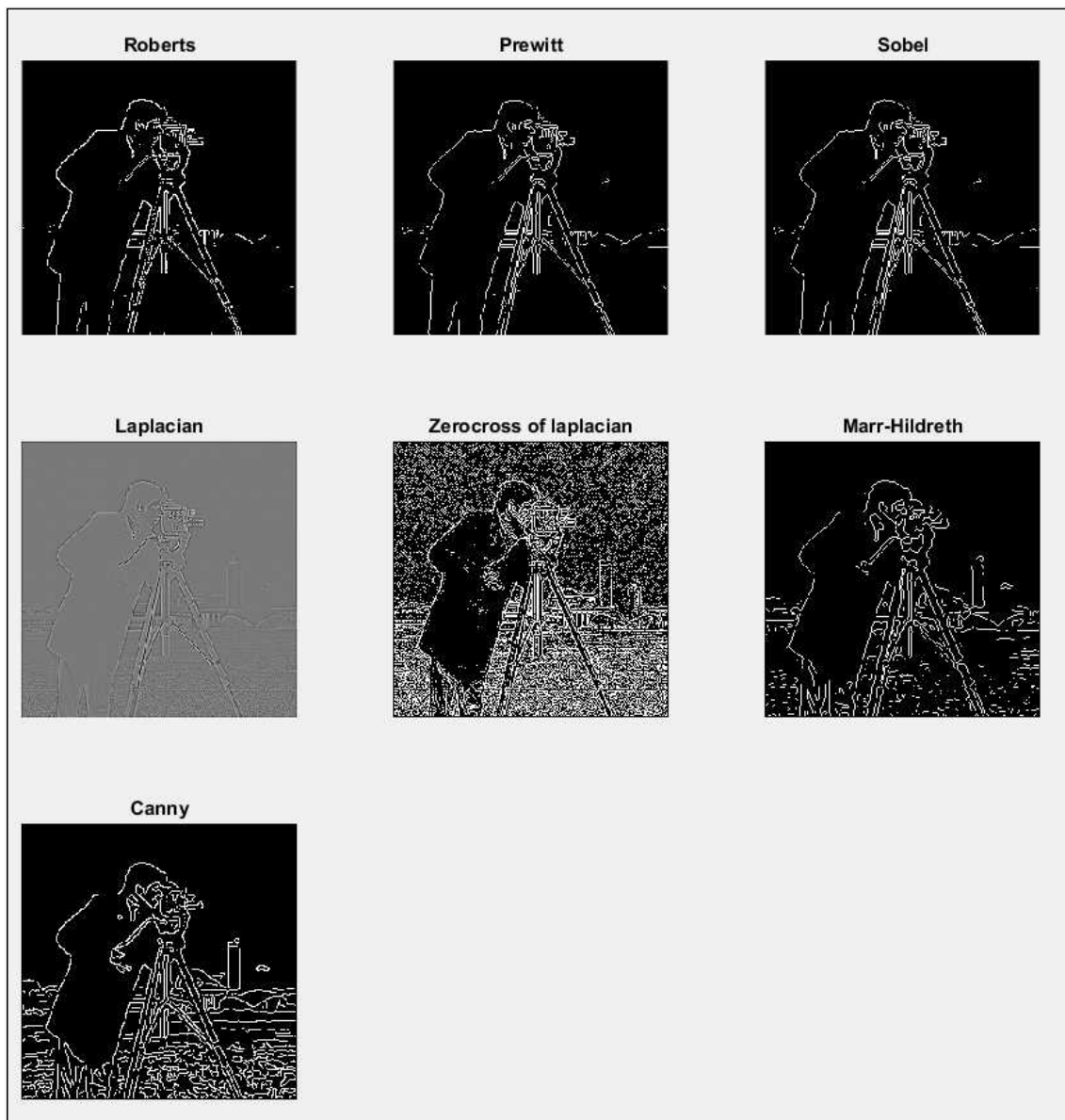
fh = figure,
subplot(3,3,1), imshow(detect1), title('Roberts')
subplot(3,3,2), imshow(detect2), title('Prewitt')
subplot(3,3,3), imshow(detect3), title('Sobel')
subplot(3,3,4), imshow(detect4), title('Laplacian')
subplot(3,3,5), imshow(detect5), title('Zerocross of laplacian')
subplot(3,3,6), imshow(detect6), title('Marr-Hildreth')
subplot(3,3,7), imshow(detect7), title('Canny');

set(fh, 'Name', title_name);
```

edge detection 알고리즘들을 cameraman.tif 이미지에 적용하여 보는 문제이다.

Roberts, Prewitt, Sobel, Laplacian, zerocross, Marr-Hildreth, Canny 알고리즘들을 edge 함수를 이용해서 적용해본다.

Laplacian의 경우, 이미지에 Laplacian 필터를 적용한 뒤, zerocross 기법을 이용하는 것이고, Marr-Hildreth의 경우에는 우선 이미지에 gaussian filter를 적용하여 smoothing 작업을 거친 뒤, Laplacian 필터를 적용하여 zerocross 기법으로 마무리 하는 방법이다.



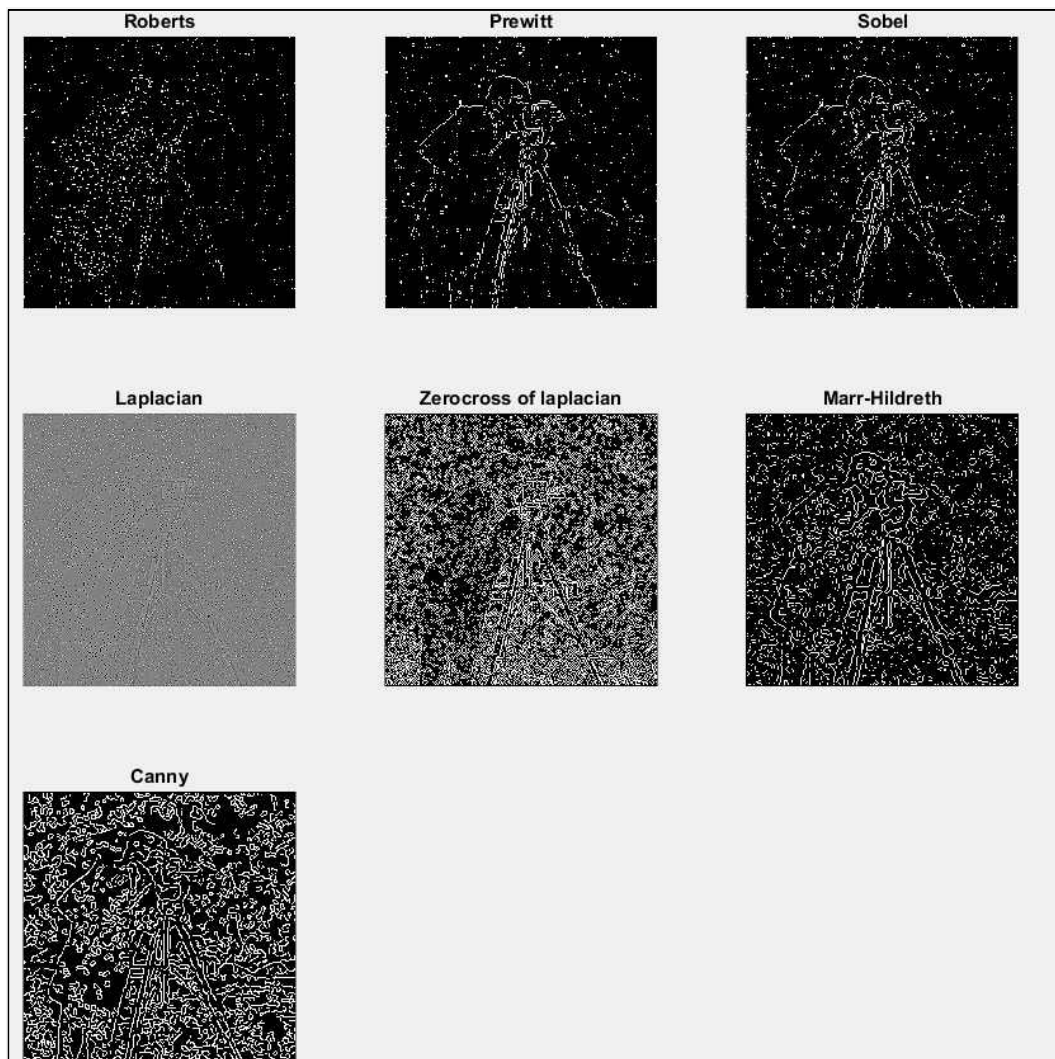
결과를 보면 Canny 알고리즘이 존재하는 모든 edge를 표시하면서 가장 좋은 성능을 내고 있음을 알 수 있다.

Exercise 13

```
function exercise13()  
  
    im = imread('cameraman.tif');  
    im1 = imnoise(im, 'salt & pepper', 0.1);  
    im2 = imnoise(im, 'gaussian', 0, 0.02);  
  
    exercise10(im, 'original image')  
    exercise10(im1, 'Salt & pepper noise')  
    exercise10(im2, 'gaussian noise')
```

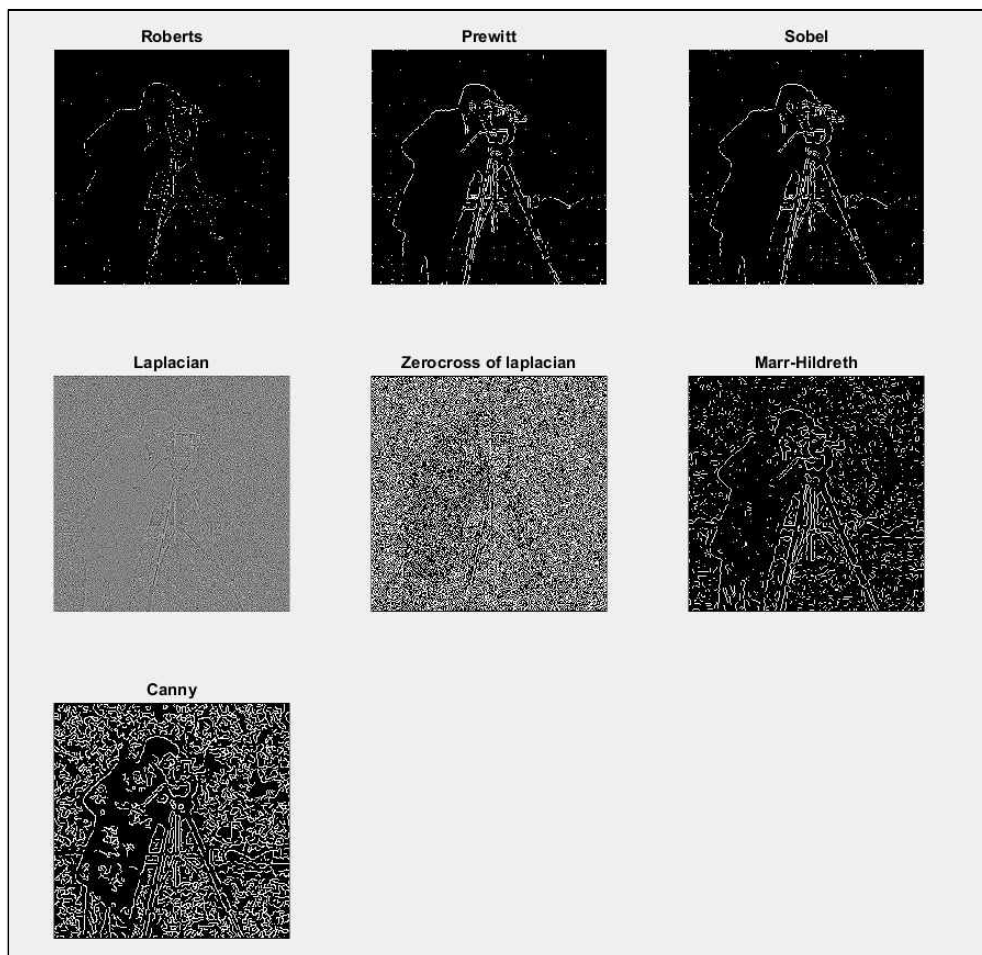
salt & pepper 와 gaussian noise를 이미지에 적용하여 이전에 해보았던 edge detection을 적용해본다.

Salt & Pepper Noise



salt & pepper noise에 대해서 각각의 알고리즘을 적용해보면 Canny 알고리즘은 비교적 지저분해 보이지만 경계들이 그려져 있긴 함을 알 수 있다. 반면, Sobel 알고리즘은 이전에는 그다지 경계를 많이 잡아내진 못했지만 noise가 적용된 이미지에 대해서도 경계가 잘 보이도록 구분해주고 있음을 알 수 있다. 이를 보면 오히려 Sobel 알고리즘이 더 나은 성능을 보여주고 있고, Laplacian filter가 제일 안좋은 성능을 보여주고 있음을 알 수 있다.

Gaussian Noise

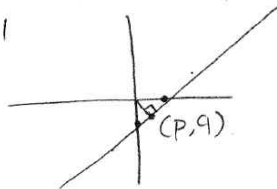


Gaussian noise에 대해 적용한 결과를 보면 Prewitt과 Sobel filter가 상당한 성능을 보여주고 있음을 알 수 있고, Zerocross of laplacian은 이미지가 구분이 되지 않을 정도로 엉망임을 알 수 있다.

각각의 알고리즘이 복잡도는 다르지만, 특정 이미지의 상태에 대해서만 좋은 결과를 보여줄 수 있는 비교이다. 만약 이미지에 noise가 없다면 Canny와 Marr-Hildreth 알고리즘이 제일 좋은 성능을 보여주겠지만, noise가 존재하는 이미지라면 오히려 Prewitt나 Sobel 알고리즘이 더 나은 결과를 보여줄 것이다.

Exercise 14

$$y = x - 1$$

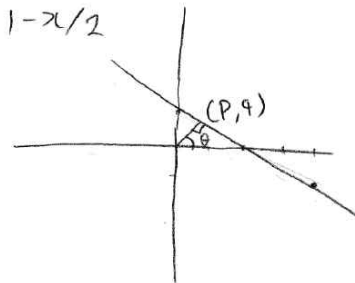


$$(P, q) = (r \cos \theta, r \sin \theta)$$

$$y \sin \theta + x \cos \theta = r$$

(x, y)	$-\frac{\sqrt{2}}{2}$ -45°	$\frac{\sqrt{2}}{2}$ 0°	$0, 1$ 45°	$\frac{\sqrt{2}}{2}, 1, 0$ 90°
$(-2, -3)$	10.7	-2	-3.5	-3
$(-1, -2)$	0.7	-1	-2.1	-2
$(0, -1)$	0.7	0	-0.7	-1
$(1, 0)$	0.7	1	0.7	0
$(2, 1)$	0.7	2	2.1	1
$(3, 2)$	0.7	3	3.5	2

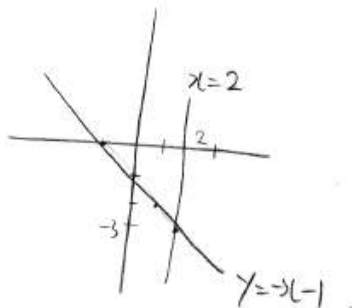
$$y = 1 - x/2$$



(x, y)	$x - y$ -45°	x 0°	$x + y$ 45°	y 90°
$(-2, 2)$	-2.8	-2	0	2
$(-1, \frac{3}{2})$	-1.75	-1	0.35	1.5
$(0, 1)$	-0.7	0	0.7	1
$(1, \frac{1}{2})$	-0.35	1	1.05	0.5
$(2, 0)$	1.4	2	1.4	0
$(4, -1)$	3.5	4	2.1	-1

Exercise 15

(x, y)	-45°	0°	45°	90°
$(-3, 2)$	-3.5	-3	-0.7	2
$(-1, 0)$	-0.7	-1	-0.7	0
$(0, -1)$	0.7	0	-0.7	-1
$(2, -3)$	3.5	2	-0.7	-3
$(2, -1)$	2.1	2	0.7	-1
$(2, 2)$	0	2	2.8	2



	-3.5	-3	-1	-0.7	0	0.7	2	2.1	2.8	3.5
-45°	1				1	1				
0°		1	1		1		1			1
45°				1		3				
90°		1	2	4		1	2		1	

$$\Rightarrow (r, \theta) = (2, 0^\circ)$$

$$(-0.7, 45^\circ)$$

$$\Rightarrow 0 + x = 2$$

$$0.7y + 0.7x = -0.7 \Rightarrow y = -x - 1$$

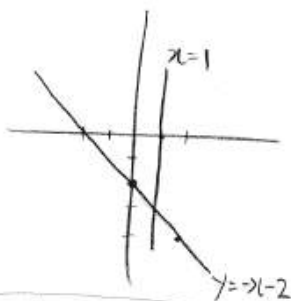


$$\therefore y = -x - 1, x = 2$$

Exercise 16

(x, y)	-45°	0°	45°	90°
$(-3, 1)$	-2.8	-3	-1.4	1
$(-2, 0)$	-1.4	-2	-1.4	0
$(-1, -1)$	0	-1	-1.4	-1
$(0, -1)$	0.7	0	-0.7	-1
$(0, 3)$	-2.1	0	2.1	3
$(1, -3)$	2.8	1	-1.4	-3
$(1, 0)$	0.7	1	0.7	0
$(1, 2)$	-0.7	1	2.1	2
$(1, 3)$	-1.4	1	2.8	3
$(2, 2)$	0	2	2.8	2
$(3, -1)$	2.8	3	1.4	-1
$(3, 1)$	1.4	3	2.8	1

	-45°	0°	45°	90°
-3		1		1
-2.8	1			
-2.1	1			
-2		1		
-1.4	2		4	
-1		1		3
-0.7	1		1	
0	2	2		2
0.7	2		1	
1		4		2
1.4	1		1	
2		1		2
2.1			2	
2.8	2		3	
3		2		
3.5				2



$$\Rightarrow (r, \theta) = (-1.4, 45^\circ)$$

$$(1, 0^\circ)$$

$$\Rightarrow 0.7x + 0.7y = -1.4 \Rightarrow x + y = -2$$

$$\leftarrow x = 1$$

$$y = -x - 2$$

$$\therefore y = -x - 2, x = 1$$

(x, y)	-45°	0°	45°	90°
$(-3, -2)$	-0.7	-3	-3.5	-2
$(-3, 2)$	-3.5	-3	-0.7	2
$(-2, 1)$	-2.1	-2	-0.7	1
$(-1, 0)$	-0.7	-1	-0.7	0
$(-1, 3)$	-2.8	-1	1.4	3
$(0, -3)$	2.1	0	-2.1	-3
$(0, 1)$	-0.7	0	0.7	1
$(1, -2)$	2.1	1	-0.7	-2
$(2, 0)$	1.4	2	1.4	0
$(2, 3)$	-0.7	2	3.5	3
$(3, 2)$	0.7	3	3.5	2

	-45°	0°	45°	90°
-3.5	1			
-3		2		
-2.8	1			1
-2.1	1			
-2		1		
-1.4			1	
-1		2		2
-0.7	4		4	
0		2		2
0.7	1		1	
1		1		2
1.4	1		2	
2		2		2
2.1	2			
2.8		1		
3			2	
3.5				

$$\Rightarrow (r, \theta) = (-0.7, 45^\circ)$$

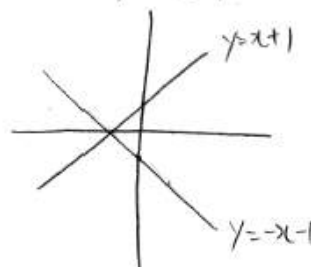
$$= (-0.7, 45^\circ)$$

$$\Rightarrow 0.7x - 0.7y = -0.7$$

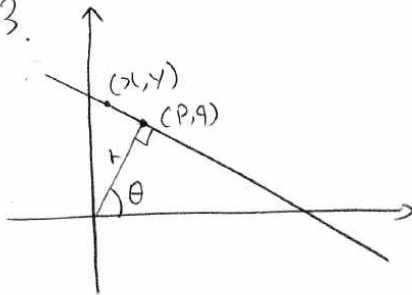
$$0.7x + 0.7y = -0.7$$

$$\Rightarrow y = x + 1$$

$$y = -x - 1$$



3.



(p, q) 는 $(r \cos \theta, r \sin \theta)$ 에 대응되고, 직선인 직교선의 기울기는 $\frac{\sin \theta}{\cos \theta}$ 이다
 직선의 기울기는 $\frac{y-q}{x-p} = \frac{y-r \sin \theta}{x-r \cos \theta}$. 직교선이 $\tan \theta$ 이므로 $\frac{1}{\tan \theta} = -\frac{\cos \theta}{\sin \theta}$ 이다.

이 둘을 항등식으로 구성하면 $\frac{y-r \sin \theta}{x-r \cos \theta} = -\frac{\cos \theta}{\sin \theta}$ 이고 분모를 정리하면
 $y \sin \theta - r \sin^2 \theta = -\cos \theta (x - r \cos \theta)$ 이다.

$$\Rightarrow y \sin \theta = -\cos \theta \cdot x + r \cos^2 \theta + r \sin^2 \theta$$

$$\Rightarrow y \sin \theta + x \cos \theta = r(\cos^2 \theta + \sin^2 \theta) = r.$$

$$\therefore y \sin \theta + x \cos \theta = r$$