**(CH2) Geometric transform and its properties.**

2D, 3D에 대한 points, lines and planes(면) 표현

-2D points: **x**=(x, y, w)로 표시되면 w: scaling(원근)

(w, y, w) 🡪 (X, Y) 🡪 (x/w, y/w) 표시

예) (1, 2, 3) 🡪 (1/3, 2/3) and (2, 4, 6) 🡪 (2/6, 4/6) 🡪 (1/3, 2/3)

(1,2,3)과 (2,4,6) 같은 (X, Y)에 찍히나 w값이 다름

-2D lines: **x**ㆍ**l** = ax + by + c = 0 – 내적관련

**l** = (a, b, c)

-3D points: **x** = (x, y, z, w)

-3D planes: **x**ㆍ**m** = ax + by + cz + d =0

**m** = (a, b, c, d)

2D Transformation

-이동: T(ty, tx) = ([[1,0,0], [0,1,0], [ty,tx,1]]) -> x’=x+tx, y’=y+ty

-회전: R(θ) = ([[cosθ,-sinθ,0], [sinθ,cosθ,0], [0,0,1]]) -> 시계방향으로 θ만큼(반시계 -θ)

-크기: S(sy, sx) = ([[sy,0,0], [0,sx,0], [0,0,1]]) -> y방향 sy, x방향 sx만큼 확대

x’ = Ax : A가 상기 행렬임

2D transformations

-rigid: 마름모이동, DoF(3-tx, ty, θ), Preserves(lengths)

-affine: 평행사변형이동, DoF(6-tx, ty, θ, scaling 2개, shearing), Preserves(parallelism)

3D transformations

-rigid: DoF(6-tx, ty, tz, θx, θy, θz), Preserves(lengths)

-affine: DoF(12), Preserves(parallelism)

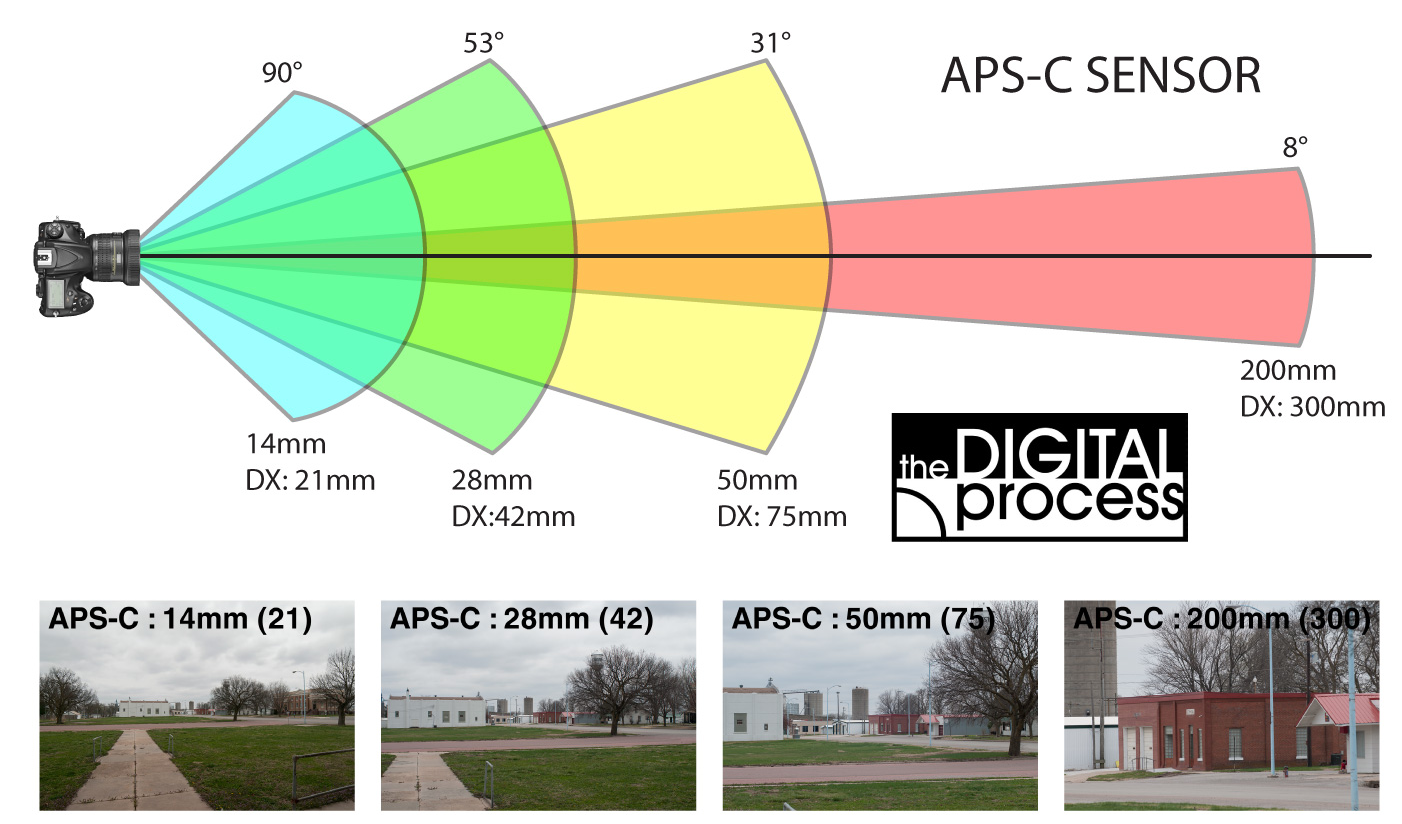
**(CH2) filed of view and focal length**

Field of view(시야)

-Field of view depends on focal length(f)

-As f gets smaller, 넓은 부분을 투영하고

-As f gets larger, 좁은 영역을 망원효과로 투영한다.



-시야(φ)와 초점길이(f)는 반비례: 초점길이가 짧아지면 넓은 시야, 초점길이가 길어지면 좁은 시야 🡺 φ = d/2f

(CH3) two types of reflections and their properties.

1)BRDF

2)Diffuse reflection – 난반사

사방팔방으로 균일하게 재발산 (의류, 종이 및 아스팔트같이 미세하게 거친 표면에서 반사)

색은 표면의 특성에 따라 크게 달라짐

3)Specular reflection - 정반사

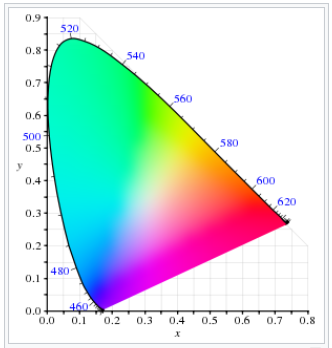
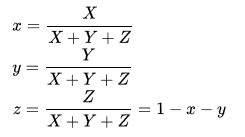
거울과 같은 매끄러운 표면 반사,

높은 방향성

입사광과 동일한 색상(물체에 독립적인 것으로 가정)

(CH4) color spaces and their properties.

1)CIE color



2)RGB color:

-컬러모니터, 비디오카메라에 사용

-RGB는 모두 Black (0,0,0)으로부터 White (1,1,1)로

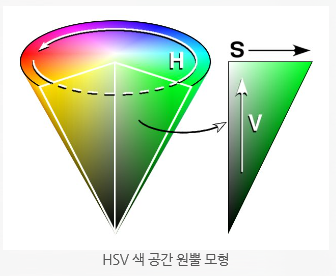
3)CMY(cyan, magenta, yellow) color:

-컬러프린터에 사용(CMYK), RGB를 45도 기울이는 것

-따라서 CMY는 black (1,1,1)이 된다.

-[C,M,Y] = [1,1,1] – [R,G,B]

3)HSV/HSI color

-

-HSV:

Hue(색상: 0, 360도를 빨강을 기준으로 원)

Saturation(채도: 중심에서 밖으로 흰 🡪 원색)

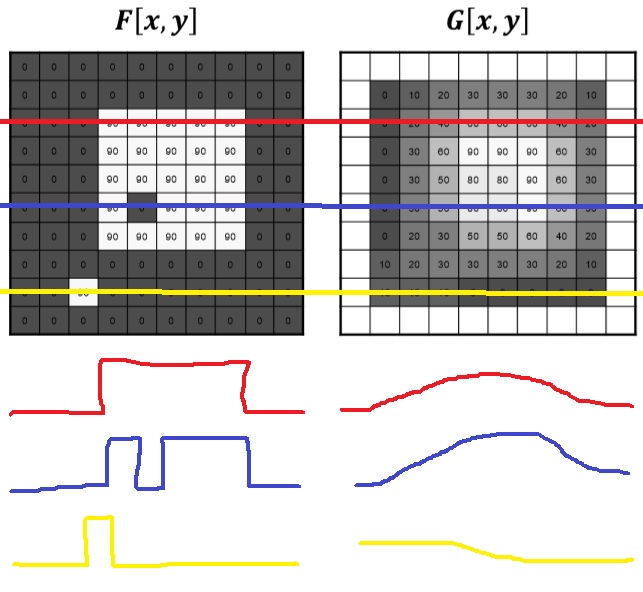
Value or Intensity(명도: 원뿔 하단에서 위로 검 🡪 흰(밝은)색)

1. 조도(H, S)와 휘도로 구성요소가 분리됨
2. Hue와 Saturation은 인간의 시각 시스템과 관련(이래서 인공지능에 활용)
3. RGB는 이미지 색 **생성에** 적합, HSV는 이미지 색 **인식에** 적합

(CH5) properties of moving average filter. Comparison between convolution and correlation.

-이미지에 커널의 평균으로 convolution하면 이미지의 프로파일을 보면 Smoothing효과가 발생한다.

예)Moving Average(=커널 평균으로 convolution), 3\*3커널이면 1/9 \* np.ones



Convolution:

-필터가 이동하면서 겹치는 양을 나타냄.

-g[i,j] = ΣΣh[u,v]f[i-u,j-v]

Correlation:

-필터가 이동하면서 유사성의 측정값을 계산.

-상관 결과는 두 신호가 가장 일치하는 시점에 최대치

-상관성은 두 신호의 관련성의 척도

- g[i,j] = ΣΣh[u,v]f[i+u,j+v]

Gaussian filter: ” 패턴이면 가우시안 패턴임

-당연히 filter의 크기가 커질수록 Smoothing 커지고

-σ가 커질수록 봉우리가 완만해져 Smoothing 커짐

-커널size와 σ의 두가지 복잡도를 가진다.

(CH6) Important Fourier Transform properties.

-space domain ->FT-> frequency domain

: 이미지를 FT한 후 log, abs, shift를 하면 중심(u=0, v=0:저주파수)이 밝은 magnitude 그림이 나옴.

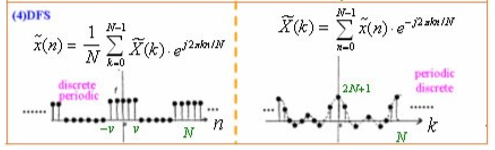
DFT는 시간과 주파수 도메인 모두 불연속적이며 유한한 시퀀스임.

Fourier Transform: 비주기이며,

x(t)

X(w)

대부분의 데이터는 주기적 이산 시간 신호로 제공됨.



-Fourier transform: F(u,v) = ΣΣf(x,y)

-Inverse Fourier transform: f(x,y) = ΣΣF(u,v)

-Polar representation: F(u,v) = |F(u,v)| 🡺 magnitude & phase

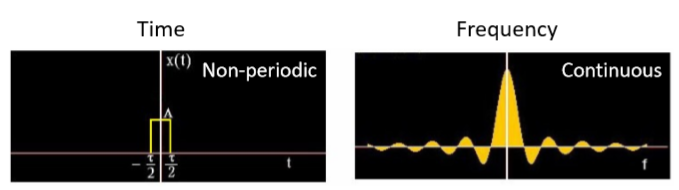
-Spectrum(Magnitude): |F(u,v)| = R(u,v) + jI(u,v) 🡺 실수부(R)와 허수부(I) 나뉨.

-Convolution: f(x,y) \* h(x,y) = F(u,v) X H(u,v) 🡺 \*행렬곱, X곱셉

🡪컨볼루션은 FT하면 행렬곱을 곱셉으로 바뀌어 계산이 용이

-Gaussian: Gaussian은 FT해도 Gaussian형태로 변환

-Rectangle:



(CH6) Comparison among three low pass filters. Implementing high pass filter from low pass filter.

**[Low Pass Filters]**

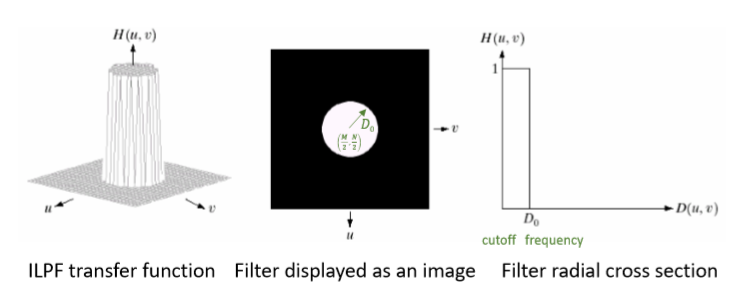
-Fourier Transform을 하면 Low frequency에 에너지들이 몰려 있음

-High frequency: intensity가 급격히 변하는 엣지, 노이즈 등

-Low Pass Filter(LPF): High frequency를 제거하여 “Smoothing”효과

-종류:

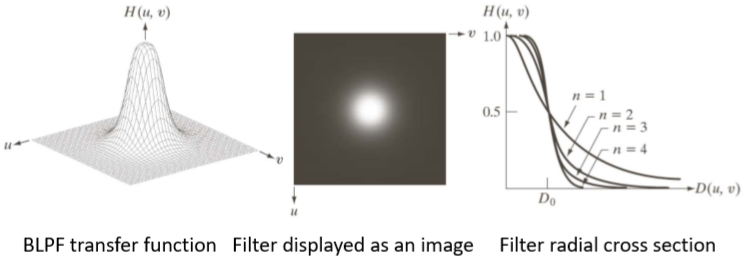
(1) Ideal LPF: D0를 조절하며 ringing이 존재



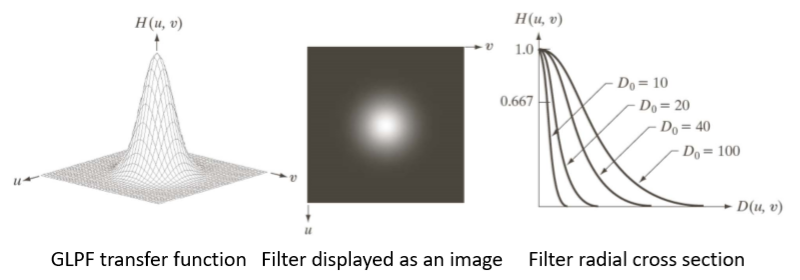
(2) Butterworth LPF: H(u,v) = 1/1+|D/D0]^2n

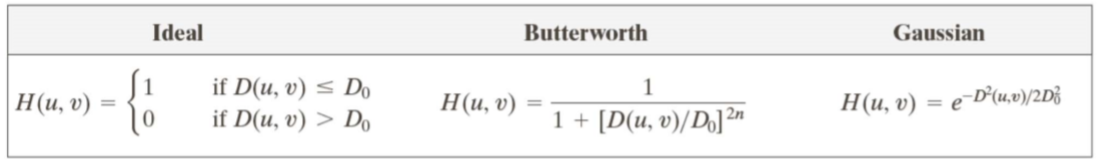
-n을 값으로 기울기를 변화(n이 커지면 ILPF와 비슷해짐)

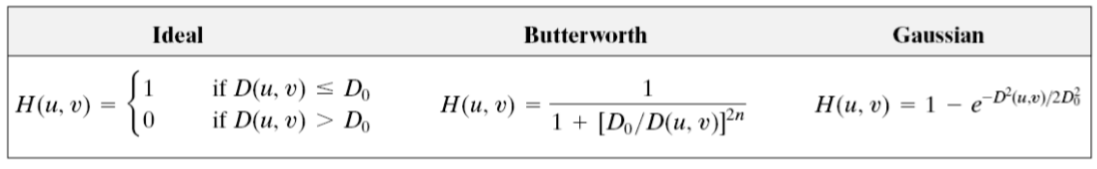
-ringing이 없어짐



(3)Gaussian LPF: ILPF, BLPF는 D0고정, GLPF는 D0가변

 -

-LPF

-HPF

(CH7) Properties of two multi-scale representations.

-original image(N x N) ->smoothing(Guassian) -> down sampling(N/2 x N/2) -> up sampling(N x N)

-up sampling image: smoothing을 통해 고주파수(엣지) 정보가 없어짐

-따라서, original image – up sampling image = 엣지 정보

-smoothing & down sampling을 계속하면 최하위 이미지에서는 큰 엣지정보를 찾아내고 up sampling을 하면서 점점 섬세한 엣지를 역으로 찾아가야 한다.

-original image를 Gaussian smoothing, up/down sample을 하면 복잡한 굴곡을 완만하게 하여 local minima를 피할 수 있다.

-Gaussian pyramid: smoothing + down sampling … 1X1 image까지

-Laplacian pyramid: Gaussian pyramid를 통해 1X1영상에서 up sampling하면서 해당 gaussian image와 “-“서 얻어내는 이미지

-Wavelet: 고주파 필터 + down sampling, 저주파 필터 + down sampling을 통해

(CH8) Edge property and image derivative.

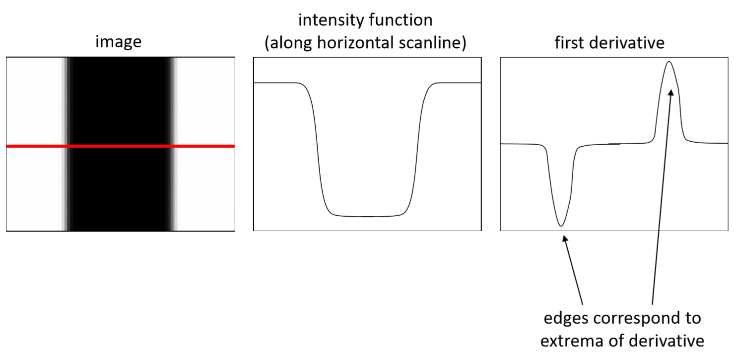
-Filtering(smoothing)

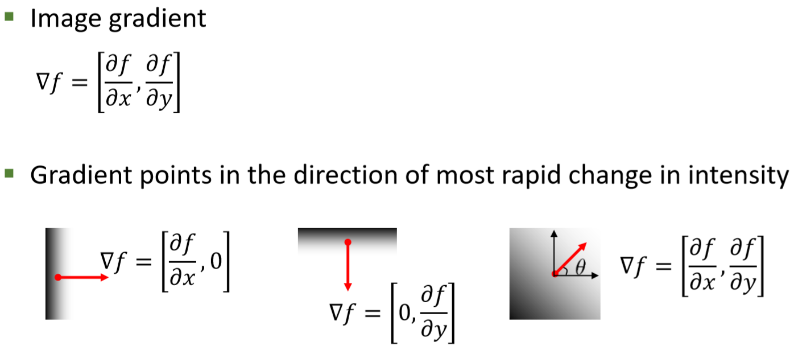
-Enhancement(image negative(반전), power law: 감마값을 변하여 밝기, 그래프 생각해봐)

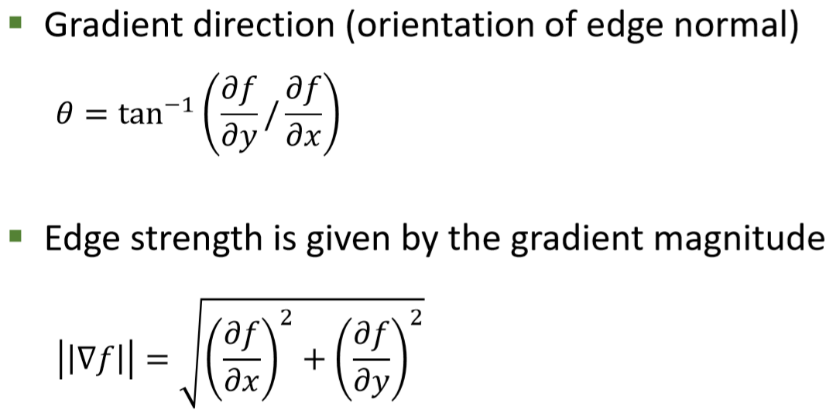
-Detection

1)edge: intensity에 가파른 변화가 되는 곳

2)line image를 profiling하면



-



(CH8) Important edge detection filters (mask coefficients)

-Basic edge detection filter

(1)Prewitt: 대칭 3x3에 <x축 좌우쪽 -1, 0, 1><y축 상하 1, 0, -1>

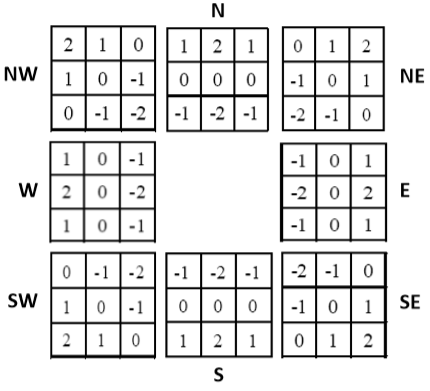
(2)Sobel: 대칭 3x3에 Prewitt의 센터 값을 -2, 2로

(3)Roberts: 비대칭 2x2에 대각선으로 각각 1, -1

-상기 필터는 x축, y축으로 각각 convolution한 값을 더하고 threshold를 통해 버릴 edge값 설정

-**Compass gradient masks**: 8개 direction으로 magnitude를 표시

(1)나침반 방향으로 8개 마스크로 정렬

 (2)gradient magnitude가 큰 쪽을 선택

-**Canny edge detector**:

(1)Guassin 커널 사용

(2)최대값에 직각으로 찾아감

(3)Thresholding을 연결해 나감(\*hysteresis threshold)

-이미지의 윤곽선을 복구개선

-높은 임계값에서 시작해서 작은 임계값으로 윤곽선을 이어감.

-**LoG (Laplacian of Gaussian)**: 엣지들이 너무 많으니 1차미분/2차미분을 통해 엣지를 찾아냄

(1) Edge from derivatives

-1차미분의 peak 🡪 gradient(벡터)

-2차미분의 Zero-crossing 🡪 Laplacian(스칼라)

-두가지를 만족하는 부분이 엣지다.

-식: g(x) = d2/dx2(h\*f) = d2/dx2(h)\*f -h: gaussian filter

(2) LoG 정리

-Smoothing filter – Gaussian

-2차 미분(라플라시안) 사용

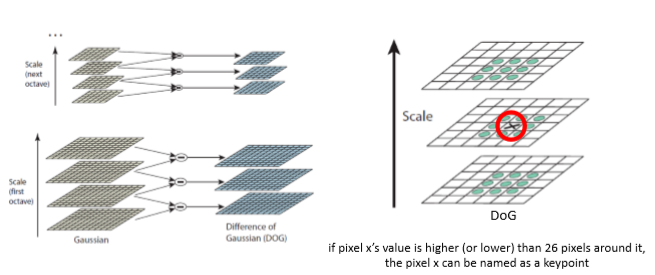
-1차 미분 peak과 2차 미분 zero-crossing이 있어야 엣지

-SIFT(scale invariant feature transform)

(1) keypoint detection

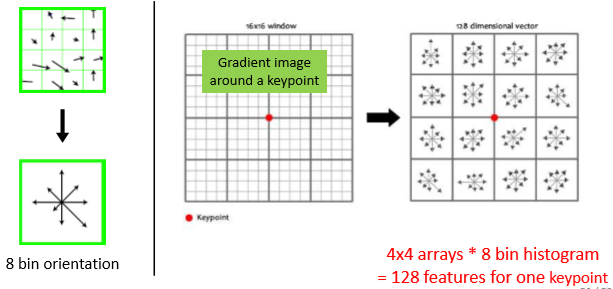
-DoG(라플라시안)과 가우시안을 사용하여 keypoint를 찾아냄

-multi scale에서 동일 영역에서 가장 높은 값을 keypoint라 함



(2) keypoint description

-주어진 scale에 local gradient directions를 히스토그램으로 만듬

 -keypoint를 중심으로 하는 4x4배열에 1배열당 8 방향 히스토그램을 표시

(따라서 128개의 features로 표현됨)

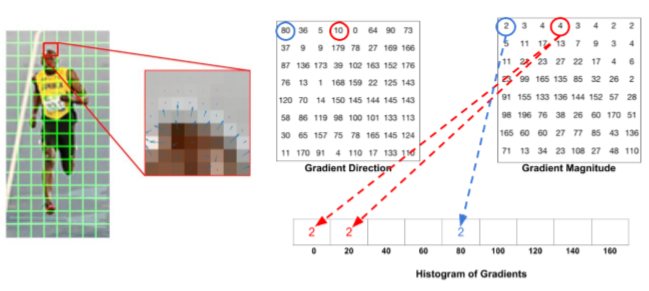
-HoG(Historam of Oriented Gradient)

1) Gradient 계산 (prewitt으로?)

2) 8x8 cells의 Gradient 히스토그램 계산

- Gradient Direction(9개 각도로)과 Gradient Magnitude

- 해당 direction각도에 magnitude값을 히스토그램화



3) 4개씩 block화하여 Normalization

4) HOG feature vector를 계산