

## Lab 2. Image Filtering

**실습 목표:** 몇 가지 기본적인 디지털 이미지 필터링 기법, Shrpeneing, Deblurring 등의 이미지 화질 향상 기법에 대하여 이해하고 이에 대한 프로그램을 작성한다.

### 목차

- 이미지 필터링
- Average 필터링
- Median 필터링
- Sharpening
- Deblurring
- 기타 이미지 필터링 및 화질 향상 기법
- 이미지 필터링 프로그램 구현
- 숙제: Average 필터와 Median 필터의 효율적 구현
- 참고문헌

## 이미지 필터링

이미지 필터링은 기본적으로 노이즈 제거, 주요 부분의 강조 등을 통하여 이미지의 화질 및 유용성을 향상시키는 것을 목적으로 한다. 이러한 필터는 선형 및 비선형으로 나뉘며 사각형, 원형 등 다양한 모양을 사용할 수 있지만 주로 사각형 모양의 필터가 가장 많이 쓰인다. 선형 필터의 경우 주로 convolution 연산을 통하여 구현하고, 이 경우 필터를 커널(Kernel)이라고 지칭한다. 공식 1은 이미지  $f$ 로부터 커널  $h$ 를 적용하여 Convolution 연산을 통해 필터링된 이미지  $g$ 를 얻는 방법을 보여준다. 이는 일반적인 표현식이며 실제 구현 방식에 따라 연산 방식에 약간의 수정이 필요하다.

$$g(m,n) = f \star h = \sum_{l=l_0}^{l_1} \sum_{k=k_0}^{k_1} f(m-k, n-l)h(k,l)$$

식 1. 일반적인 Convolution 공식

## Average 필터

Average 필터는 가장 기본적인 필터링 기법으로, 각 픽셀 값에 대하여 주변의 픽셀 값들과의 평균 값을 구하여 결과이미지에 할당하는 방법이다. 이는 시그널 프로세스의 로우 패스 필터에 해당하며 작은 노이즈들을 제거해 주는 효과가 있다. Averag 필터는 보통 smoothing 또는 blurring 으로도 불리며, 그 이름이 나타내는 바와 같이 이미지의 세부적인 영상을 흐리게 하므로 과도한 average 필터링은 전체적인 화질의 저하를 불러올 수 있다. 그림 1은 간단한 average 필터링 및 가우시안 커널을 보여준다.

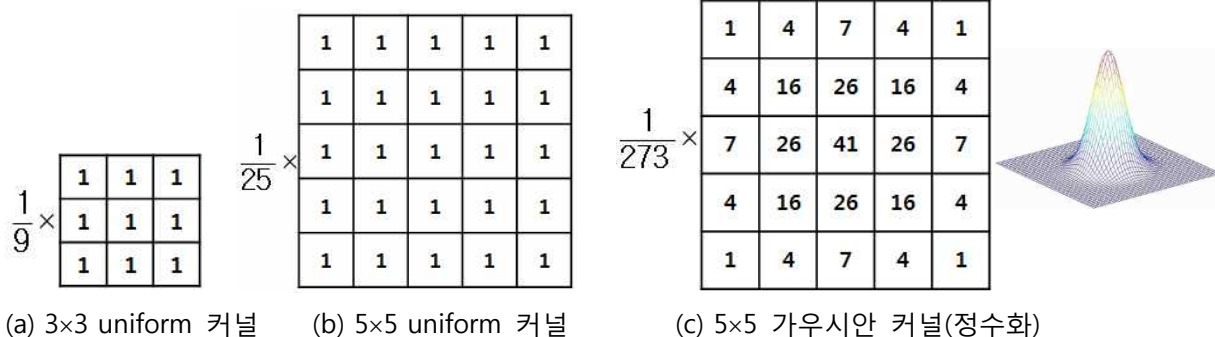


그림 1. 몇가지 기본적인 이미지 필터링 커널. 커널 값의 합은 항상 1이 된다.

그림 2는 3×3 uniform 필터를 이용하여 average 필터링을 수행하는 과정을 도식적으로 보여준다. 여기서 커널이 이미지의 영역을 벗어나는 경우, 이미지 외부의 값들을 정의해 주어야 하는데, 이를 패딩(padding)이라고 한다. 주로, 0의 값을 할당해 주게 되나, 원래 이미지의 픽셀 값들을 확장하는 방식으로 패딩을 하기도 한다. 공식 1을 그림 2에 보여지는 연산 방식으로 표현하면 공식 2와 같이 된다. 실제 구현 시에는 Array의 인덱스가 음수가 되지 않도록 주의하여야 한다.

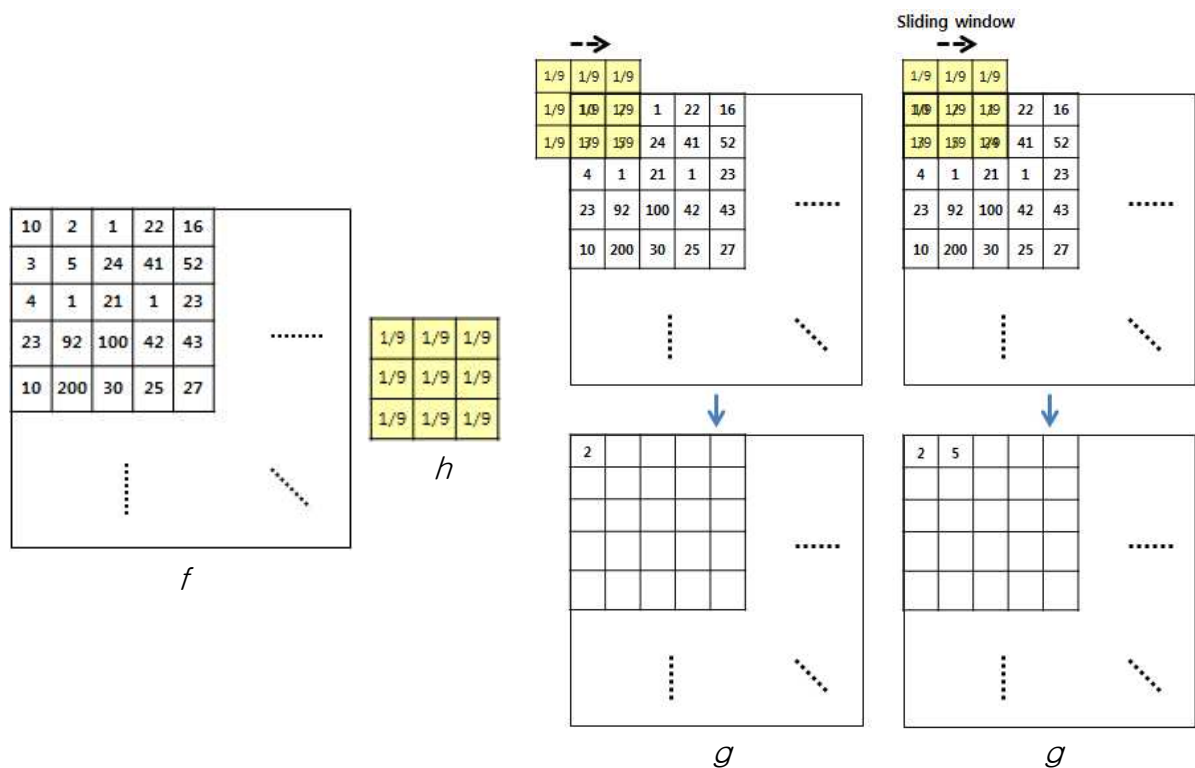


그림 2. 3×3 uniform 커널을 이용한 Average 필터링의 예

$$g(m, n) = f * h = \sum_{l=-w}^w \sum_{k=-w}^w f(m+k, n+l) h(k, l), \quad w = \lfloor w_0 / 2 \rfloor$$

식 2. 그림 2 에 해당하는 Convolution 공식.  $w_0$ 는 커널의 폭을 나타냄 (보통 가로, 세로의 길이가 같으며, 홀수임).

Uniform 필터를 이용한 Average 필터링은 종종 과도한 smoothing을 초래하므로 커널의 중앙 값에 더 큰 가중치를 주는 weighted average filtering이 사용되기도 한다. 그림 1에 보여진 Gaussian 커널이 대표적인 weighted average filtering 커널이며, 가우시안 함수로부터 커널 값을

구한다. 공식 3은 일차원 및 이차원 Gaussian 함수를 보여준다( $\sigma_x = \sigma_y, \sigma_{xy} = 0$ ). 그림 1(c)는 디지털 이미지 연산의 편의성을 위하여 정수화된 커널 값을 보여준다.

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu_x)^2 + (y-\mu_y)^2}{2\sigma^2}}$$

식 3. 1-d 와 2-d Gaussian 커널.

가우시안 함수는 여러 개의 가우시안 함수를 서로 더하여도 다시 가우시안 함수가 얻어지는 특성이 있으므로, 이미지에 반복적으로 가우시안 필터를 적용하여, 더 강한 가우시안 필터를 적용한 효과를 얻을 수 있다. 그림 3은 여러가지 사이즈의 uniform 커널을 이용하여 Average 필터링을 한 예를 보여준다. 커널 사이즈가 커짐에 따라 과도한 blurring이 일어나는 것을 알 수 있다.

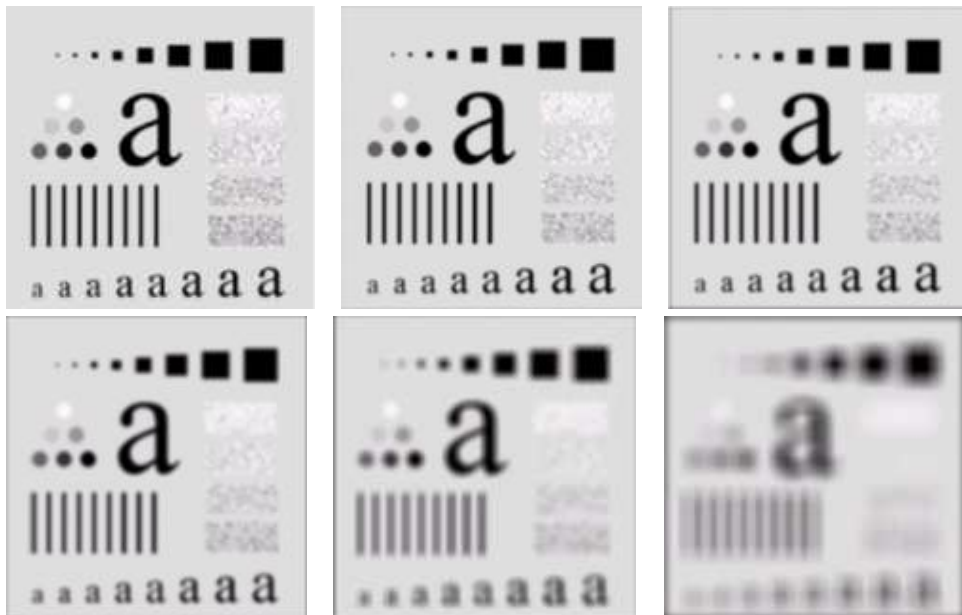


그림 3. 500×500 이미지를 3, 5, 9, 15, 35 Average 필터로 처리한 결과

Average 필터링의 연산 과정을 보면 굉장히 많은 중복된 연산이 수행된다는 것을 알 수 있다. 이를 해결하기 위해 Moving Average라는 연산법이 쓰이고 있으며 이는 실습 시간에 다루게 된다.

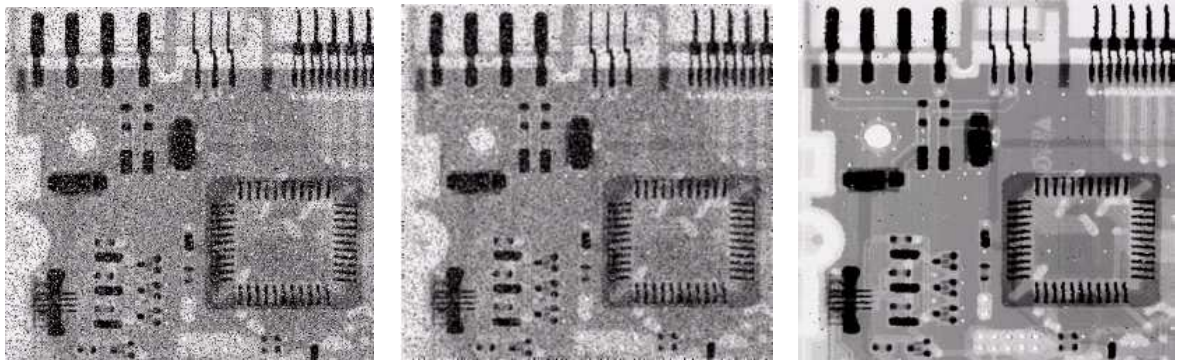
이 밖에, 노이즈를 제거하는 동시에 blurring 효과를 상쇄하기 위해 adaptive average 필터링, edge preserving average 필터링 기법 등이 쓰이기도 한다.

## Median 필터링

---

Median 필터링은 Average 필터링과 비슷한 방법으로 필터 윈도우(박스)를 이동시켜 가며, 윈도우 안의 픽셀 값들의 미디언 값을 결과 이미지(g)에 할당한다(3×3 윈도우를 사용한다면, sorting 후 5 번째 값에 해당). Average 필터링과 다르게 작은 노이즈 값들이 주변의 모든 픽셀 값에 영향을 미치는 효과를 억제할 수 있으며, 따라서 Salt & Pepper 타입의 작은 노이즈들을 제거하는데 탁월하고 이미지의 sharpness도 잘 보존된다. 그림 4 는 average 필터링과 median 필터링의 결과를 비교하여 보여준다.

Median 필터링은 Average 필터 등과 같이 커널, 또는 convolution 연산으로 표현할 수 없으며 linear 연산에도 해당되지 않는다. Sorting 작업으로 인하여 Average 필터링 등의 convolution 기반 필터링보다 처리 속도가 느리다. Median 필터링을 효율적으로 구현할 수 있는 방법에 대하여는 속제 2에서 다루게 된다.



(a) Salt & pepper 노이즈 이미지    (b) average 필터링 결과    (c) median 필터링 결과

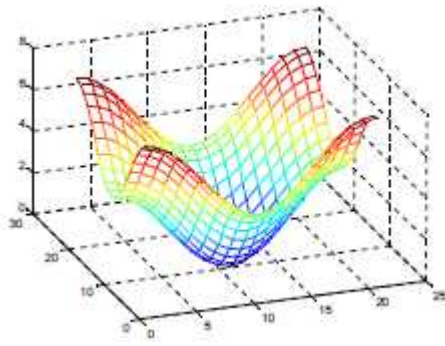
그림 4. Average 필터링과 median 필터링의 예.

## Sharpening

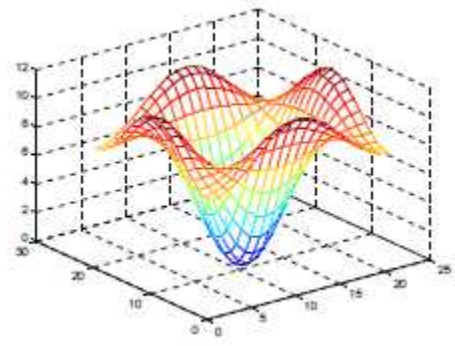
Sharpening 은 이미지의 선 등의 세부적인 부분을 선명하게 하는 것을 목적으로 한다. 보통 하이패스 필터링 결과를 원본 이미지에 더함으로써 선이나 모서리 등의 영역을 부각시키게 된다. 그림 5는 두가지 하이패스 필터 커널을 보여준다. 그림 6은 이러한 필터를 이용하여 sharpening을 실행한 예를 보여준다.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



(a)



(b)

그림 5. 두가지 하이패스 필터 커널의 도식적 표현.



(a)



(b)



(c)

그림 6. Image sharpening의 예. (a) 원본 이미지 (b) 하이패스 필터링 결과 (c) Sharpened 이미지(원본 이미지에 하이패스 필터링 결과를 더한 이미지).

## Deblurring

Deblurring은 다양한 이유로 화질이 저하된 이미지의 화질을 다시 향상시키는 것을 목적으로 한다. 공식 1에서와 같이 필터링된 이미지를  $g = f * h$ 로 표현한다면, 원래의 이미지는  $f = g * h^{-1}$ 의 관계를 이용하여 구할 수 있다. 예를 들어 이미지에 적용된 가우시안 필터에 관한 정확한 정보가 있다면, blurred된 이미지로부터 원래의 이미지를 복원하는 것이 가능하다. 하지만 일반적인 경우에는 훨씬 복잡하고 다양한 형태의 노이즈가 가미되므로 화질의 저하를 가져온 필터의 정확한 정보를 구하는 것은 거의 불가능하다고 할 수 있다. 따라서, 여러가지 필터를 이용하여 얻어진 결과를 정성적으로 평가함으로써 Deblurring을 수행하는 것이 일반적인 방법이다. Deblurring의 대표적인 방법으로는 Wiener 필터가 있다.

## 기타 이미지 필터링 및 화질 향상 기법

이미지에서 윤곽선을 검출하는 에지 검출(Edge detection) 기법은 앞에서 살펴본 필터링 기법과 유사하게 커널 convolution 방법을 통하여 구현한다. 그림 7은 에지 검출에 많이 쓰이는 Sobel 연산자(커널)을 보여준다. Sobel 연산자를 사용하여 구한 X, Y 방향의 Gradient(변화량)를 각각  $G_x$ ,  $G_y$  라고 할 때, 이미지의 Gradient Magnitude는 공식 4와 같이 구한다.

$$G = \sqrt{G_x^2 + G_y^2}$$

식 4. Gradient Magnitude

Gradient Magnitude를 구한 후에는 임의의 Threshold 값을 기준으로 강한 Gradient 값을 나타내는 픽셀들을 윤곽선으로 간주한다.

그림 8은 Sobel 연산자를 이용한 필터링과 기타 추가적인 처리 과정(Canny edge detector)을 거친 에지 검출 결과를 보여준다. 에지 영상은 이미지에서 객체를 검출할 때 이용될 수 있으며 특히 선형이나 원형 도형의 물체를 검출할 때 쓰이는 Hough transformation의 중요한 전처리 방법이다.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

그림 7. 에지 검출에 쓰이는 Sobel 연산자



그림 8. Canny edge detector를 이용한 에지 검출의 예.

그 밖에, 사용하는 이미지의 특성에 따라 다양한 필터링 기법을 생각할 수 있다. 예를 들어, 비디오인 경우, 여러 프레임을 시간 축 방향으로 더함으로써 노이즈를 상쇄할 수 있으며, 관찰하고

자 하는 영역이 움직이고 그 외의 영역은 정지해 있다면 동적인 부분만을 검출하여 영상 정보의 유용성을 높일 수 있다.

## 이미지 필터링 프로그램 구현

---

이번 실습에서는 Average 필터링, Median 필터링 및 에지 검출에 관한 프로그램을 작성해 본다. .

### 1) Average 필터링

(a) 입력된 이미지를 3×3, 5×5, 9×9 사이즈의 uniform 커널을 이용하여 Average 필터링을 수행하고 결과를 화면에 출력한다. 커널 사이즈가 커짐에 따라 이미지의 blurring 효과가 증가하는지 관찰한다.

(b) 위에서 행한 연산을 Moving Average 기법을 이용하여 수행한다. 필터링 시간을 측정하여 (a) 에서 걸린 시간과 비교해 본다. 각 커널 사이즈 별로 시간을 측정해 보자. Moving Average 필터링을 구현하는 방법은 커널의 이동 방법에 따라서 여러 가지가 있을 수 있지만, 어떤 방식으로 구현되었든 Average 필터링에 비해 실행 속도가 현저히 빨라야 한다.

### 2) Median 필터링

입력된 이미지를 3×3, 5×5, 9×9 사이즈의 윈도우를 이용하여 median 필터링을 수행하고 결과를 화면에 출력한다. 각 커널 사이즈 별로 연산 시간을 측정해 보고, 1) 번에서 걸린 시간과 비교해 보자.

### 3) 에지 검출

Sobel 연산자를 이용하여 입력된 이미지의 에지를 검출하고 결과를 화면에 출력한다.

## 숙제: Average 필터와 Median 필터의 효율적 구현

---

1) Average 필터링을 효율적으로 구현할 수 있는 Moving Average에 대하여 공식을 이용하여 설명해보자.

2) Median 필터링을 효율적으로 구현할 수 있는 방법에 관하여 도식적으로 설명해 보자. 3×3



윈도우 사이즈를 사용하는 경우를 가정하여 구체적으로 설명한다.

## 참고 문헌

---

M. Petrou and C. Petrou, Image Processing: The Fundamentals, 2nd ed., Wiley.

R. Gonzalez, R. E. Woods and S. L. Eddins, Digital Image Processing Using MATLAB, 2nd ed., Gatesmark Publishing.