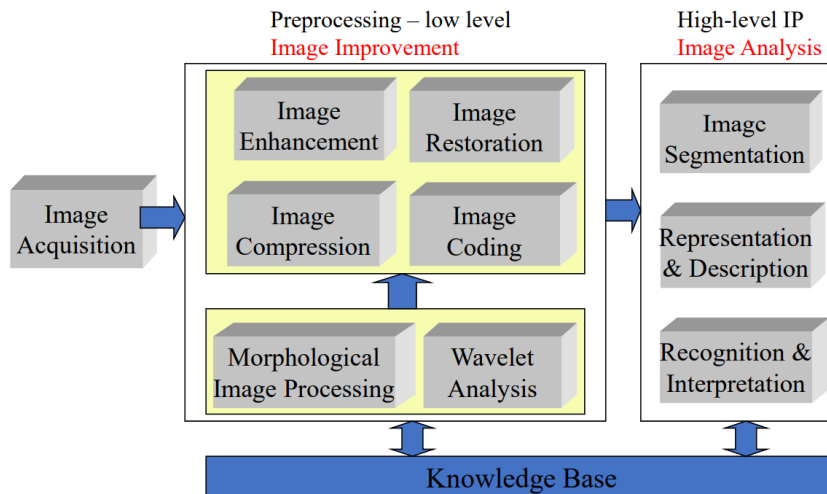


범위 : 강의노트 1-3 (영상처리 개요, 컬러 값 처리, 윤곽선 검출)

2장. Color image processing 컬러 값 처리

1) 3쪽, Image Processing



- High-level(Image Analysis)

: 이미지의 영역을 분할하거나, 이미지를 표현하거나 구분자의 형태로 표현하거나, 이미지 안의 내용을 인식하고 풀이하는 작업

2) 6쪽, Bitmap vs Vector

(1) Bitmap : pixel 형태의 표현법, 픽셀 단위, 아주 복잡한 형태의 색과 형태를 잘 표현할 수 있다, resolution이 정해져 있어서 사이즈가 커지면 깨진다, implement를 구현하기 쉽다.

(2) Vector : object 형태의 표현법, 객체 단위, 간단한 형태만 표현할 수 있다, resolution이 flexible해서 사이즈를 키워도 깨지지않는다, implement를 구현하기 어렵다.

3) 26쪽, Color fundamentals

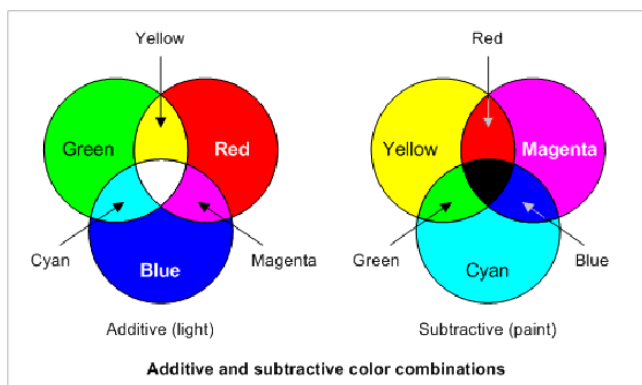
3-1) CIE(International Commission on illumination)

- Primary colors: Red, Green, Blue

- Secondary colors : Cyan, Yellow, Magenta

- Yellow = Green + Red Cyan = Green + Blue Magenta = Red + blue

- White = Yellow + Cyan + Magenta

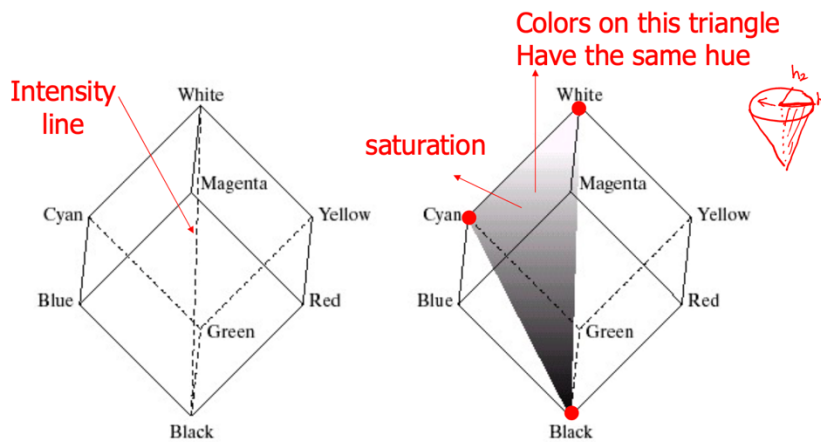


4) 32쪽, Color Models

[1] RGB model [2] CYM model [3] CYMK model [4] HSI model

4-1) 37쪽, [4] HSI model

RGB → HSI model



- Hue : 컬러값
- Saturation = 색깔의 순도(0이면 white, 1이면 primary color)
- Brightness = 색깔의 세기
- Intensity line : white와 black을 연결한 선

5) 45쪽, Monochrome

- 하나의 컬러 채널로 이루어진 이미지

6) 47쪽, Color vector

(1) Per-color-component processing

- 각각 R, G, B 연산을 3번 진행

(2) Vector-based processing

- 각 픽셀이 표현하는 R, G, B vector값에 대해서 어떤 처리를 진행한다.

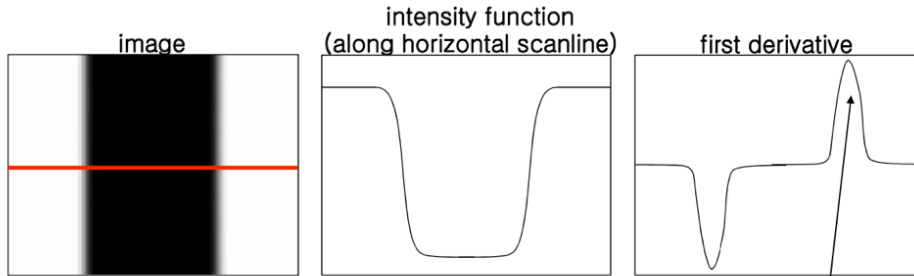
3장. Edge Detection 윤곽선 검출

1) 9쪽, Edge Detection

- Edge direction(orientation) <-> normal

- Edge strength(magnitude)

- Gradient : 어두운 곳에서 밝은 곳으로 가는 방향, $df(x, y)$ 라는 이미지가 있을 때의 dx 방향의 증감량 또는 dy 방향의 증감량이다. 증감량이란, $f(x+1, y)$ 에서 $f(x, y)$ 를 뺀 것이다.



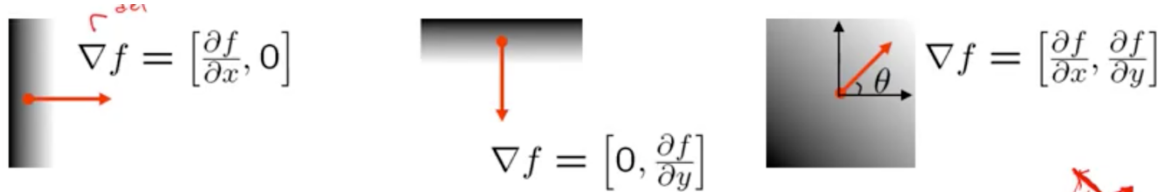
First derivative $f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = f(x+1) - f(x) \quad (h=1)$

Mask $\begin{bmatrix} -1 & 1 \end{bmatrix}$ or $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ (kernel)

Second derivative $f''(x) = \lim_{h \rightarrow 0} \frac{f'(x+h) - f'(x)}{h} = f''(x+1) - f''(x) = f''(x) = f(x+2) - 2f(x+1) + f(x) \quad (h=1)$

Mask $\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

edges correspond to extrema of derivative



The gradient vector points in the direction of most rapid increase in

intensity. The gradient direction is given by $\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$

edge orientation normal \perp gradient orientation

- edge orientation과 gradient orientation은 90도의 관계를 갖고 있다.

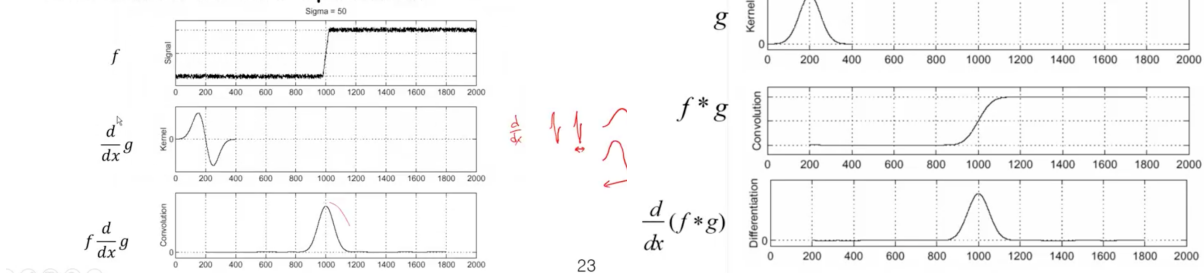
- 미분이란, 오른쪽에서 왼쪽 빼기이다. 미분을 취하면 매우 작은 x 값에 대한 $f(x)$ 의 변화량이다.

2) 22쪽, Smoothing

• This theorem gives us a very useful property:

$$\frac{d}{dx} (f * g) = f * \frac{d}{dx} g$$

• This saves us one operation:



(1) 방법 1

- 입력 신호가 들어오면 가우시안 커널을 이용해 컨볼루션 연산을 수행하고 미분을 수행한다.(2번)
- 위와 같은 방법은 한번 스무딩을 하고 미분을 하고를 2번 연산하게된다.

(2) 방법 2

- 입력 신호가 들어오면 가우시안 커널에 x방향의 미분을 한 것으로 컨볼루션 연산을 수행한다.(1번)

3) 26쪽, Operator

- Edge의 Gradient를 구할 때 마스크를 씌울 수 있다.

(1) Prewitt operator

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(2) Sobel operator

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- 가운데 정보를 더 중요하게 여기겠다는 의미로 가운데를 2로 바꿔준다. 가운데 값이 edge라고 좀 더 가중치를 준다. 노이즈에 좀 더 robust한 결과를 기대할 수 있다.

4) 27쪽, Edge Detection Step

$$(1) \quad f_x = f(x, y) \frac{\partial G}{\partial x} \quad f_y = f(x, y) \frac{\partial G}{\partial y}$$

$$(2) \quad \|\nabla f\| = \sqrt{f_x^2 + f_y^2}$$

$$(3) \quad grad_ori(x, y) = \tan^{-1}(f_x/f_y) \quad \leftarrow \text{Could be used for edge linking}$$

$$(4) \quad \text{if } \|\nabla f\| \geq \text{Threshold, then possible edge point}$$

(1) x방향 gradient를 구하고

(2) y방향 gradient를 구하고

(3) magnitude를 구하고

(4) tan inverse 구하고

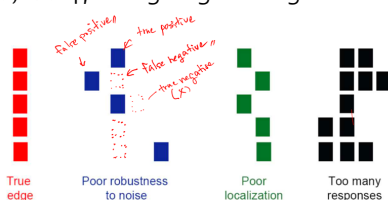
(5) gradient magnitude가 특정 임계값보다 크거나 같으면 그 해당 픽셀이 edge일 가능성이 있다.

- Smoothing을 할 때 smaller kernel(smaller mask)로 하면 작은 edge를 검출할 수 있고, larger kernel로 하면 noise(작은 edge)가 많이 제거된 edge를 검출 할 수 있다.

- Smoothing을 많이 하면 noise는 제거되지만 localization에는 좋지 않다.

- gradient magnitude를 작은 임계값으로 처리하면 잡을 추출해야하는데 잡 외적인 것들이 나온다.

5) 35쪽, Designing an Edge detector (edge 검출기 성능 평가)



- 엣지 검출에서 대다수에 해당되는 true-negative는 배제한다.

- 엣지 검출의 목표는 false-positive와 false-negative를 최소화하는 것으로 한다.

6) 36쪽, Canny Detector

- gaussian 1차 미분 계수를 이용해서 edge를 구하는 방법에 몇가지를 추가한 것이다.

(1) 오리지널 이미지에 대해서 gaussian 커널을 x방향, y방향으로 Derivative를 구하고 미분 계수를 구한다.

(2) 미분형 커널을 만들고

(3) x방향, y방향 gradient를 구한다.

(4) gradient magnitude를 구하게 된다.

(5) 탄젠트 인벌스를 구한다.

- 비슷한 orientation을 나타내는 값(edge 값)은 비슷한 컬러 값을 가지고 있다. (비슷한 방향끼리 매핑되어있음)

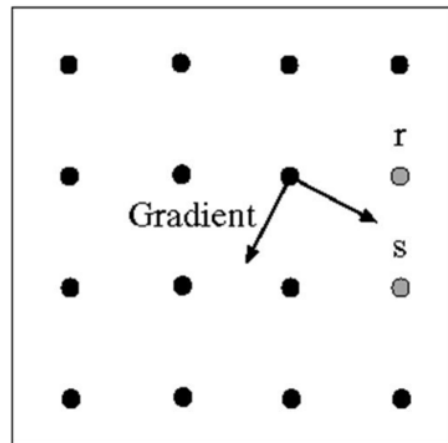
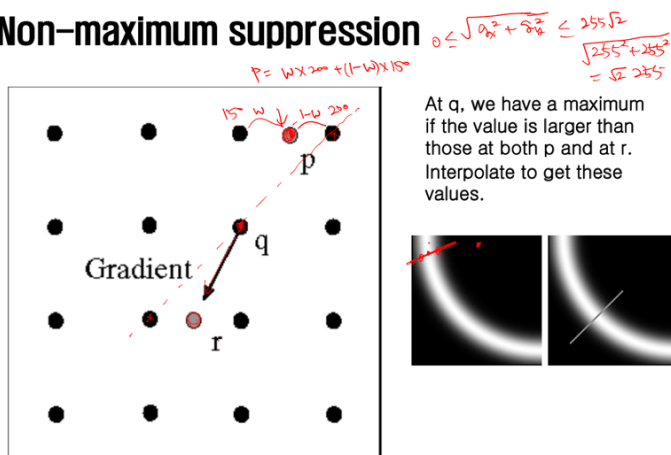
(6) Non-maximum suppression

- q를 기준으로 봤을 때, q값이 인접한 2개의 픽셀(gradient orientation 방향에 있는 인접한 pixel)인 p, r을 봤을 때, q보다 작으면 q는 maximum이다. p, r이 q보다 크면 q는 maximum이 아니다.

- maximum이 아니면 suppression 시키겠다.

- 해당 작업을 모든 pixel에 대해서 수행하겠다.

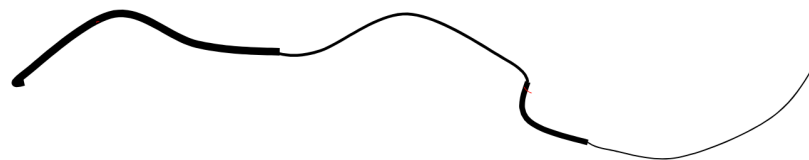
Non-maximum suppression



(7) Edge Linking

- gradient orientation의 방향에 수직인 gradient normal인 방향으로 선을 그어 그 방향으로 인접한 두 픽셀의 중간값(interpolation)을 통해 끊어진 edge에도 연결하여 edge를 구할 수 있다.

(8) Hysteresis thresholding



- 2개의 threshold(임계값)을 이용한다.

(1) 세타1보다 큰 edge로 검출하고(두꺼운 edge 검출) 세타2보다 큰 edge로 본다.

(2) 세타2보다 큰 edge로 잡게되면 noise한 edge들이 많이 검출된다.

(3) 그래서 세타2보다 큰 것 들은 세타1에서 얻어진 edge하고 연결된 것들만 edge로 살리겠다.

(4) 그러면 세타1에서 나온 edge와 연결되지 않기때문에 noise한 edge들이 제거된다.

(5) 결과적으로 edge를 잘 뽑아낼 수 있다.