

Homework 2

2016-16908 심영인

1 Ans)

(10X8) array가 convolution 연산 이후 (8X8) array 결과가 나온 것으로 보아 (3X1) array 임을 알 수 있다. Convolution 연산의 정의를 생각하면 operator (?)는 아래의 array와 같다.

-1
2
-1

이 array는 second finite difference operator와 관계가 있다. 바로 second derivative의 discrete한 form이다. 이 convolution 연산은 horizontal edge를 detect해준다. 이 연산을 통해 나온 output에서 -1과 1 사이의 선이 바로 horizontal edge 이다.

2-1 Ans)

Image $I(x, y)$ 에서 edge(방향에 상관 없음)에 대응된다. 또한 완전히 uniform 한 image의 구역이라면 pixel의 value가 constant하므로 $h(x, y) = 0$ 인 구역과 대응된다.

2-2 Ans)

이 parameter에 따라 edge를 detect하는 threshold가 달라진다. 이 parameter에 따라 더 많은 혹은 더 적은 edge를 detect할 수 있다.

이 Value가 증가하면 $h(x, y) = 0$ 인 점은 적어진다. detect 되는 edge가 더 적어진다.

3-1 Ans)

(x, y) 을 (ρ, θ) parameter를 이용하여 sinusoid 로 나타내야 한다. Image의 임의의 점 (x, y) 을 생각해보자. 주어진 식 $\rho = x * \cos(\theta) + y * \sin(\theta)$ 에 (x, y) 와 원점과의 거리인 $\sqrt{x^2 + y^2}$ 으로 나누어 주면 $\rho = \sqrt{x^2 + y^2} * (\sin(\alpha) * \cos(\theta) + \cos(\alpha) * \sin(\theta)) = \sqrt{x^2 + y^2} * (\sin(\alpha + \theta))$ 로 나타낼 수 있다. 여기서 $\sin(\alpha) = x/\sqrt{x^2 + y^2}, \cos(\alpha) = y/\sqrt{x^2 + y^2}$ 이다. 즉 magnitude는 $\sqrt{x^2 + y^2}$ 이고 phase는 $\tan^{-1}(x/y)$ for y is nonzero, $\pi/2$ for y is zero 인 sinusoid로 나타낼 수 있다.

3-2 Ans)

Phase와 magnitude는 변하지만 period는 변하지 않는다.

4-2 Ans)

gaussian filter는 sobel filter의 경우는 관습적으로 많이 사용되는 filter를 사용하였습니다. 특히 gaussian filter는 sigma가 2이고 크기가 (9X9) 크기의 filter, sobel filter는 (3X3) 크기의 filter로 사용했습니다. (강의자료 lec05_edge_wide page.10)

Nms(Non-maximum-suppression)는 이웃한 gradient 방향의 pixel 들의 interpolation한 값을 비교하여 수행하였습니다. 먼저 임의의 pixel q가 maximum인지 확인하기 위해 gradient 방향의 두 pixel a, b를 잡고 $\tan(\theta)$ 를 이용해 α 를 잡습니다. 그 다음 r 을 $ab + (1 - \alpha)a$ 로 잡습니다. 마찬가지로 p를 잡은 후 q가 $\max(p, r)$ 보다 작으면 0으로 suppress하는 과정을 수행하였습니다. (그림 1 참조) θ 를 크게 4가지 부분으로 나누어 위의 과정을 수행하였습니다.(y = x, y = -x 축으로 하여 기준을 나누었음) 그림 2와 3을 비교하면 두꺼웠던 선들이 nms를 통해 얇은 선으로 detect 된 것을 확인할 수 있습니다.

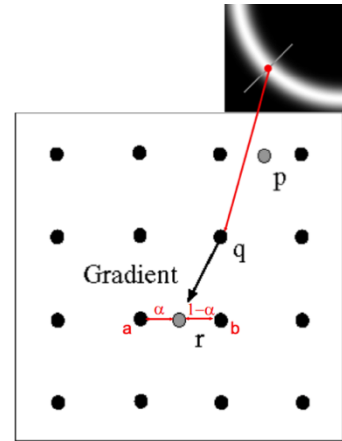


그림 1 nms method

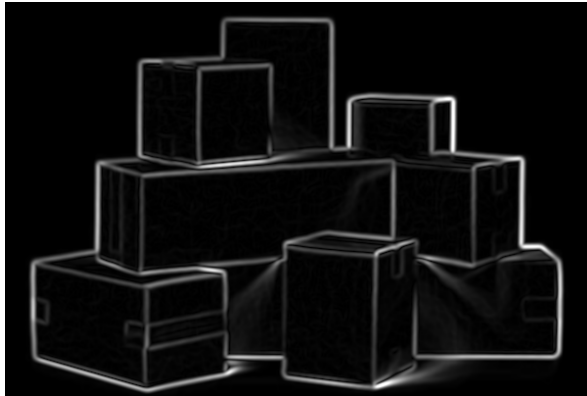


그림 2 Before nms

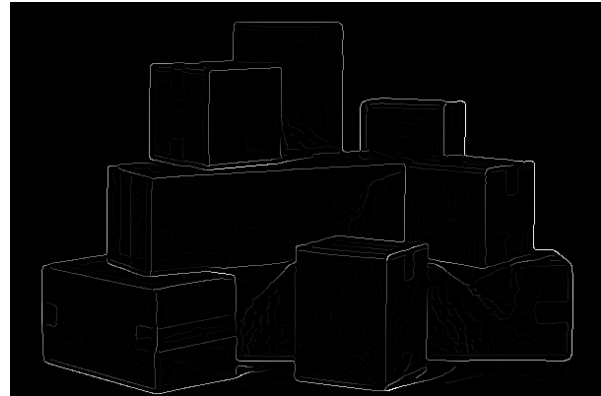
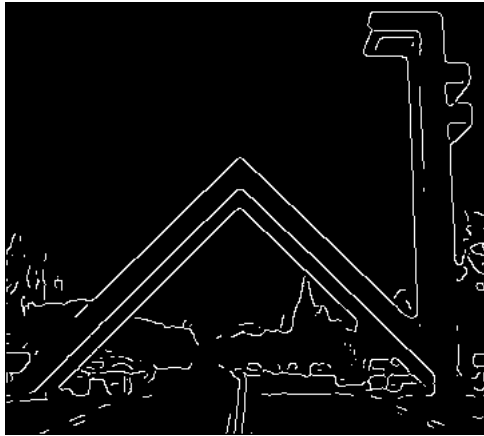


그림 3 after nms

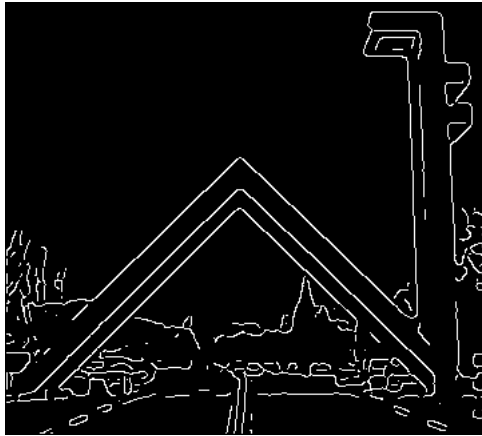
Weak edge 에 대한 처리는 재귀 함수를 이용하여 구현했습니다. 각 pixel을 순회하며 먼저 high threshold임을 확인합니다. High threshold이면 그 주변 픽셀이 low threshold 이상인지를 확인하고 만약 low threshold 이상이면 edge로 detect하도록 하였습니다. 그 주변 픽셀도 체크하여 low threshold 이상이면 edge로 detect하도록 하여 weak edge이지만 high threshold와 연결되어 있는 edge는 억제하지 않았습니다. Low threshold와 high threshold는 경험적으로 구했습니다. 다음은 경험적으로 구한 5가지 이미지에 대한 threshold 표입니다.

	Img01	Img02	Img03	Img04
Low threshold	0.1	0.1	0.1	0.06
High threshold	0.2	0.18	0.2	0.13

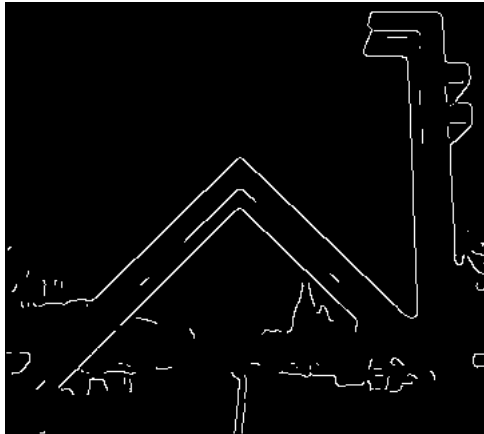
Img 04에 대해서 예를 들어보겠습니다. 가와 나 는 low threshold는 같지만, high threshold가 다른 예입니다. High threshold 가 높을수록 edge로 인식하는 라인이 적어지는 것을 알 수 있습니다. 다와 라는 high threshold는 같지만, low threshold가 다른 예입니다. Low threshold가 낮을수록 weak edge들이 더 잘 인식되는 것을 예상하였으나 low threshold에 의해서는 크게 바뀌지 않고 value가 작은 픽셀이 조금 더 뚜렷해지는 결과를 보였습니다. 두 threshold를 다양한 실험을 통해 각 사진에 적합한 값을 찾았습니다.



가. 0.05 0.2



나. 0.05 0.15



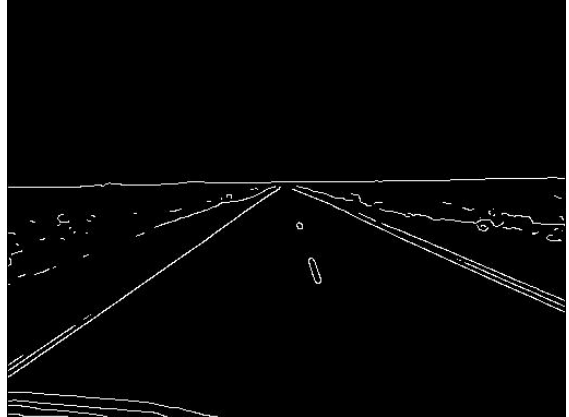
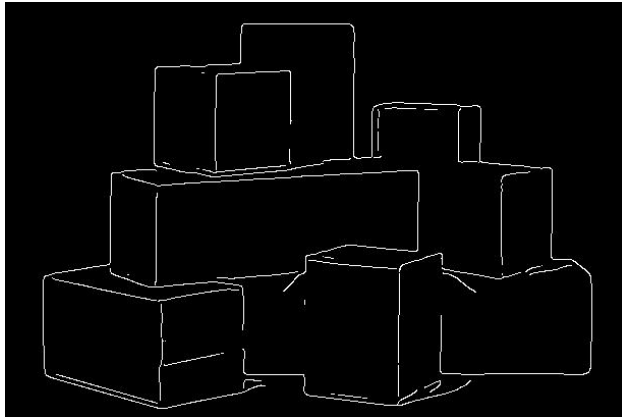
다. 0.25 0.3

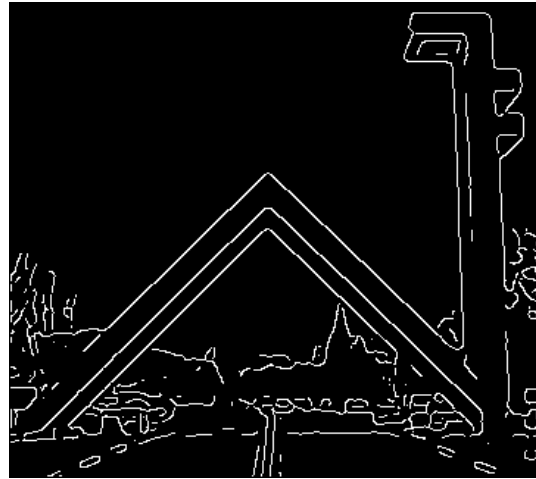
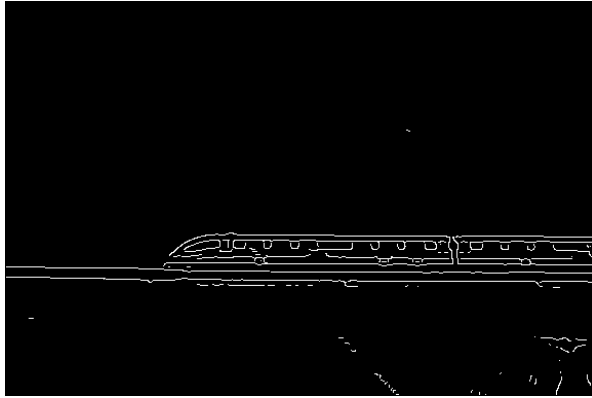


라. 0.01 0.3

4-3 Ans)

Im)





H)



Img 01

Img 02

Img 03

Img 04

각 이미지의 Parameters)

Img #	Sigma	highThreshold	lowThreshold	rRes	thetaRes
1	2	0.2	0.1	1.0	$\pi/180$
2	2	0.18	0.1	1.0	$\pi/180$
3	2	0.2	0.1	1.0	$\pi/180$
4	2	0.13	0.06	1.0	$\pi/180$

Rho resolution과 theta resolution은 각각 1, $\pi/180$ 으로 설정하였습니다. 실험결과 합리적인 시간에 합리적인 결과를 내보였기 때문에 설정하였습니다. Resolution을 너무 작게 하면 더 자세한 직선을 표현할 수 있으나 연산 시간이 오래 걸리는 단점이 있었습니다. 또한 4-4)에서 non maximal suppression에서 설정한 이웃(d)의 기준이 많이 달라져 위와 같이 적당한 수로 통일하였습니다.

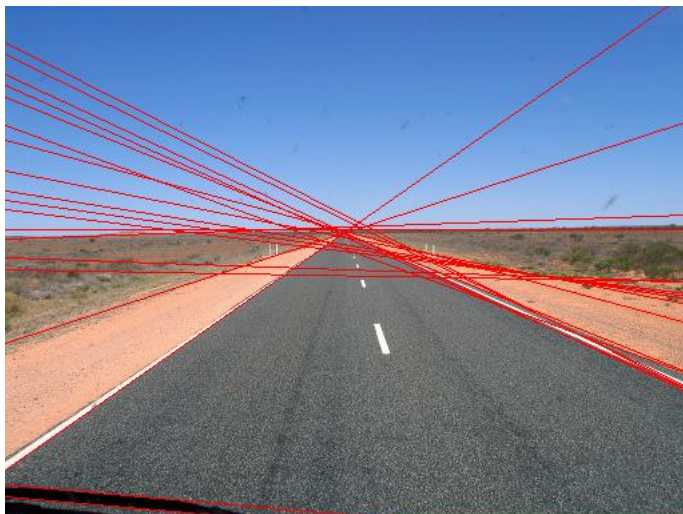
4-4 Ans)

<non maximal suppression>

이미지 크기에 맞는 적당한 크기 d를 잡습니다. 4-2)와는 다르게 이웃한 노드가 아닌 suppression할 대상의 점 (i, j)을 기준으로 d x d 의 박스를 설정하여 nms를 진행하였습니다. 이 박스 내에서 (i, j) 보다 큰 점이 있을 경우 (i, j) 픽셀의 값을 0으로 suppression하고 local maximum이라면 값을 그대로 살려 두었습니다. 이번 실험에서는 d를 5로 설정하였습니다.

< line 선택하기>

Line을 선택할 때는 먼저 (rho, theta) 의 accumulate 값이 최대인 것을 뽑습니다. 그리고 후보에 넣습니다. 다시 accumulate 값이 큰 (rho, theta)를 뽑는데 근방에서 뽑으면 비슷한 직선만 나오게 됩니다. distant_constraint를 정의하여 distant_constraint 안의 영역에서 중복적인 (rho, theta)가 나오지 않도록 설정하였습니다. 위의 방법으로 accumulate 값이 큰 (rho, theta)를 설정하여 상위 20 개를 선정하였습니다. 여기서 distant_constraint는 image의 대각선 길이의 0.03 배로 설정하였습니다.



Rho resolution과 theta resolution의 경우 non maximal suppression과정에서 local maximum d를 잡을 때 중요한 역할을 합니다. 두 resolution을 d보다 작도록 설정해야 정확한 결과를 도출할 수 있었습니다.

4-5 Ans)



<저장 방법>

H, Im, Houghline, Hough segment 는 .py의 디렉토리에 각각 'H.jpg', 'Im.jpg', 'houghline.jpg', 'segment.jpg' 로 저장된다.