

Filter

영상처리의 세 가지 기본 연산

점 연산

- 오직 자신의 명암값에 따라 새로운 값을 결정
- 식으로 쓰면
 - 대부분은 $k=1$ (한 장의 영상을 변환)

$$f_{out}(j, i) = t(f_1(j, i), f_2(j, i), \dots, f_k(j, i)) \quad (2.10)$$

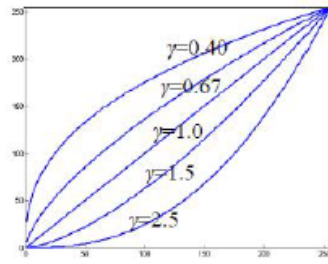
- 선형 연산

$$f_{out}(j, i) = t(f(j, i)) \\ = \begin{cases} \min(f(j, i) + a, L - 1), & \text{(밝게)} \\ \max(f(j, i) - a, 0), & \text{(어둡게)} \\ (L - 1) - f(j, i), & \text{(반전)} \end{cases}$$



- 비선형 연산
 - 예) 감마 수정(모니터나 프린터 색상 조절에 사용)

$$f_{out}(j, i) = (L - 1) \times (\hat{f}(j, i))^\gamma \quad \text{이때} \quad \hat{f}(j, i) = \frac{f(j, i)}{(L - 1)}$$



(a) 감마값에 따른 변환 함수의 모양



(b) $\gamma=0.40$



(c) $\gamma=0.67$



(d) $\gamma=1.0$ (원본 영상)



(e) $\gamma=1.5$



(f) $\gamma=2.5$

그림 2-19 감마수정

- 디졸브
 - $k = 2$ 인 경우

$$f_{out}(j, i) = \alpha f_1(j, i) + (1 - \alpha) f_2(j, i) \quad (2.13)$$



그림 2-21 디졸브 효과

- 알파 값이 1이면 뒷향이 없어지고, 0이면 앞 향이 없어짐
- 0~1 비율을 조절해 두 개의 영상을 결합

영역 연산(Filtering)

- 이웃 화소의 명암값에 따라 새로운 값 결정
- Image filters in spatial domain(영상평면에서의 연산)
 - Filter is a mathematical operation of a grid of numbers
 - smoothing, sharpening, measuring texture
- Image Filtering : compute function of **local neighborhood** at each position
- 매우 중요함
 - Enhance images
 - denoise, resize, increase contrast
 - **Extract information from images**

- texture, edges, distinctive points
- Detect patterns
 - template matching

상관과 컨볼루션

- 상관
 - 원시적인 매칭 연산 (물체를 윈도우 형태라고 표현하고 물체를 검출)
 - 윈도우와 가장 유사한 위치 찾기
 - 최대값 29를 갖는 위치6에서 물체 검출

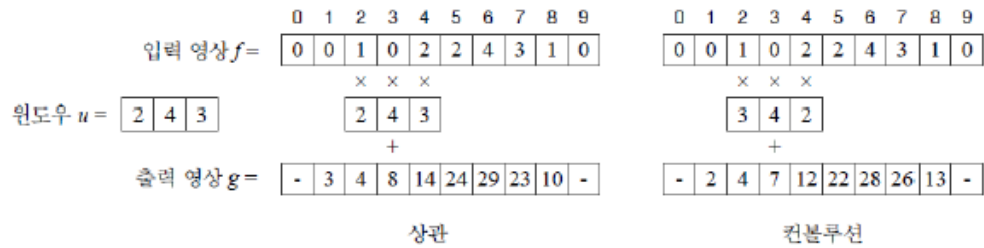


그림 2-22 상관과 컨볼루션의 원리

- 영상은 고차원 데이터(엄청 큼)
 - 차원을 줄이기 위해 컨볼루션 이용, 의미가 있을 것 같은 정보들을 뽑아서 인식
 - 앞부분에 불필요한 차원을 줄이고 의미있는 정보(내가 원하는 정보)만 추출
- 컨볼루션
 - 윈도우를 뒤집은 후 상관 적용
 - 임펄스 반응
- 2차원

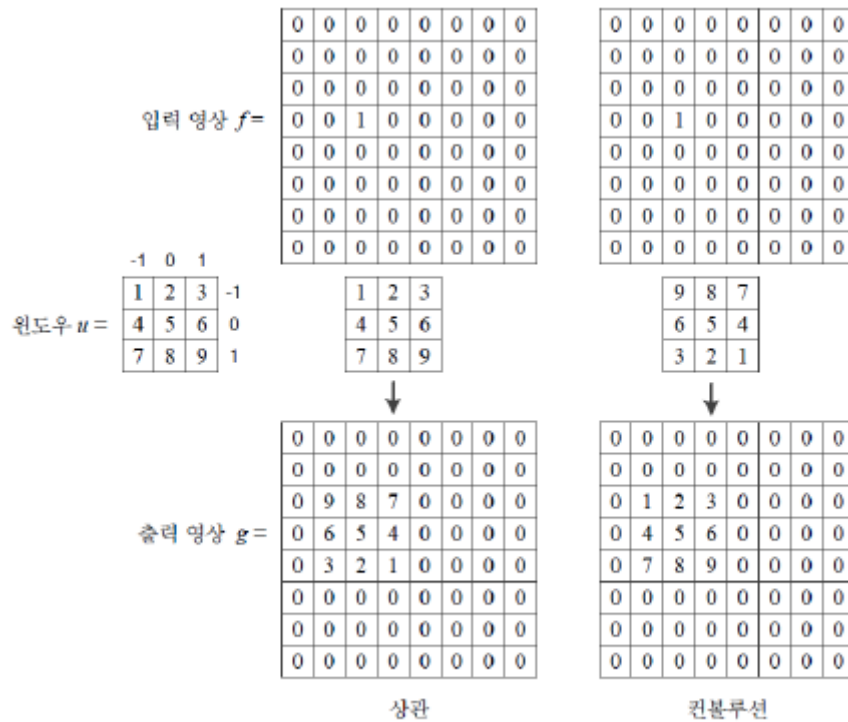


그림 2-23 2차원 상관과 컨볼루션

- 상관 대신 컨볼루션을 하면, 윈도우와 똑같은 형태를 가지도록 아웃풋을 만들 수 있다
- 수식 표현

$$\left. \begin{array}{l} \text{상관 } g(i) = u \otimes f = \sum_{x=-(w-1)/2}^{(w-1)/2} u(x)f(i+x) \\ \text{컨볼루션 } g(i) = u \circledast f = \sum_{j=-(w-1)/2}^{(w-1)/2} u(x)f(i-x) \end{array} \right\} \text{1차원} \quad (2.14)$$

$$\left. \begin{array}{l} \text{상관 } g(j,i) = u \otimes f = \sum_{y=-(h-1)/2}^{(h-1)/2} \sum_{x=-(w-1)/2}^{(w-1)/2} u(y,x)f(j+y,i+x) \\ \text{컨볼루션 } g(j,i) = u \circledast f = \sum_{y=-(h-1)/2}^{(h-1)/2} \sum_{x=-(w-1)/2}^{(w-1)/2} u(y,x)f(j-y,i-x) \end{array} \right\} \text{2차원}$$

- 따로 둘을 구분하지 않고 컨볼루션이라는 용어 사용
- 컨볼루션 예제
 - 박스와 가우시안은 스무딩 효과
 - 샤프닝은 명암 대비 강조
 - 수평 에지와 수직 에지는 에지검출 효과



(a) 원래 영상과 여러 가지 마스크들



(b) 다양한 마스크로 컨볼루션한 영상들

그림 2-24 다양한 마스크와 컨볼루션 효과

- 컨볼루션은 선형 연산
 - 주요 특성
 - Linearity : $\text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2)$
 - Shift Invariance : Same behavior regardless of pixel location
 - $\text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f))$
 - 모든 linear, shift invariant 연산자가 컨볼루션이라고 볼 수 있음

○ 다른 특성

- Commutative : $a * b = b * a$
- Associative : $a * (b * c) = (a * b) * c$
- 분배법칙
- Scalars factor out
- Identity

박스 필터 예제

- 각 픽셀을 주변 픽셀의 평균값으로 대체
- 스무딩 효과

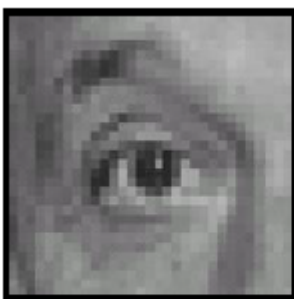
$$f[.,.] \quad g[.,.] \quad \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l] f[m+k,n+l]$$

- 박스 필터의 설정에 따라 이미지 필터링 효과



Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)



Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

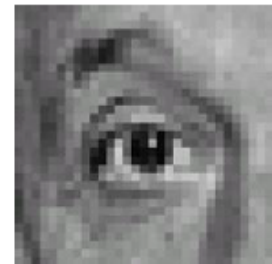


Original

0	0	0
0	2	0
0	0	0

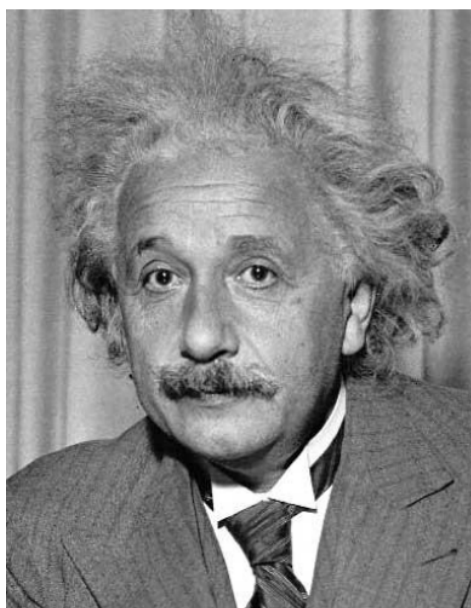
— $\frac{1}{9}$

1	1	1
1	1	1
1	1	1



Sharpening filter

- Accentuates differences with local average



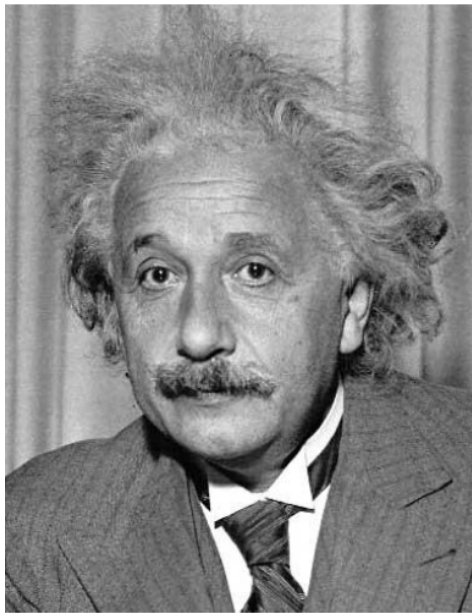
1	0	-1
2	0	-2
1	0	-1

Sobel



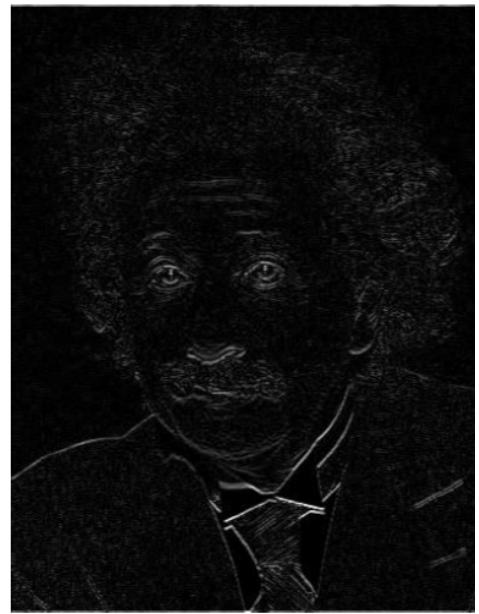
Vertical Edge
(absolute value)

- 좌우의 명암차가 큰 영역만 뽑기



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

Sobel

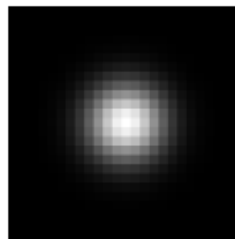
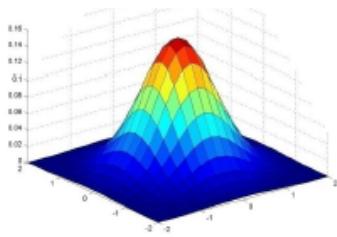


Horizontal Edge
(absolute value)

- 상하의 명암차가 큰 영역만 뽑기

중요필터 : 가우시안

- Weight contributions of neighboring pixels by nearness



0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

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

- 박스필터보다 스무스하게 느껴짐
- Remove "high-frequency" components from the image(low-pass filter)
 - Images become more **smooth**
- Convolution with self is another gaussian
- Separable Kernel(**Seperability**)
 - The 2D Gaussian can be expressed as the product of two functions, one a function of x and the other a function of y
 - 각각의 Function은 1D Gaussian

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

$$= \left(\frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}} \right) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{y^2}{2\sigma^2}} \right)$$

◦ Example

2D convolution
(center location only)

1	2	1
2	4	2
1	2	1

 \ast

2	3	3
3	5	5
4	4	6

The filter factors
into a product of 1D
filters:

1	2	1
2	4	2
1	2	1

 $=$

1
2
1

 \times

1	2	1
---	---	---

Perform convolution
along rows:

1	2	1
---	---	---

 \star

2	3	3
3	5	5
4	4	6

 $=$

	11	
	18	
	18	

Followed by convolution
along the remaining column:

- 2차원을 한꺼번에 하는건 복잡하니 1차원씩 나눠서 할 수 있다
- tensor(3차원) 역시 마찬가지로 쪼갤 수 있다 = tensor decomposition

비선형 연산

- 메디안 필터
 - 솔트페퍼 잡음(하얀색, 검은색 잡음)
 - 가우시안에 비해 에지 보존 효과가 뛰어남



(a) 원래 영상



(b) 솔트페퍼 잡음

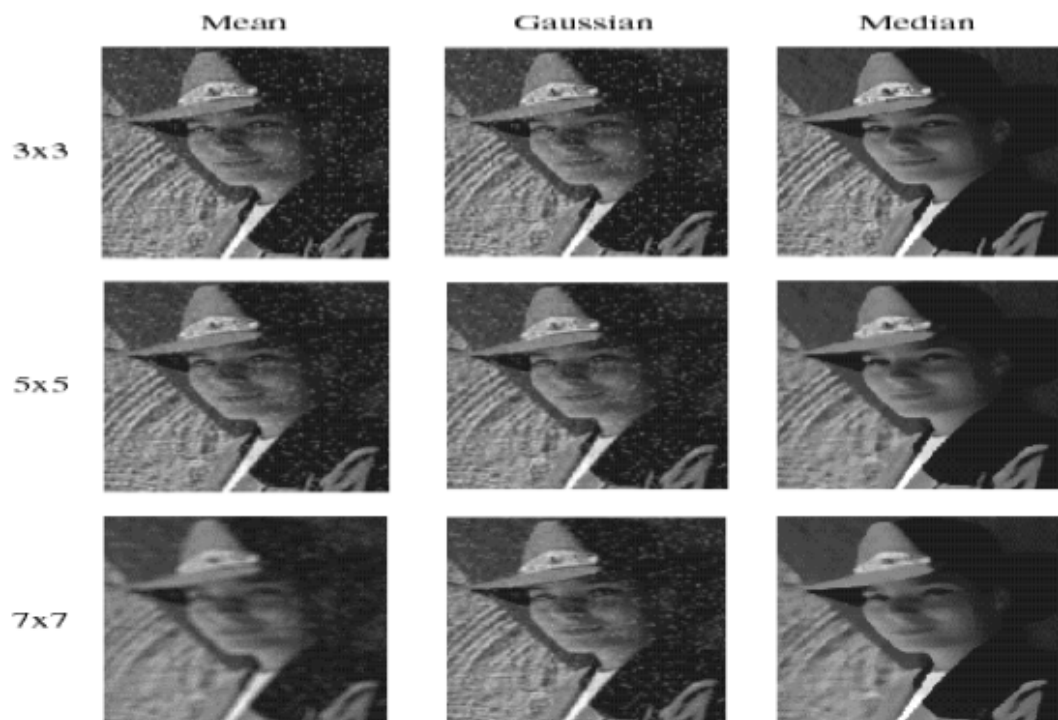


(c) 가우시안 필터



(d) 메디안 필터

그림 2-25 가우시안과 메디안 필터의 비교



- 스무딩 + 잡음 제거 유용

기하 연산

- 일정한 기하 연산으로 결정된 화소의 명암값에 따라 새로운 값 결정