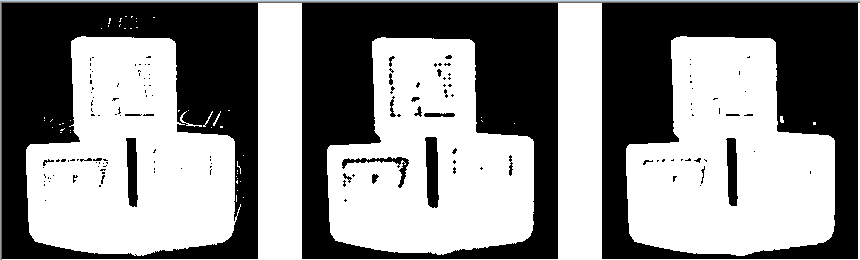
디지털 영상처리 연구실 연구보고서

김우헌

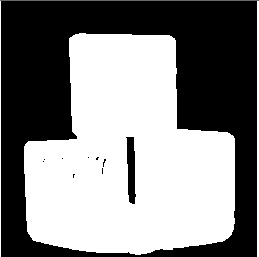
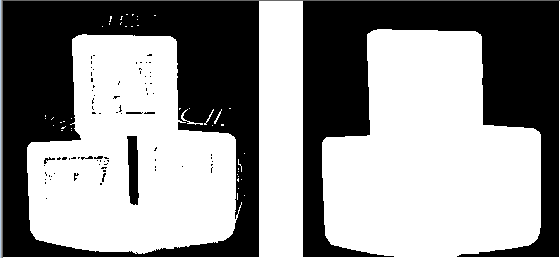
##모폴로지

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

침식과 팽창연산을 위한 마스크



->팽창 ->침식

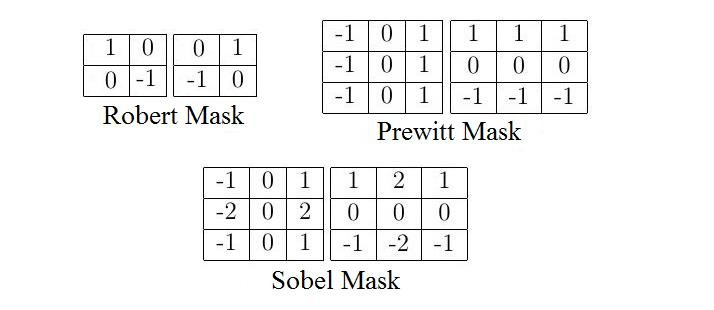
->침식연산 3회실시 후 모습 -> 침식연산 5회 실시 후 모습잡음이 완전히 사라진 모습

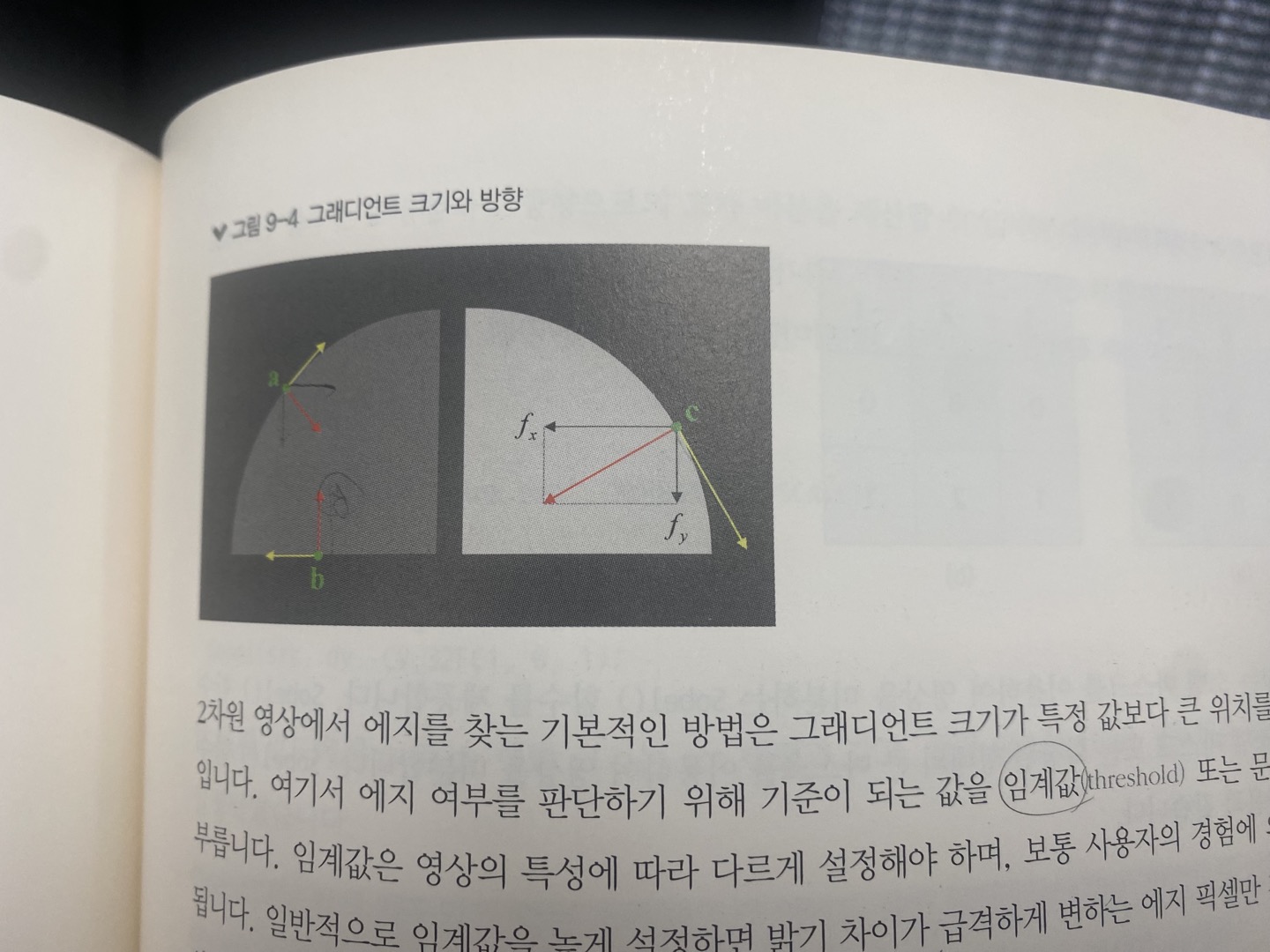
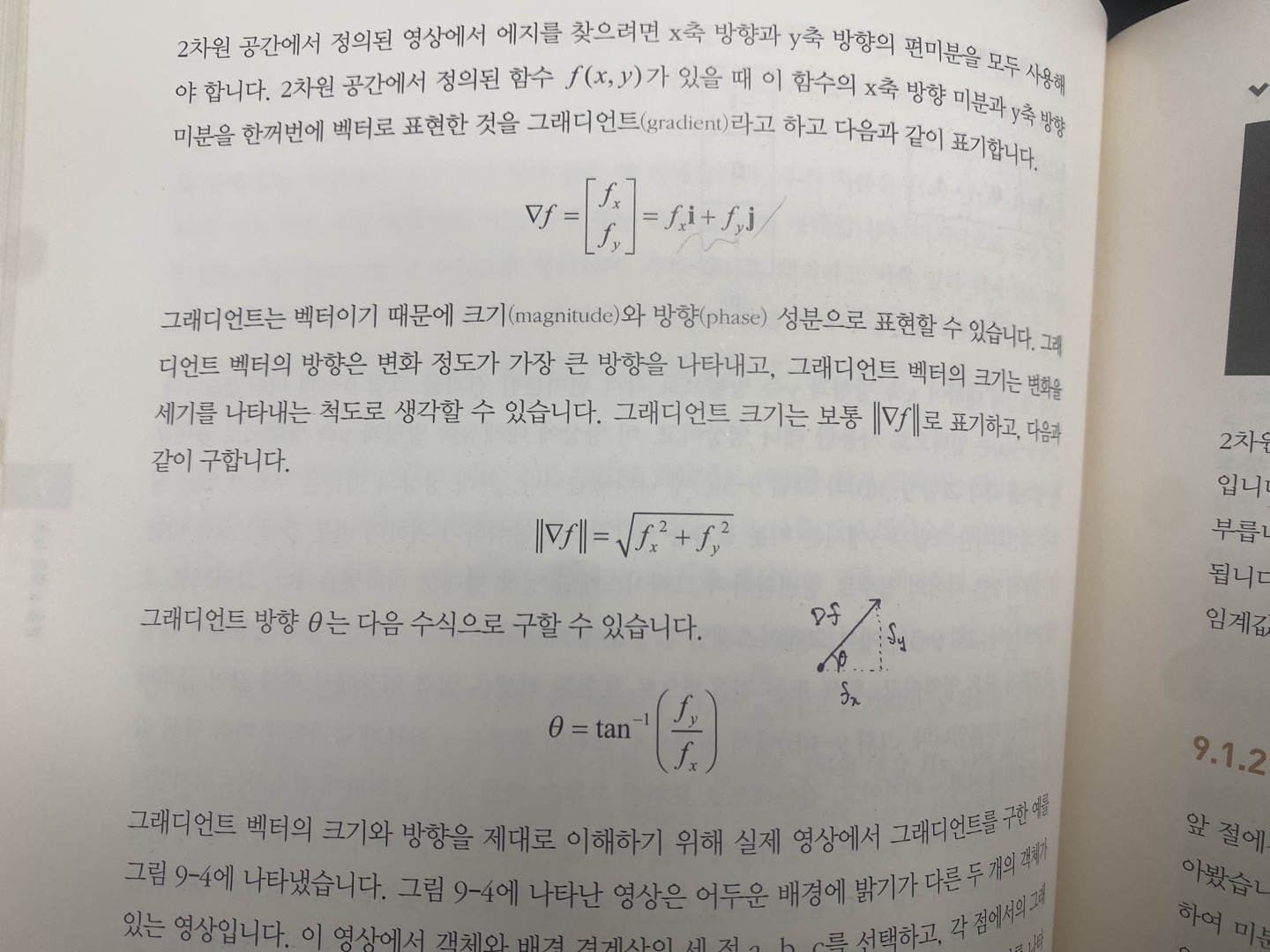
* 팽창연산후 침식연산 효과로 제거연산 역할 하는 것을 확인 하였습니다.

##canny edge 알고리즘

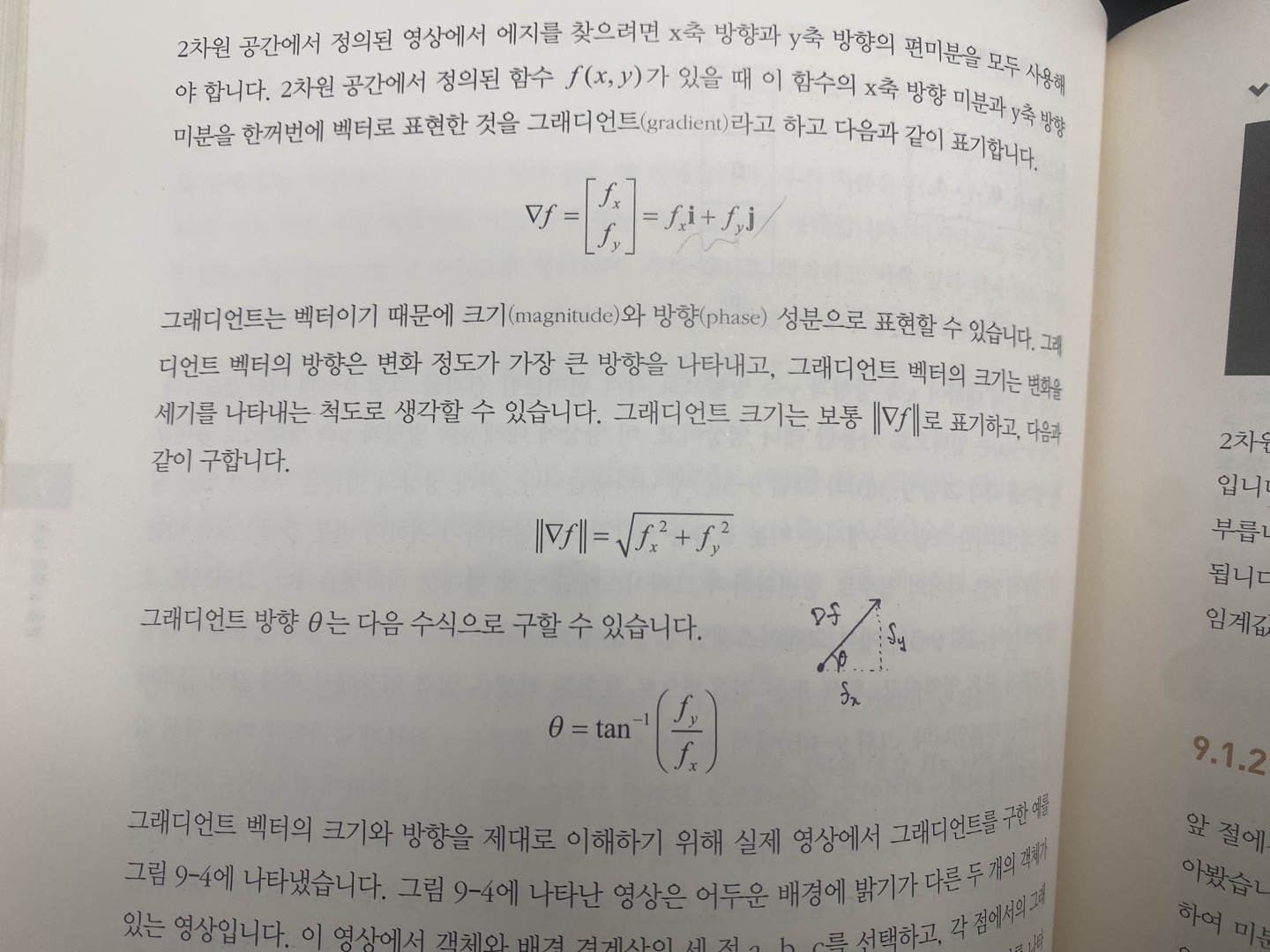
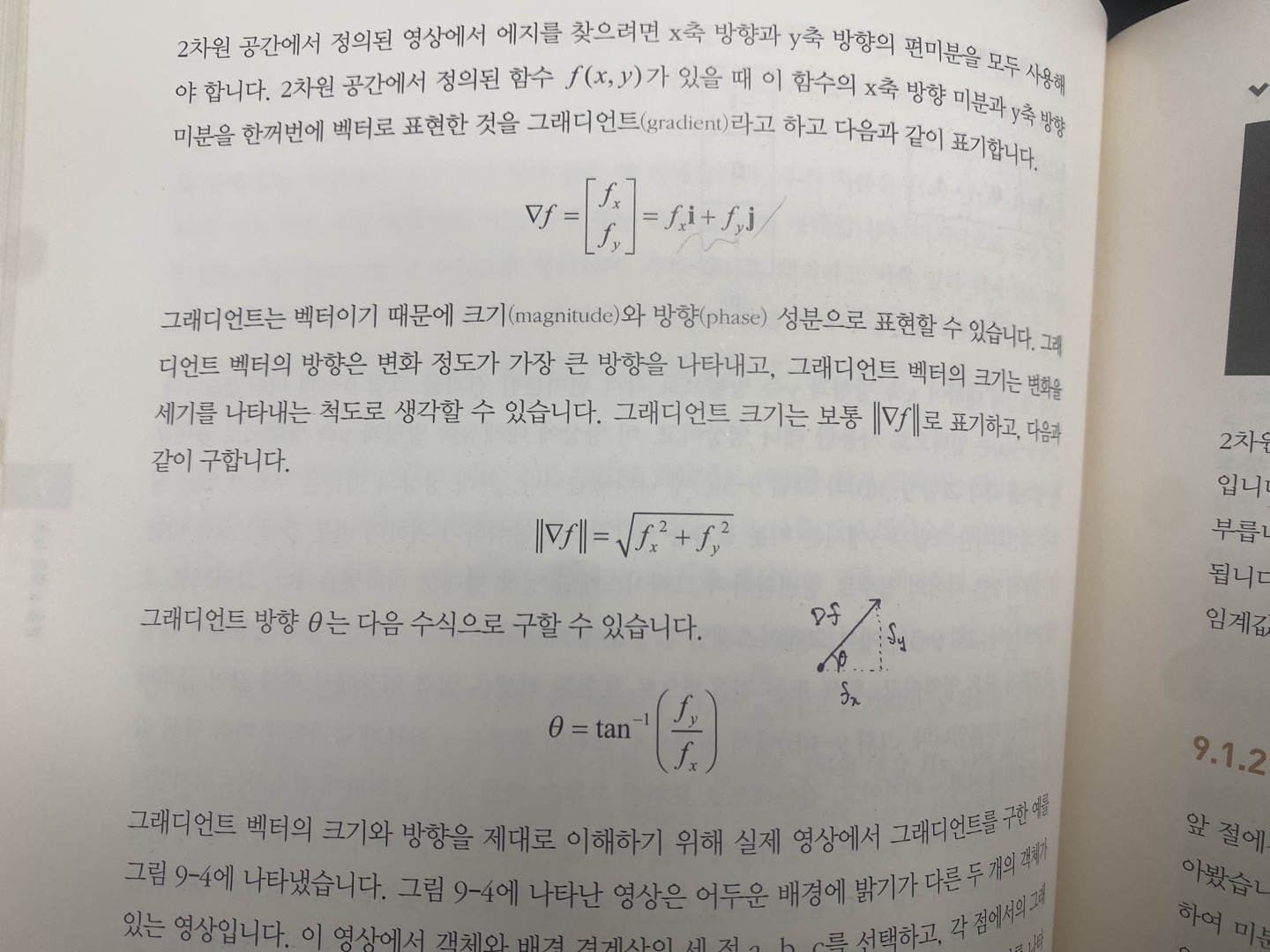
#그래디언트?

-> 프리윗필터,소벨필터는 x축,y축방향 으로 각각 편미분을 사용하는데 이두개를 하나의 벡터로 표현한 것이 그래디언트입니다.



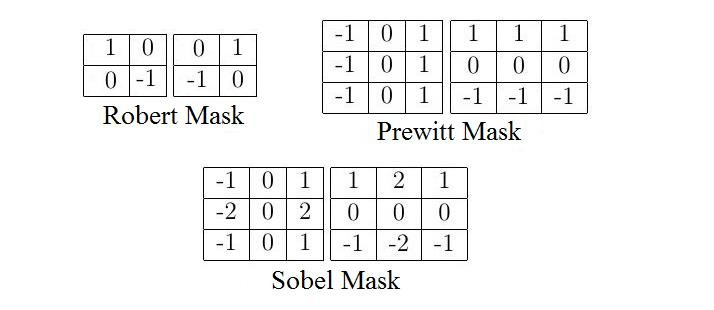
 

->이때의 그래디언트 크기와 각도는

* 그래디언트 벡터의 방향은 해당위치에서 밝기가 가장 밝아지는 방향
* 그래디언트 벡터의 수직인 방향=엣지의 방향

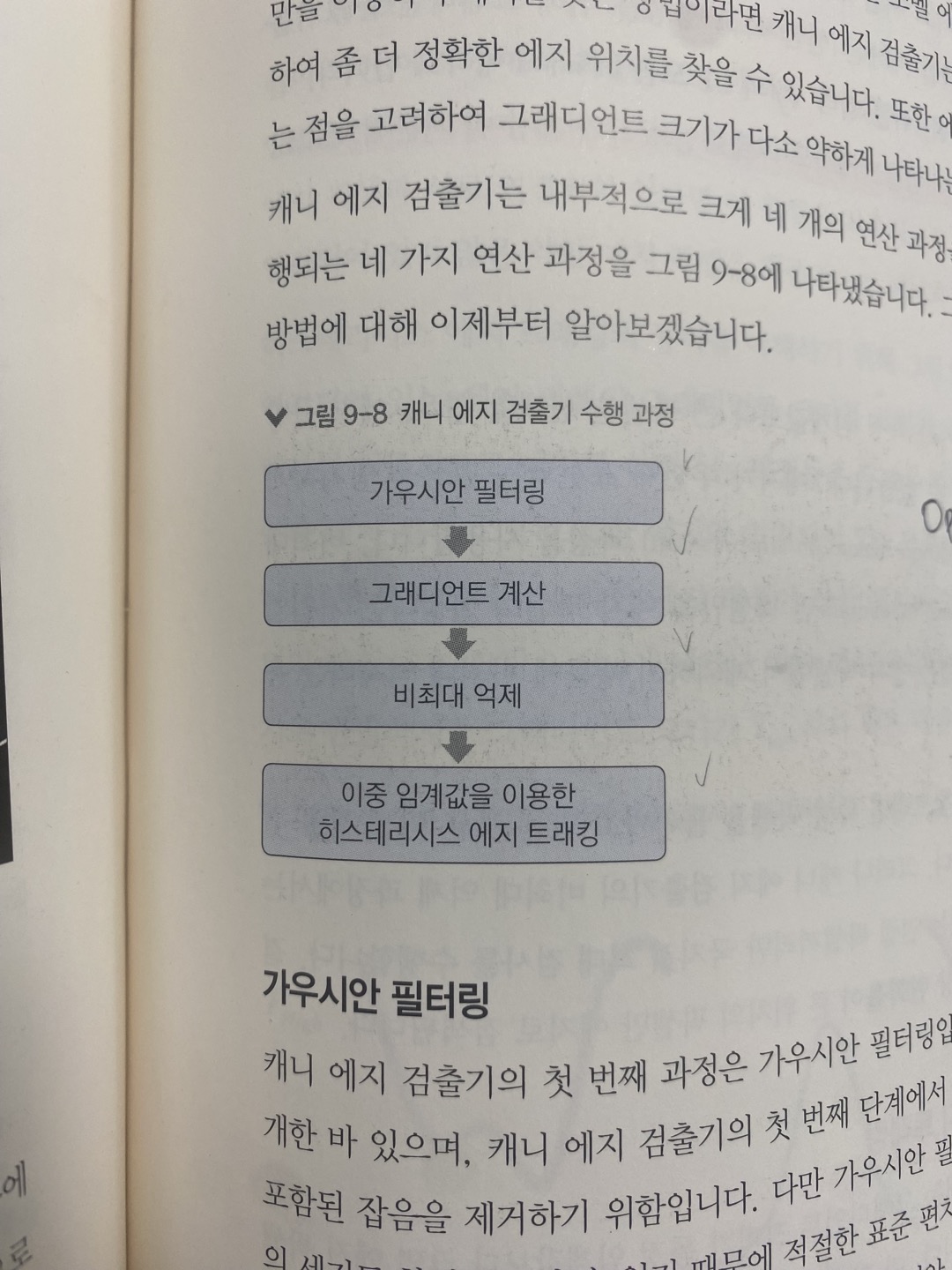
#소벨 필터





->edge가 두껍게 나타나고 정확하지 않음(그래디언트 크기만 이용)

#케니엣지 수행과정



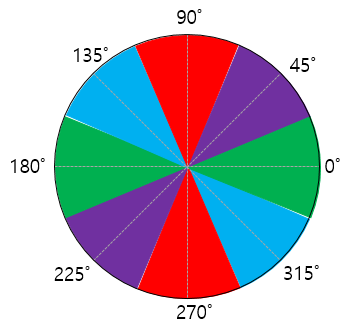
->이 과정을 통해 정확한 검출과 위치, 얇은 엣지를 추출 할 수 있는 케니 엣지 검출 알고리즘

1. 가우시안 필터링

* 잡음 제거 목적 -> 적절한 표준편차 이용하여 필터링

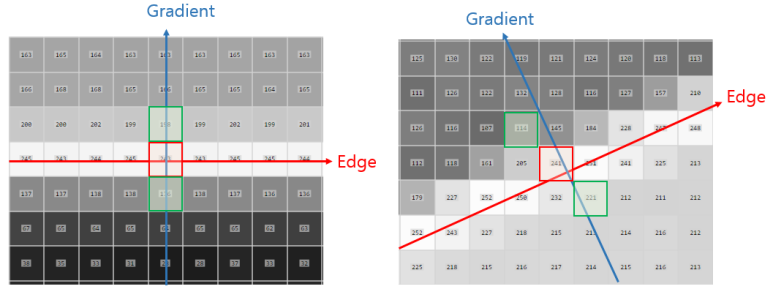
1. 그래디언트 계산

* 소벨 필터는 그래디언트의 크기로만 엣지를 탐색
* 케니 엣지 탐색은 소벨필터로 필터링 한 후 그래디언트 크기와 방향 모두 고려

-> 그래디언트 방향은 (0,45,90,135)로 그룹을 만들어 단순화

1. 비최대억제 (NMS)

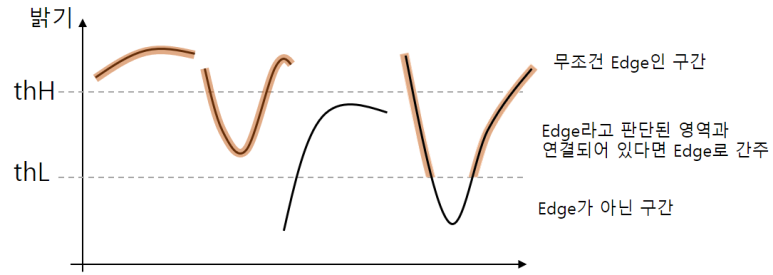
* EDGE의 두께를 얇게하기 위해서 에지가 아닌 픽셀을 제거하는 방법



* 현재 화소보다 그래디언트 크기가 크지 않으면 제거되는 방식

1. 이중 임계값을 통한 히스테리시스 에지 트래킹

* 실제 에지인지 아닌지 판단하는 단계 ->두개의 경계값으로 에지 판단



#C++을 사용한 CANNY EDGE

1. 가우시안 마스크



->가우시안 마스크를 적용하여 잡음제거

1. 그래디언트 계산

111

for (mr = 0; mr< 3; mr++)

for(mc=0;mc<3;mc++)

{

newvale1 += gx[mr][mc] \* m\_inimage[i + mr-1][j + mc-1];

newvale2 += gy[mr][mc] \* m\_inimage[i + mr-1][j + mc-1];

}

sobelsum[i \* width + j] = sqrt(newvale1 \*newvale1 + newvale2 \* newvale2);

* 소벨 필터 적용후 I,j위치에서 그래디언트 크기 (sobelsum) 구함

111

theta = atan2(newvale2, newvale1) \* 180 / 3.141592;

if ((theta > -22.5&& theta >22.5)||theta > 157.5 || theta < -157.5)

{

sobelang[where] = 0;

}

else if ((theta >= 22.5 && theta < 67.5) || (theta >= -157.5 && theta < -112.5))

{

sobelang[where] = 45;

}

111

else if ((theta >= 67.5 && theta <= 112.5) || (theta >= -112.5 && theta <= -67.5))

{

sobelang[where] = 90;

}

else

{

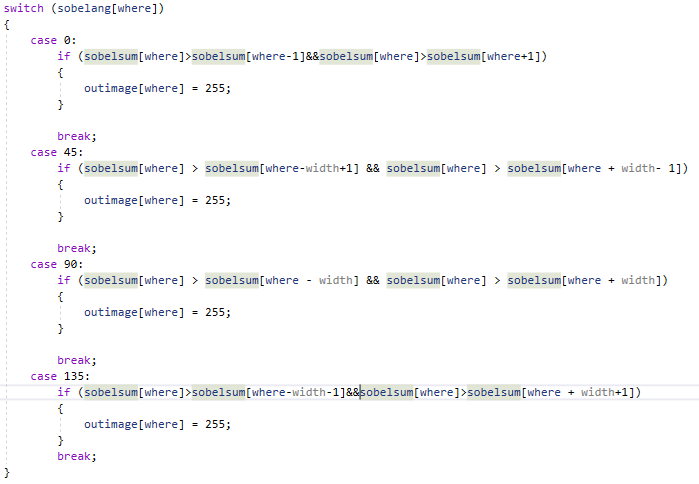
sobelang[where] = 135;

}

* 그래디언트 각도 theta 를 구한 후 그룹화하여 단순화하여 sobelang[where]에 저장

1. 비최대 억제

111



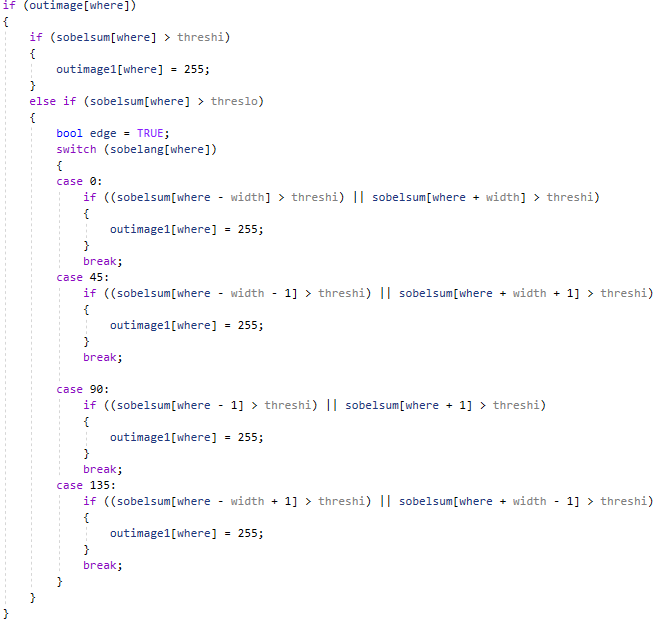
* 그룹별로 outimage에 단일 엣지를 저장합니다.



->비최대 억제를 통하여 얇아진 엣지 추출

1. 이중 임계값을 통한 히스테리시스 에지 트래킹

111



->비최대 억제를 통해 저장된값(outimage)이 255인 경우에 그래디언트 값이 thres hi보다 클경우 엣지

Thres hi보다 작고 thres lo보다 큰경우 엣지가 이어져 있으면 엣지라고 판단 !

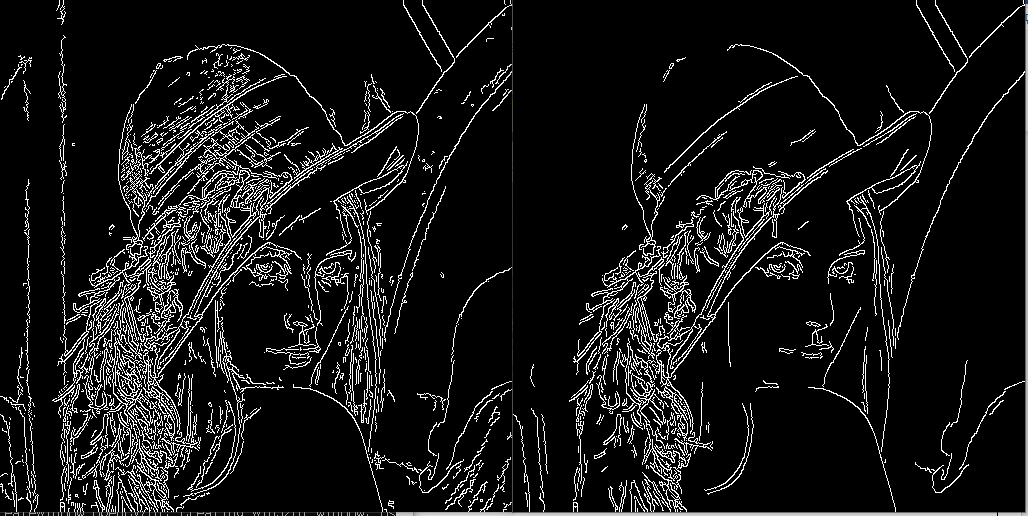


-> Thres hi =70, thres low =30

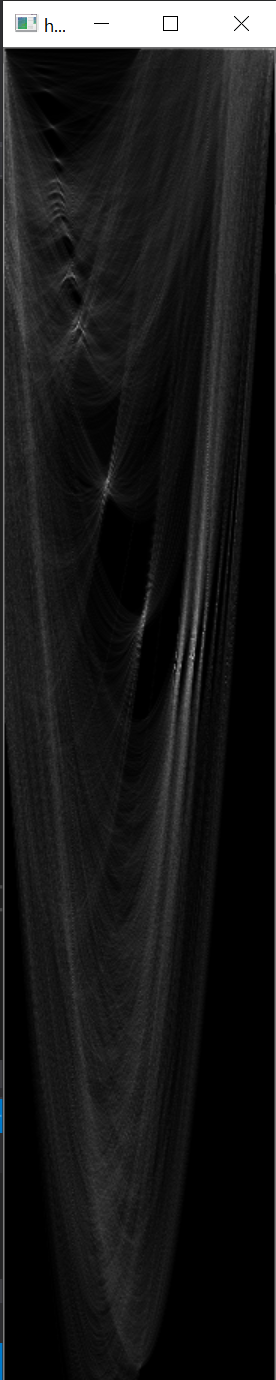
* Thres hi =90, thres low =30 **->** Thres hi =120, thres low =30

#open cv를 통한 canny edge 탐색



-> Thres hi =100, thres low =150 Thres hi =150, thres low =50

##허프 변환



for (d = 4; d <= 360; d++)

for ( k = 0; k < 360; k++)

{

for ( n = -3; n <=3; n++)

for (m = -3; m <= 3; m++)

{

if (k + n >= 0 && k + n < 360 && d + m >= 4 && d + m <= 360)

if (max < H[k + n][d + m])

{ max = H[k + n][d + m]; }

}

for (n = -3; n <= 3; n++)

for (m = -3; m <= 3; m++)

{

if (k + n >= 0 && k + n < 360 && d + m >= 4 && d + m <= 360)

if (max != H[k + n][d + m])

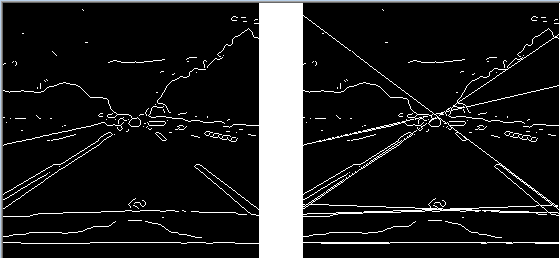
H[k + n][d + m]=0;

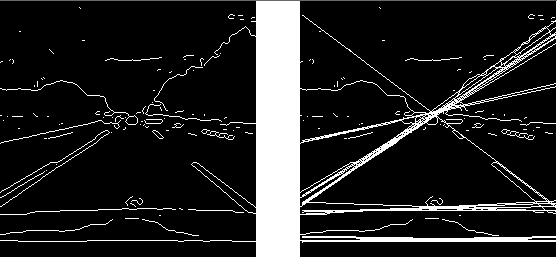
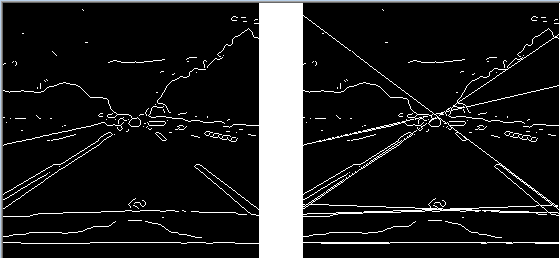
}

max = 0;

}

* 파라메타 공간 H[k][d] 에서 3x3공간내 최대값을 제외한 값들은 0으로 매핑(중복방지)

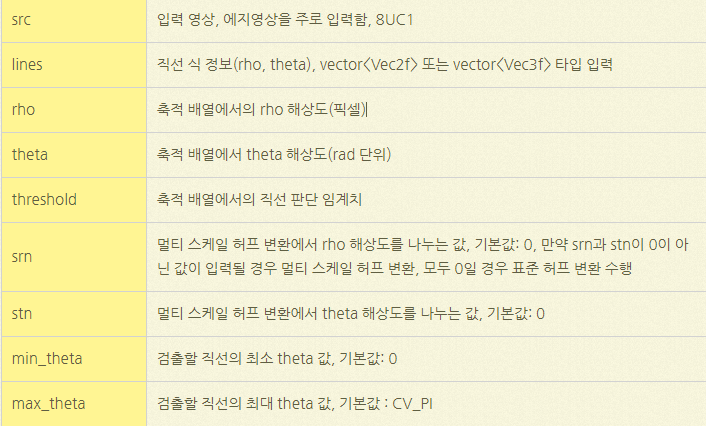
 >원본 이미지

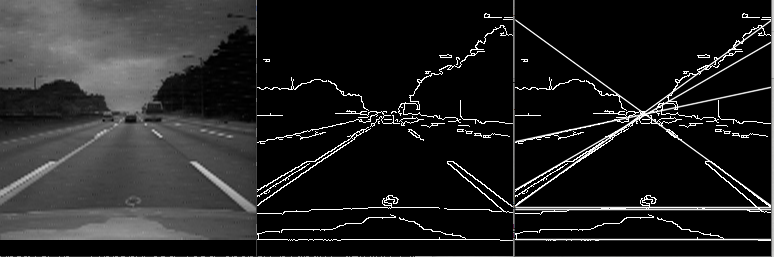
 

->Thres 60에 기존의 허프변환 ->파라메타 공간 수정후 thres60의 허프변환

#open cv를 이용한 허프변환

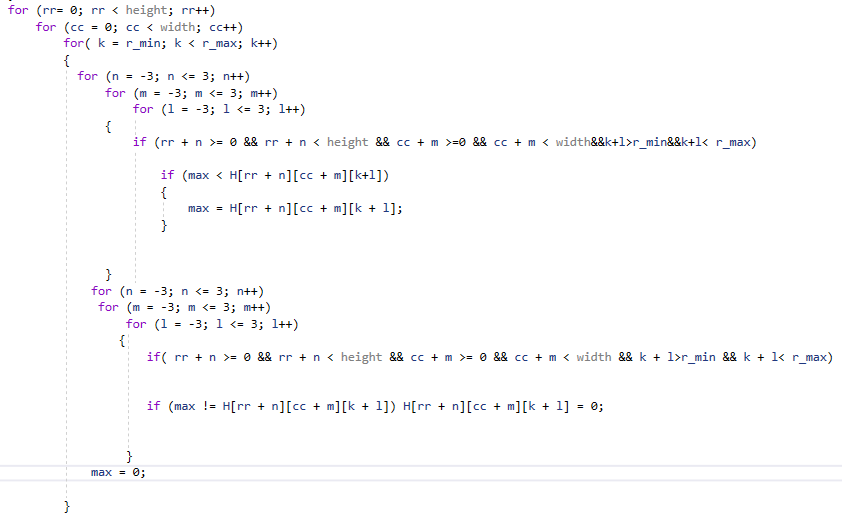




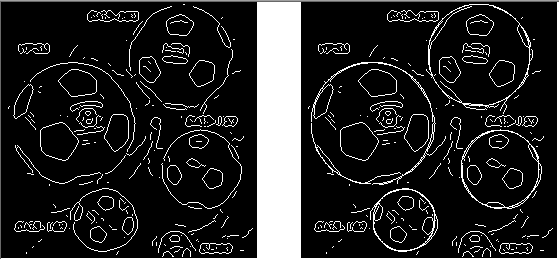


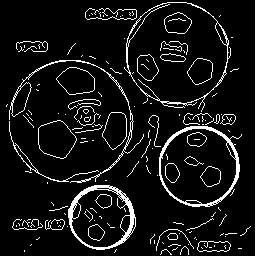
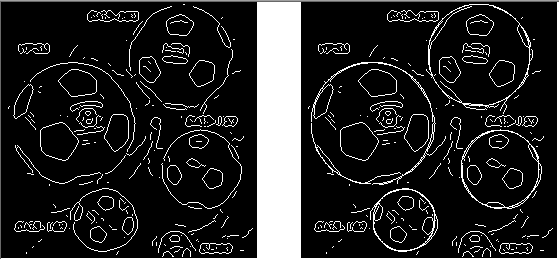
->HoughLines(edge, lines, 1, CV\_PI / 30, 70) -> 6도 간격으로설정하여 중복 최소화, thres=70

#원의 허프변환



->직선의 허프변환과 마찬가지로 파라메타 공간H[i][j][k]에서 최대값을 제외한 값들을 0으로 매핑!

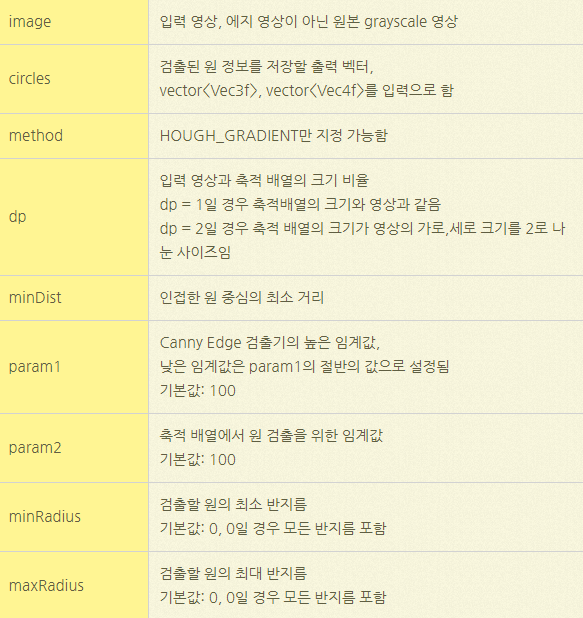
->원본 이미지

-> Thres 110에서의 기존 허프변환 ->파라메타 공간변화후 thres110에서의 허프변환

#OPEN CV에서의 허프변환





****

**-> dp1,mindist 50,param1 200,param2 40**