

Image Filtering

Grayscale

- pixel 1개 당 0-255(0 : dark, 255 : white)

Color

- 3개의 channel. channel당 RGB
- HSV(Hue, Saturation, Value)

Image Transformation

- $g(x, y) = f(x, y) + 20$: 밝기 증가(20)
- $g(x, y) = f(-x, y)$: 좌우 반전

Convolution

Image denoising(이미지 노이즈 제거)

: Image가 noise를 가지면 제거하는 방법

가장 대표적 : 주변 값을 이용하는 방법(ex) 주변 pixel 값의 평균(mean filtering))

- detail한 것들을 blurring 하면서 사라질 수 있도록 해준다.

Filtering

: 기존의 pixel 결합을 통해 새로운 image를 만들어내는 것

왜 사용?

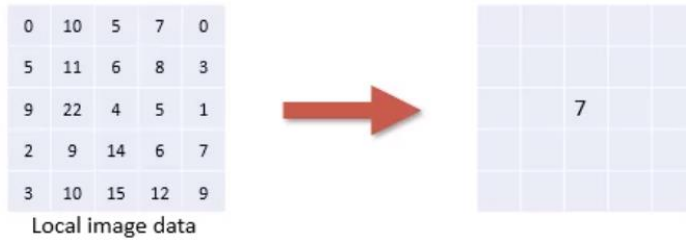
- 이미지로부터 의미있는 정보를 얻고자 할 때
- 이미지를 개선하기 위해

Linear Filtering(cross-correlation, convolution)

- kernel(=mask, filter)를 이용해서 linear filtering을 한다.

Mean Filtering

A more general version



$$S[f](m, n) = \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f(m + i, n + j)$$

$$- w(i, j) = 1/(2k+1)^2$$

- Cross correlation

$$S[f] = w \otimes f$$

$$S[f](m, n) = \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f(m + i, n + j)$$

- Convolution

$$S[f] = w * f$$

$$S[f](m, n) = \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f(m - i, n - j)$$

중요한 것!

- 1) 선형성(Linearity)

$$w' = aw + bv$$

$$w' \otimes f = a(w \otimes f) + b(v \otimes f)$$

- 2) Shift Invariance

convolve -> shift == shift -> convolve

$$f'(m, n) = f(m - m_0, n - n_0)$$

$$(w \otimes f')(m, n) = \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f'(m + i, n + j)$$

$$= \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f(m + i - m_0, n + j - n_0)$$

Filter

1) Original



Original (f)



0	0	0
0	1	0
0	0	0

Kernel (k)

= Original

2) Left, Right, Up, Down Shift

ex)



Original



0	0	0
1	0	0
0	0	0

= 1pixel left shift

3) Blur



Original (f)



$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Kernel (k)

= Blur

4) Sharpening

- 4-1) Original - Blur = detail한 부분(어떻게 보자면 Noise)
- 4-2) Original + ($\alpha \times$ 4-1에서 구한 detail한 부분) = Sharpening Image

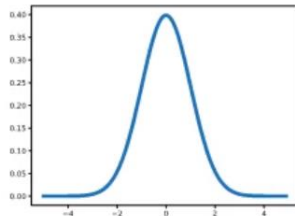
$$\begin{aligned}f(\text{sharp}) &= f + \alpha(f - f(\text{blur})) \\&= (1 + \alpha)f - \alpha(\text{blur}) \\&= (1 + \alpha)(w * f) - \alpha(v * f) \\&= ((1 + \alpha)w - \alpha v) * f\end{aligned}$$

[*는 곱셈이 아니라 convolution 연산]

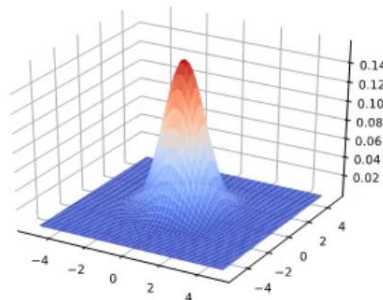
5) Gaussian Filter

- 멀리있는 것보다 가까이 있는 것의 영향이 더 크다는 점에서 착안.

$$G_{\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

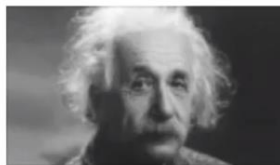
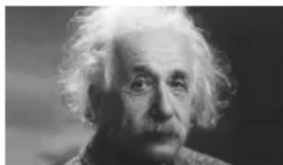


$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



계수 값에 따라 다른 결과를 반환.

- high-frequency(noise가 많은 부분)을 제거한다고 해서 **low-pass filter**라고도 부른다.
- gaussian filter를 응용하면 detail한 부분 또한 구할 수 있다. -> high-pass filter



21x21, $\sigma=0.5$



21x21, $\sigma=1$



21x21, $\sigma=3$

Convolution에서 가장 중요한 것

1) Linearity

2) Shift invariance

수학적인 처리를 잘 해낼 수 있다.(All **Linear shift invariant(LSI)** System can be expressed as a convolution)

6) 대표적인 Non-linear filter

1) Thresholding

: 특정 값을 넘은 것만.



$$g(m, n) = \begin{cases} 255, & f(m, n) > A \\ 0 & otherwise \end{cases}$$

2) Rectification

: 0 이상인 것만, 양수만 취급

$$g(m, n) = \max(f(m, n), 0)$$

3) Medium(중간값)

: 평균 값은 outlier에 매우 민감 -> 중간값(medium)을 이용

-> Noise(High Frequency)를 훌륭하게 제거 가능.

