컴퓨터 비전

동양미래대학교
인공지능소프트웨어학과
백찬은 교수

영상 처리

01

디지털 영상

02

이진 영상

03

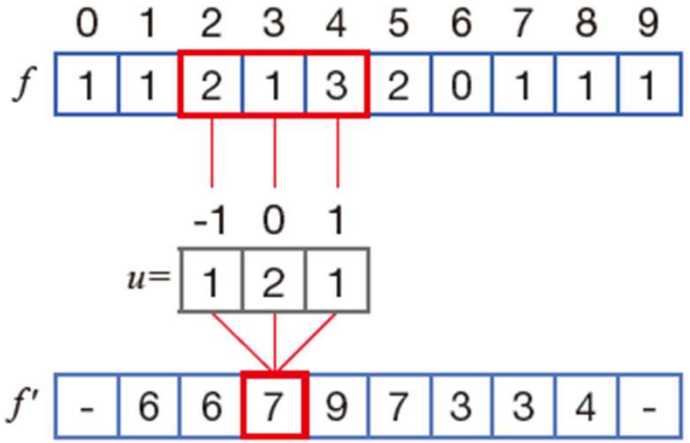
점 연산, 영역 연산, 기하 연산

3 PART

점 연산, 영역 연산, 기하 연산

영역 연산 - 컨볼루션

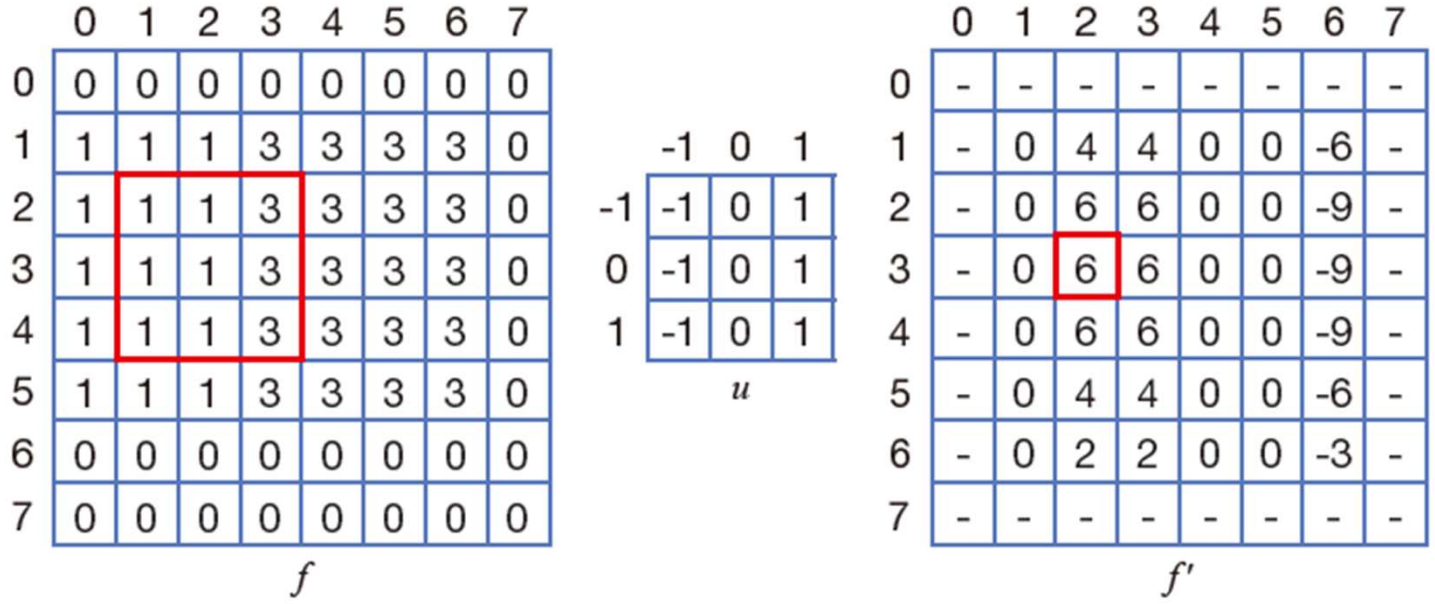
- **영역 연산**: 이웃 화소를 같이 고려해 새로운 값을 결정함
  - 주로 컨볼루션 연산을 통해 이루어짐



(a) 1차원 영상에 컨볼루션 적용

그림 3-16 컨볼루션의 원리

영역 연산 - 컨볼루션



	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	1	1	1	3	3	3	3	0
2	1	1	1	3	3	3	3	0
3	1	1	1	3	3	3	3	0
4	1	1	1	3	3	3	3	0
5	1	1	1	3	3	3	3	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

f

	-1	0	1
-1	-1	0	1
0	-1	0	1
1	-1	0	1

u

	0	1	2	3	4	5	6	7
0	-	-	-	-	-	-	-	-
1	-	0	4	4	0	0	-6	-
2	-	0	6	6	0	0	-9	-
3	-	0	6	6	0	0	-9	-
4	-	0	6	6	0	0	-9	-
5	-	0	4	4	0	0	-6	-
6	-	0	2	2	0	0	-3	-
7	-	-	-	-	-	-	-	-

f'

(b) 2차원 영상에 컨볼루션 적용

컨볼루션 연산 적용시 이미지의 크기가 줄어드는 것을 방지하기 위해, padding을 적용함

- **zero padding:** 필요한만큼 이미지의 가장자리를 확장한 후 0으로 채움
- **copy padding:** 필요한만큼 이미지의 가장자리를 확장한 후 가장자리와 같은 화소값으로 채움

Spatial Filtering

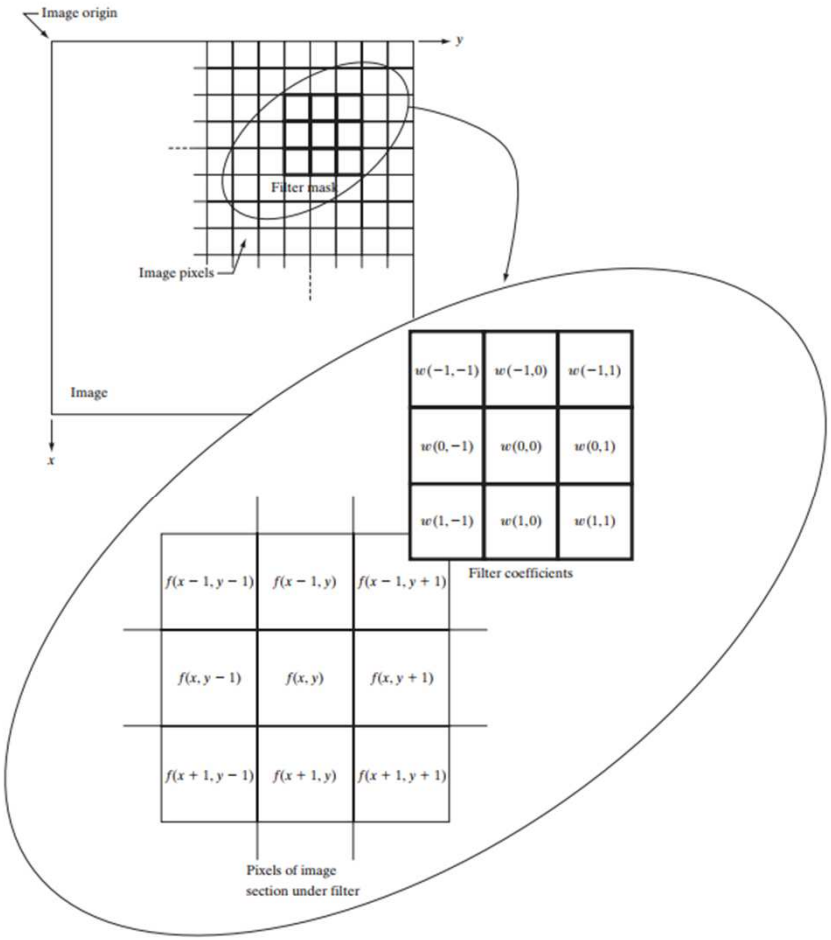
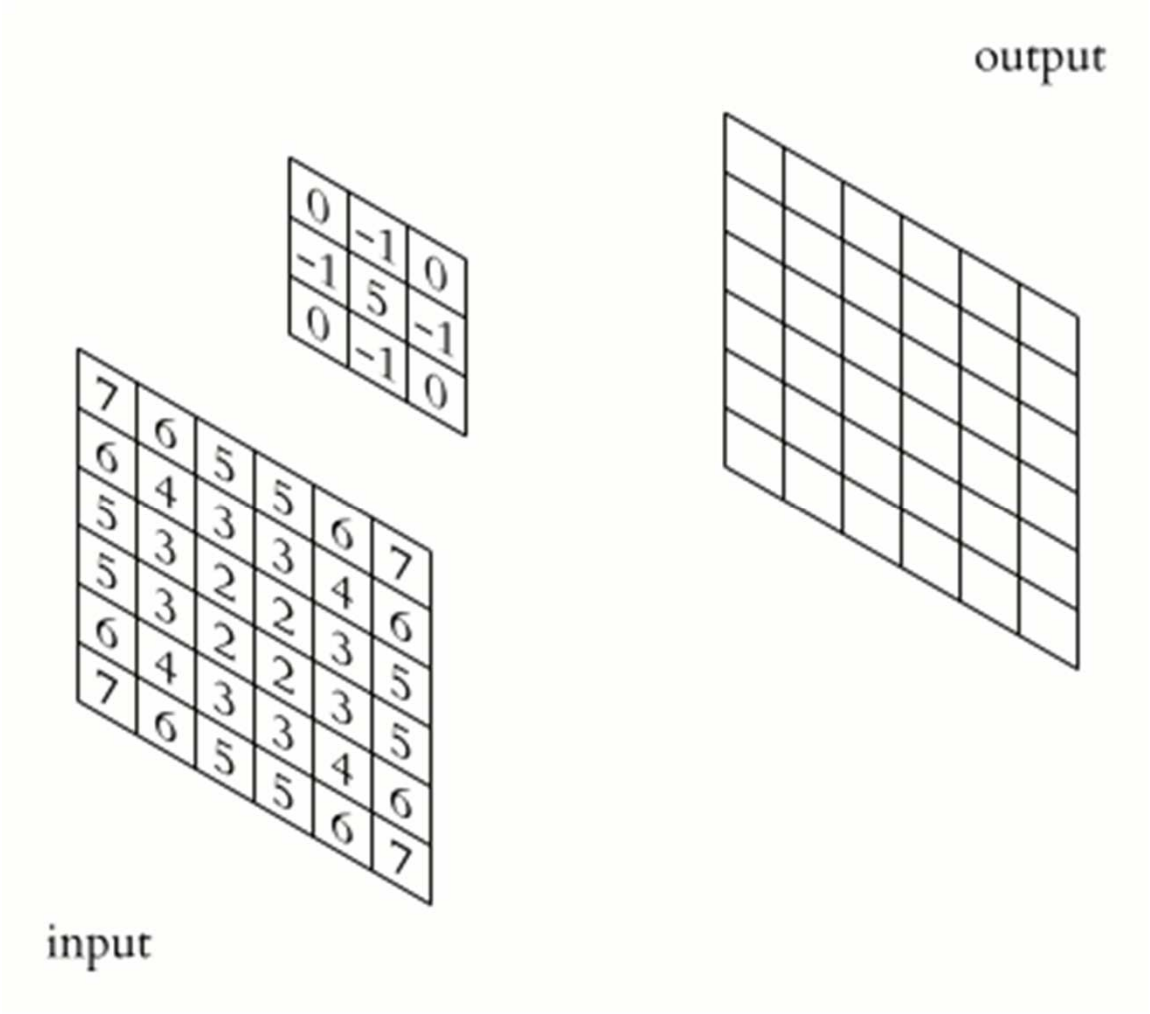


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

		이웃들..
	현재 픽셀	

Spatial Filtering

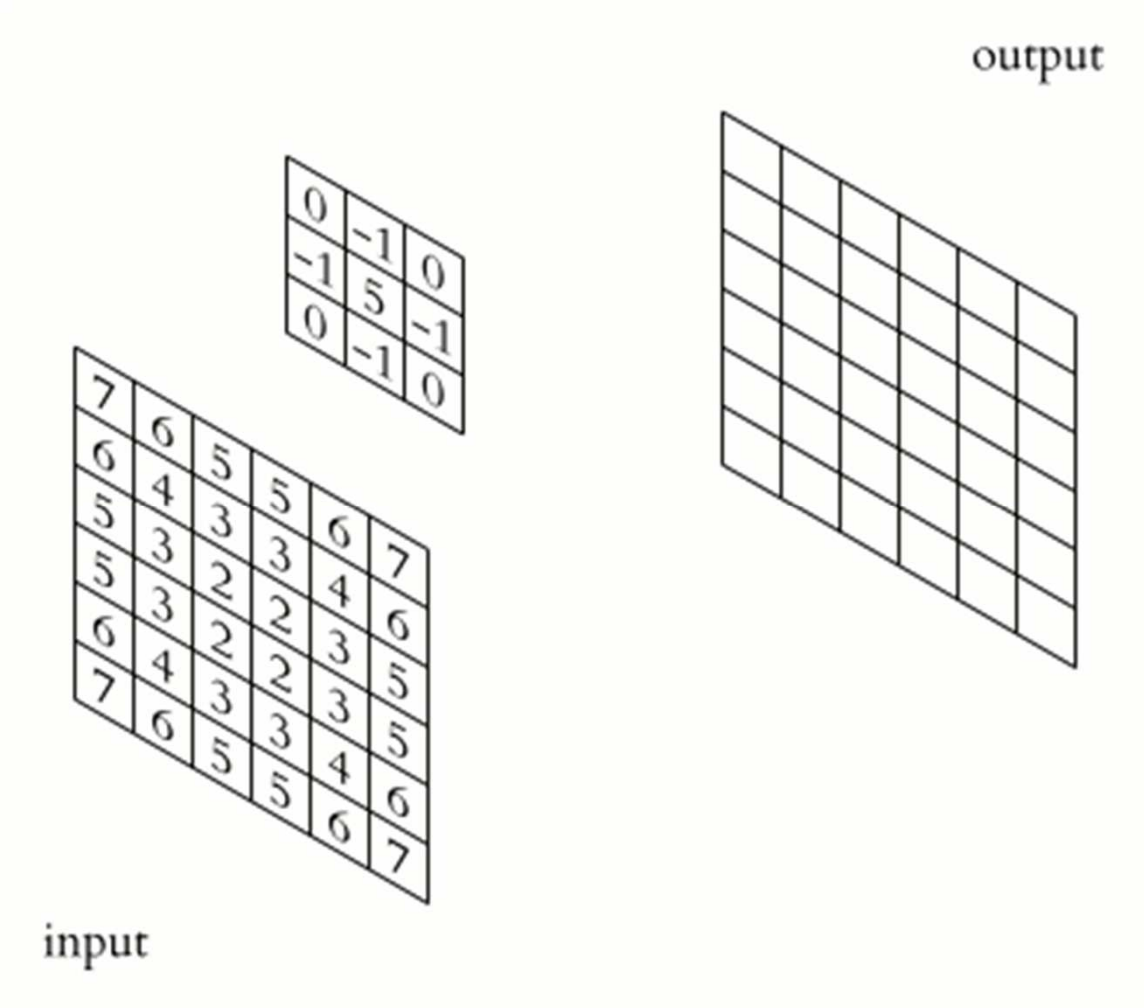


padding 방식들

7	6	5
6	4	3
5	3	2

7	6	5
6	4	3
5	3	2

Spatial Filtering



padding 방식들

0	0	0	0
0	7	6	5
0	6	4	3
0	5	3	2

7	7	6	5
7	7	6	5
6	6	4	3
5	5	3	2

영역 연산 - 다양한 필터

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

(b) 샤프닝 필터

영역 연산 - 다양한 필터

Sharpening filter

Original image

Sharpen filter applied

영역 연산 - 다양한 필터

Smoothing filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

영역 연산 - 다양한 필터

Smoothing filter - Averaging filter

normalize

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

각 픽셀들의 영향력이 동일함

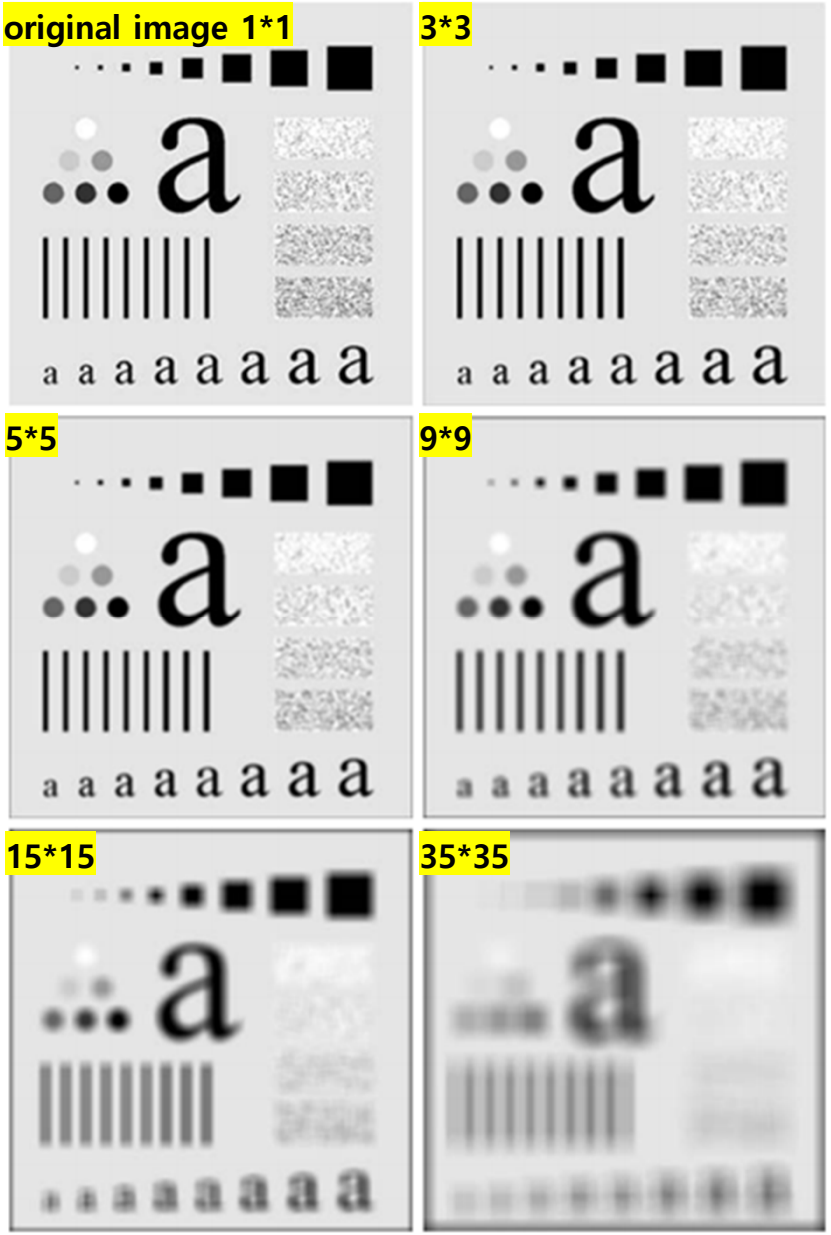
normalize

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

중심 픽셀로부터의 거리에 따라 영향력이 다르게 작용

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.



영역 연산 - 다양한 필터

averaging filter 크기가 클수록

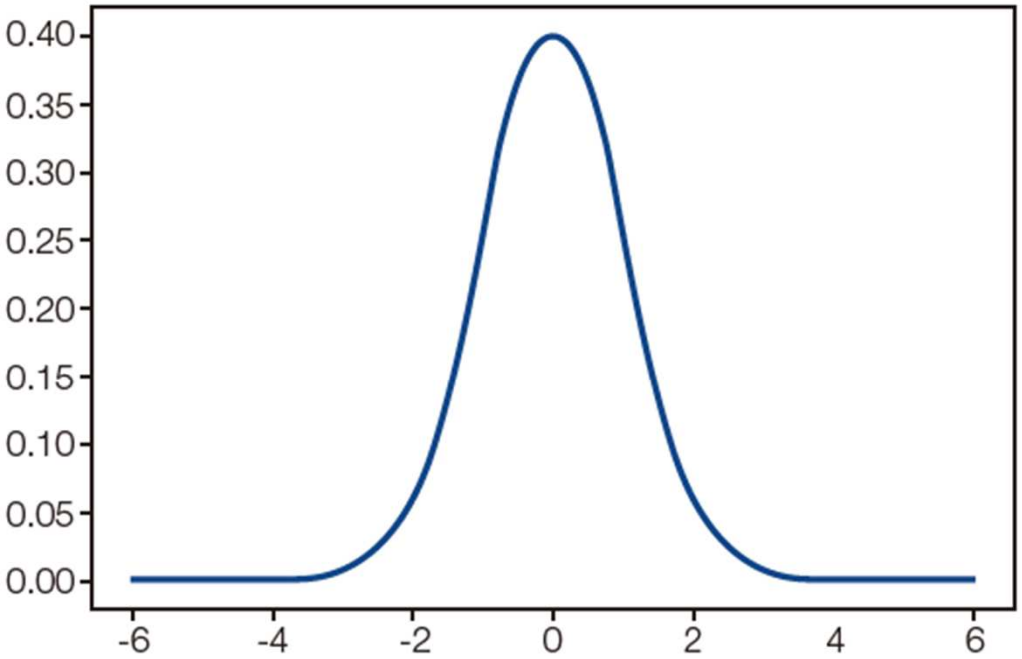
더 흐려지는 이유는? (blurring effect)

FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f

영역 연산 - 다양한 필터

Smoothing filter - Gaussian filter



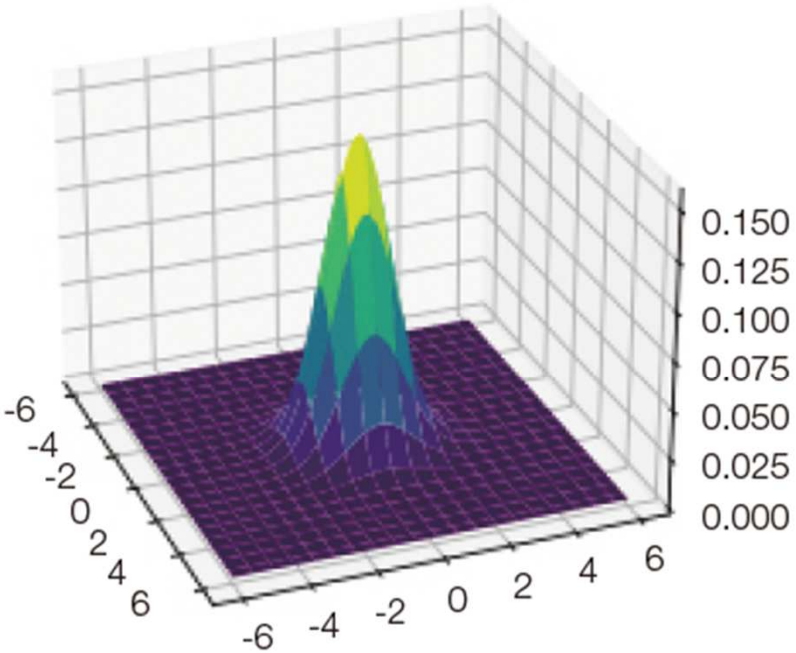
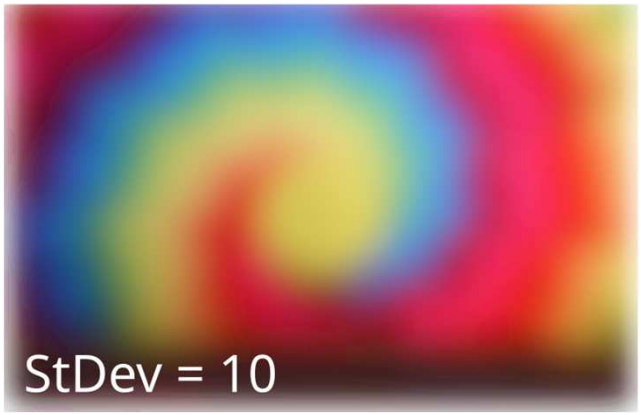


그림 3-18 1차원과 2차원 가우시안 함수

영역 연산 - 다양한 필터

Smoothing filter - Gaussian filter



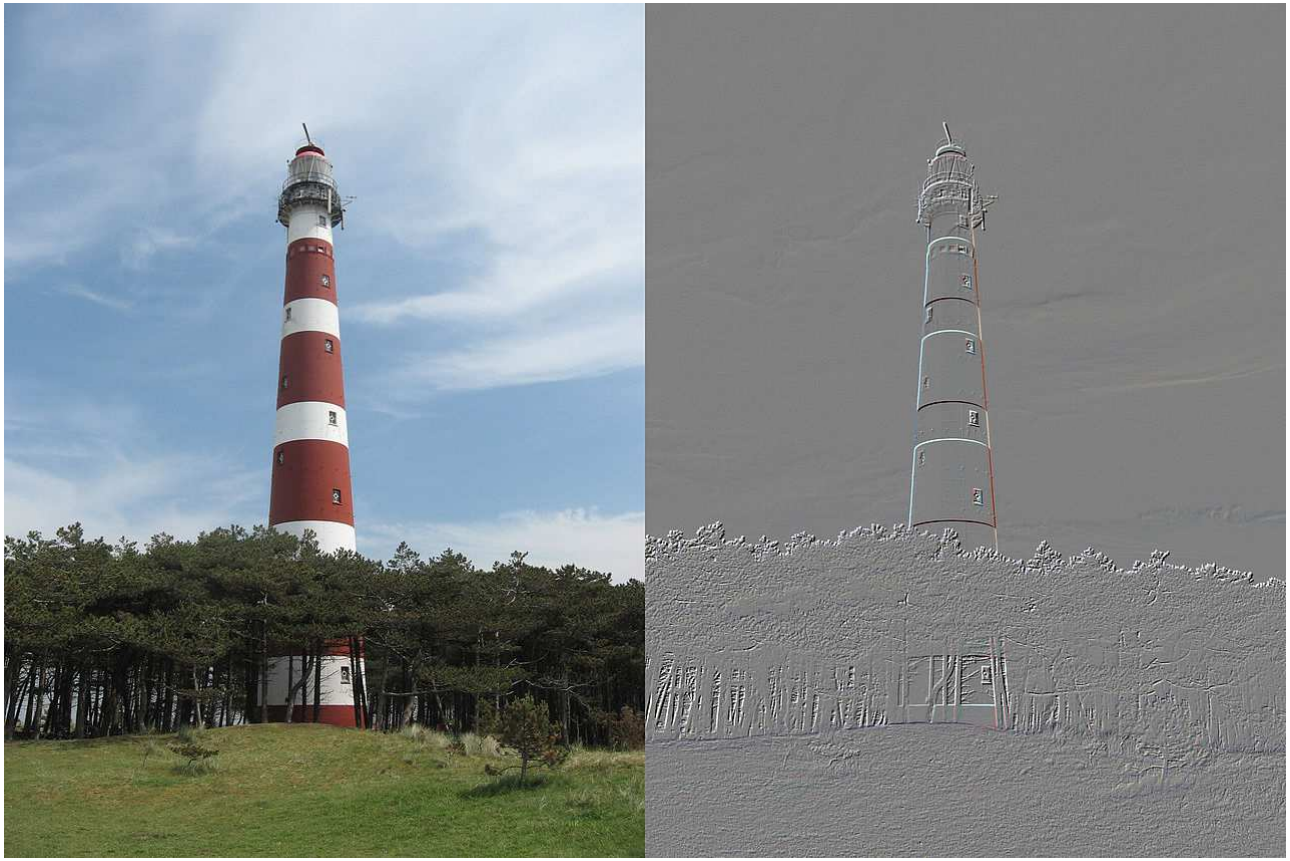
https://en.wikipedia.org/wiki/Gaussian_blur

영역 연산 - 다양한 필터

-1	0	0
0	0	0
0	0	1

-1	-1	0
-1	0	1
0	1	1

(c) 엠보싱 필터



https://en.wikipedia.org/wiki/Image_embossing

영역 연산 - 다양한 필터

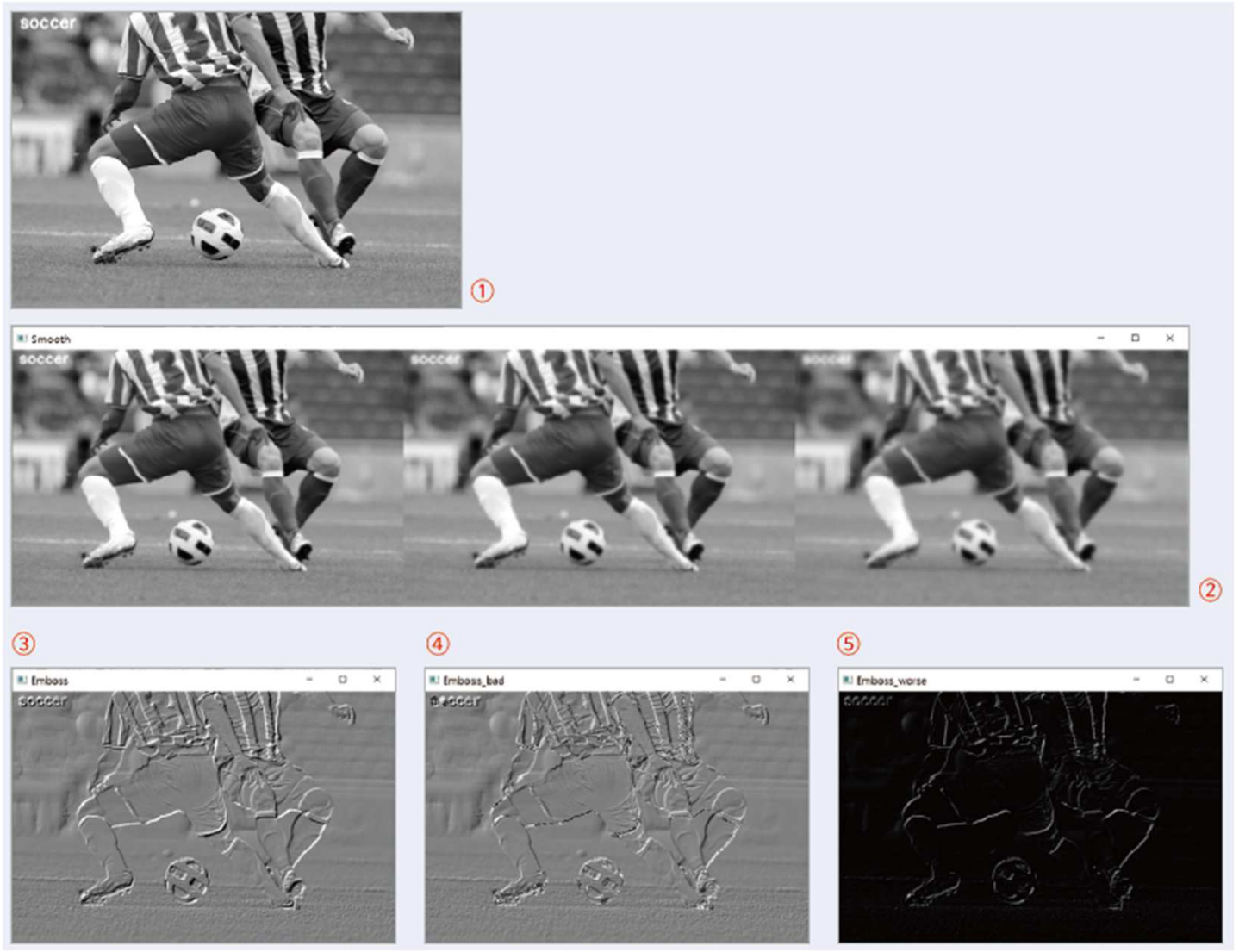
- 필터와 컨볼루션 연산을 통해 각 화소값이 0보다 작거나 255보다 크게 바뀌는 경우가 생길 수 있음.
- 이때 0 미만은 0으로, 255 초과는 255로 변환되기에 이미지의 데이터 손실이 발생함
- 연산 결과가 범위를 벗어나게 되면 clip 함수, rescale 함수 등을 활용해 처리할 수 있음

(실습) 컨볼루션 적용

프로그램 3-7 컨볼루션 적용(가우시안 스무딩과 엠보싱)하기

```
01  import cv2 as cv
02  import numpy as np
03
04  img=cv.imread('soccer.jpg')
05  img=cv.resize(img,dsize=(0,0),fx=0.4,fy=0.4)
06  gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)
07  cv.putText(gray,'soccer',(10,20),cv.FONT_HERSHEY_SIMPLEX,0.7,(255,255,255),2)
08  cv.imshow('Original',gray) ①
09
10  smooth=np.hstack((cv.GaussianBlur(gray,(5,5),0.0),cv.
                  GaussianBlur(gray,(9,9),0.0),cv.GaussianBlur(gray,(15,15),0.0)))
11  cv.imshow('Smooth',smooth) ②
12
13  femboss=np.array([[-1.0, 0.0, 0.0],
14                    [ 0.0, 0.0, 0.0],
15                    [ 0.0, 0.0, 1.0]])
16
17  gray16=np.int16(gray)
18  emboss=np.uint8(np.clip(cv.filter2D(gray16,-1,femboss)+128,0,255))
19  emboss_bad=np.uint8(cv.filter2D(gray16,-1,femboss)+128)
20  emboss_worse=cv.filter2D(gray,-1,femboss)
21
22  cv.imshow('Emboss',emboss) ③
23  cv.imshow('Emboss_bad',emboss_bad) ④
24  cv.imshow('Emboss_worse',emboss_worse) ⑤
25
26  cv.waitKey()
27  cv.destroyAllWindows()
```

(실습) 컨볼루션 적용



기하 연산 - 동차 좌표와 동차 행렬

- **기하 연산** - 영상 확대/축소, 회전, 이동 등

- **동차 좌표(homogeneous coordinate)**
- 2차원 좌표에 1을 추가해 3차원 벡터로 표현한 것
- 3개 요소에 같은 w값을 곱하면 같은 점을 나타내는 것임

  (예) (-2,4,1)과 (-4,8,2)는 (-2,4)에 해당

$$\overline{p} = (x, y, 1)$$

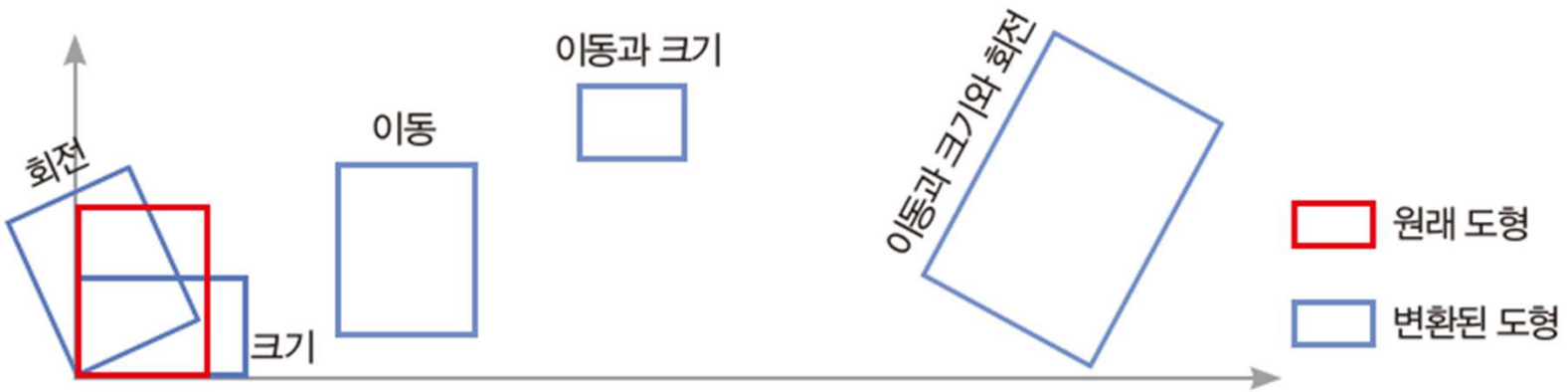


그림 3-19 여러 가지 기하 변환

기하 연산 - 동차 좌표와 동차 행렬

- **동차 행렬(homogeneous matrix)**
- 동차 좌표를 변환할 때 사용하는 행렬
- 이동, 회전, 스케일링 등을 모두 3*3 행렬로 나타낼 수 있음
- 동차 행렬을 동차 좌표에 곱하면 이동된 좌표가 계산됨

- **어파인 변환(Affine Transformation)**
- 길이, 각도, 면적은 변할 수 있지만, 비율과 평행성(평행하는 것은 변환 후에도 평행)은 유지되는 기하 변환
- (예) 평행 이동, 회전, 크기 변환, 전단(shear, 기울이기)

기하 연산 - 동차 좌표와 동차 행렬

표 3-1 3가지 기하 변환

기하 변환	동차 행렬	설명
이동	$\mathrm{T}(t_x, t_y) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$	x 방향으로 t_x, y 방향으로 t_y만큼 이동
회전	$\mathrm{R}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	원점을 중심으로 반시계 방향으로 θ만큼 회전
크기	$\mathrm{S}(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$	x 방향으로 s_x, y 방향으로 s_x만큼 크기 조정(1보다 크면 확대, 1보다 작으면 축소)

기하 연산 - 동차 좌표와 동차 행렬

- 정사각형을 x 방향으로 2, y 방향으로 -1만큼 이동한 다음 반시계 방향으로 30도 회전

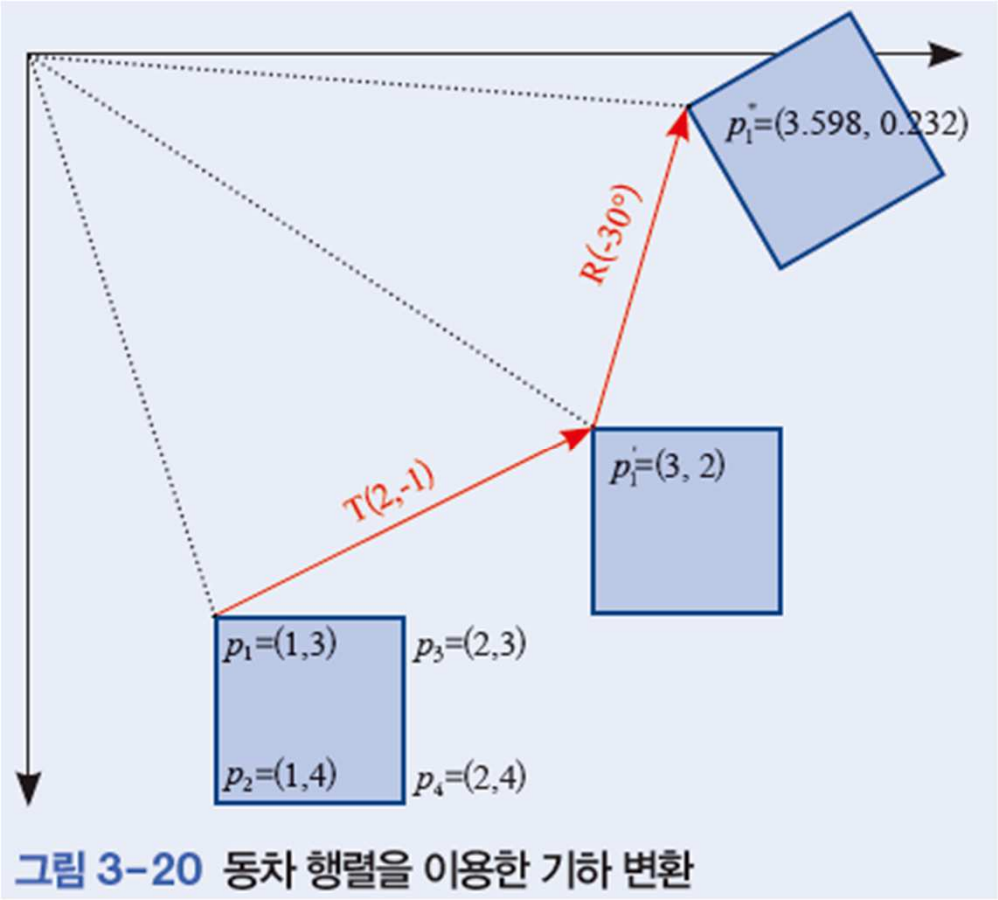


그림 3-20 동차 행렬을 이용한 기하 변환

기하 연산 - 동차 좌표와 동차 행렬

- 변환을 위한 동차 행렬

$$T(2,-1)=\begin{pmatrix}1 & 0 & 2\\ 0 & 1 & -1\\ 0 & 0 & 1\end{pmatrix},\quad R(30°)=\begin{pmatrix}0.8660 & 0.5000 & 0\\ -0.5000 & 0.8660 & 0\\ 0 & 0 & 1\end{pmatrix}$$

- 이동 적용

$$\overline{p}_1'^{T}=T(2,-1)\,\overline{p}_1^{T}=\begin{pmatrix}1 & 0 & 2\\ 0 & 1 & -1\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1\\ 3\\ 1\end{pmatrix}=\begin{pmatrix}3\\ 2\\ 1\end{pmatrix}$$

- 회전 적용

$$\overline{p}_1''^{T}=R(30°)\,\overline{p}_1'^{T}=\begin{pmatrix}0.8660 & 0.5000 & 0\\ -0.5000 & 0.8660 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}3\\ 2\\ 1\end{pmatrix}=\begin{pmatrix}3.598\\ 0.232\\ 1\end{pmatrix}$$

기하 연산 - 보간

- **영상 보간(interpolation)**

: 이미지나 영상에서 픽셀 사이의 값을 추정해 채워 넣는 과정

- **최근접 이웃 보간(Nearest Neighbor Interpolation)**

: 가장 가까운 픽셀 값을 그대로 복사해서 사용함

: 속도가 빠르지만, 계단 현상(aliasing)이 생길 수 있음

- **양선형 보간(Bilinear Interpolation)**

: 주변 4개의 픽셀 값을 선형적으로 가중 평균해 새 픽셀 값을 계산함

: 화질이 부드러워지지만 계산량이 늘어남

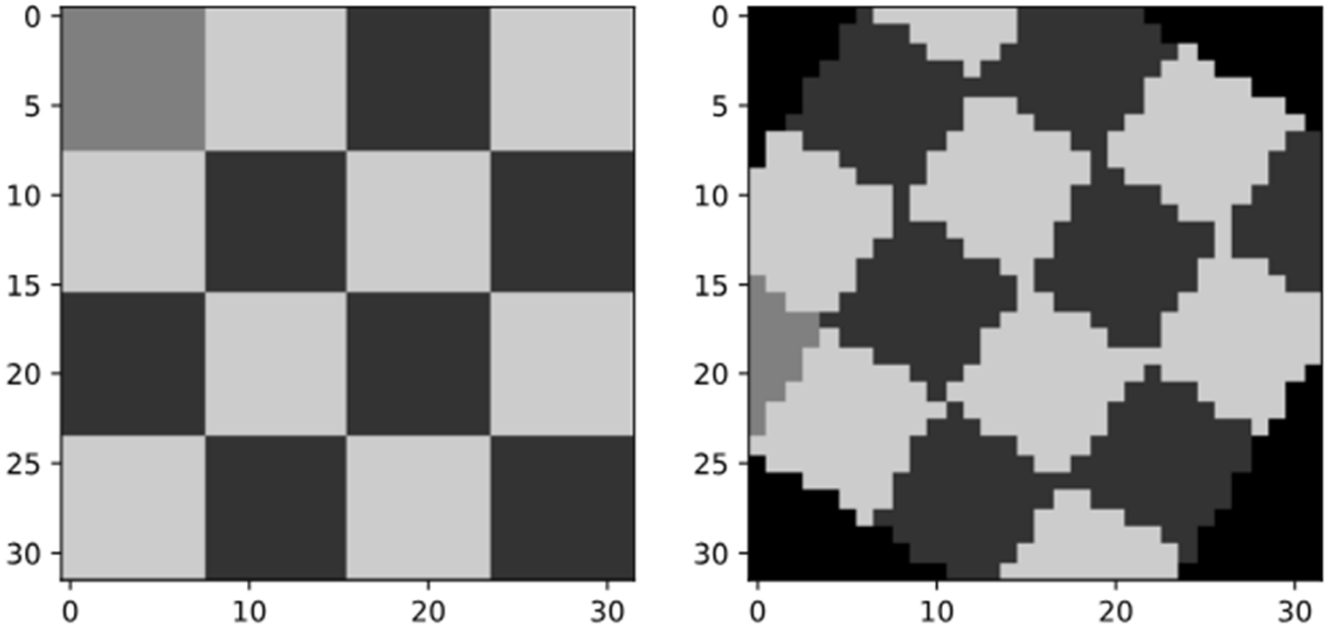
- **스플라인, 고차 보간 등(Spline, Bicubic Interpolation 등)**

: 주변의 더 많은 픽셀을 사용해 곡선 형태로 값을 추정함

: 화질이 가장 좋지만, 계산이 매우 복잡함

기하 연산 - 보간

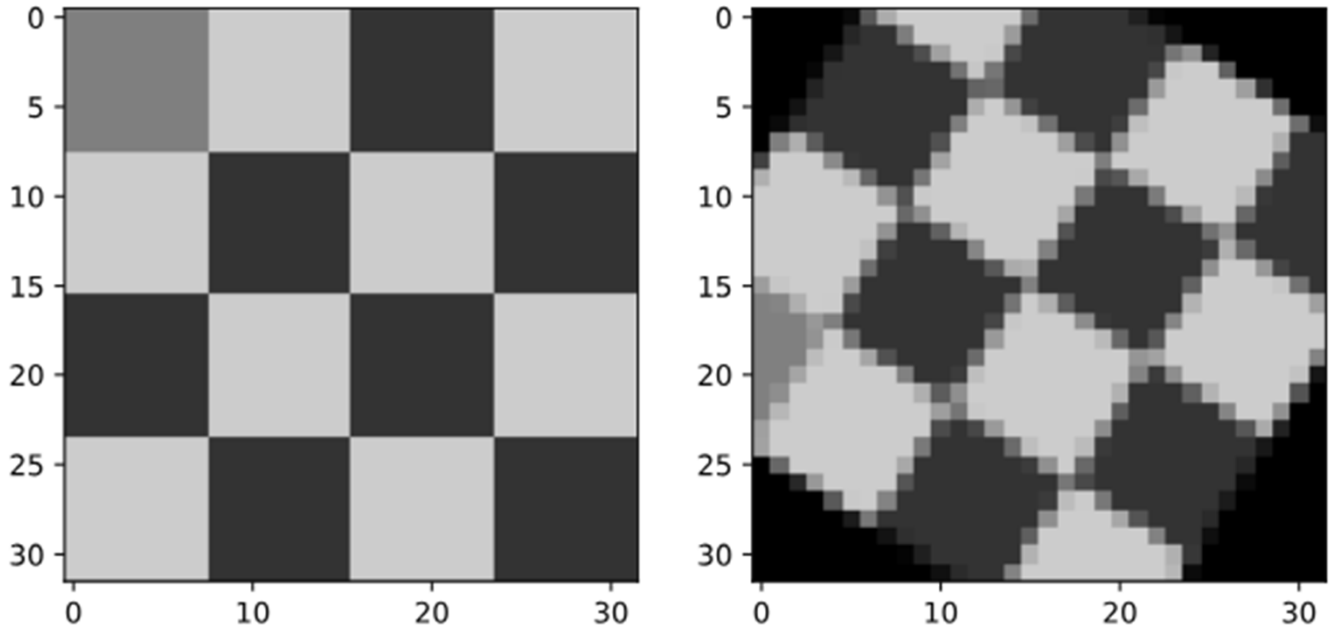
- **최근접 이웃 보간(Nearest Neighbor Interpolation)**



https://vincmazet.github.io/bip/interpolation/interpolation.html

기하 연산 - 보간

- **양선형 보간(Bilinear Interpolation)**



https://vincmazet.github.io/bip/interpolation/interpolation.html

기하 연산 - 보간

- 양선형 보간(Bilinear Interpolation)

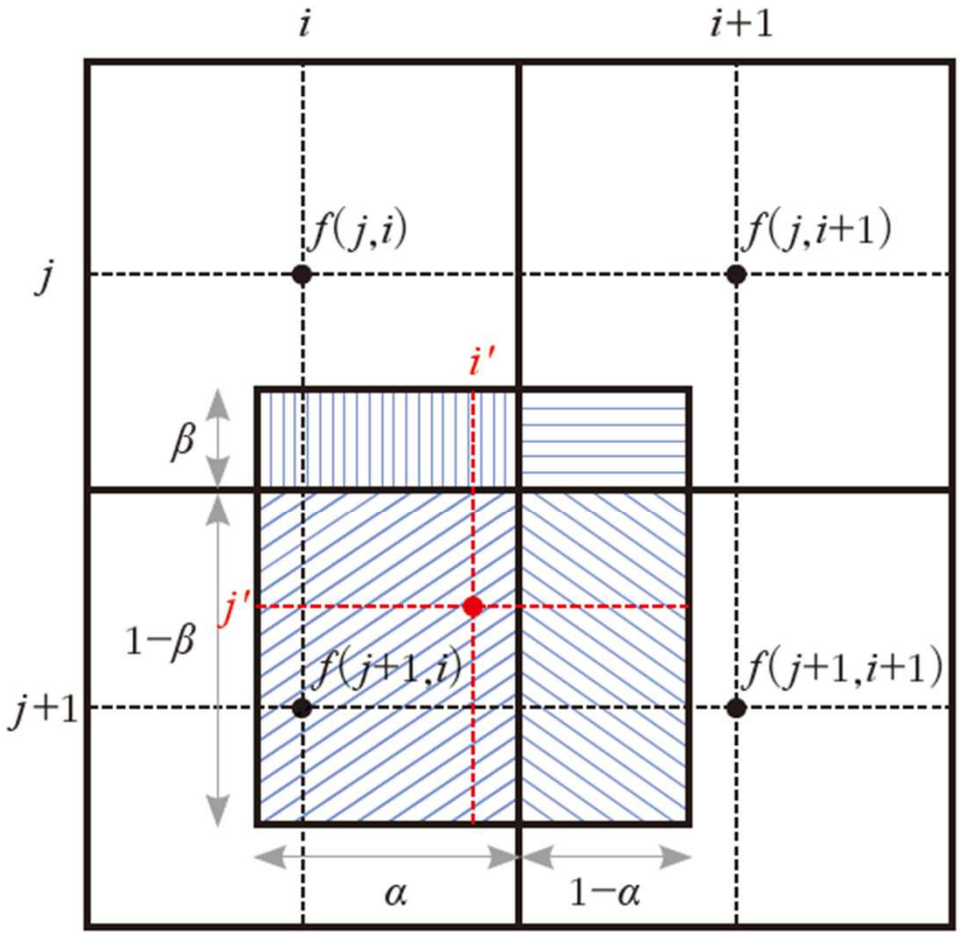
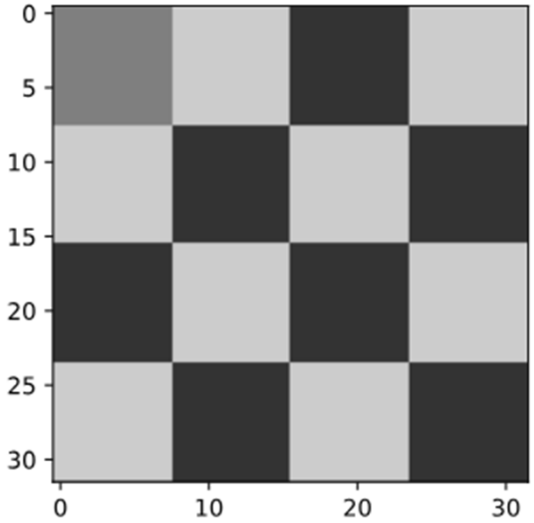
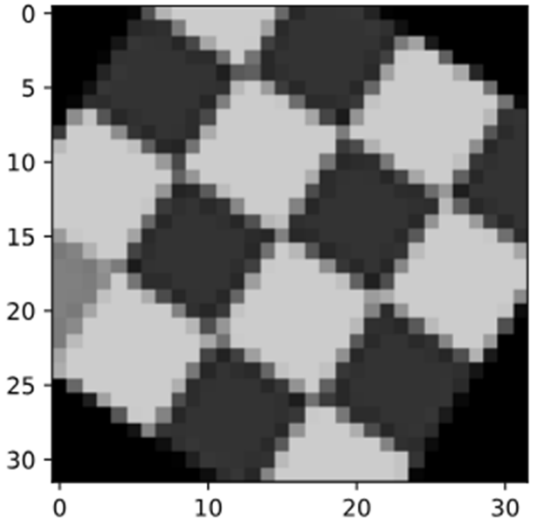


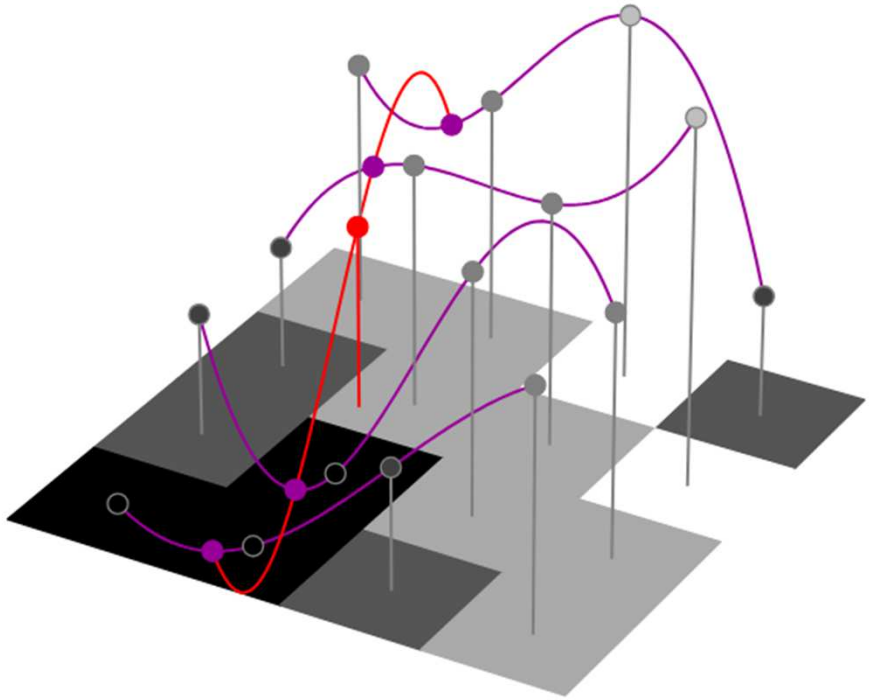
그림 3-22 실수 좌표의 화솟값을 보간하는 과정

기하 연산 - 보간

- **Bicubic Interpolation**







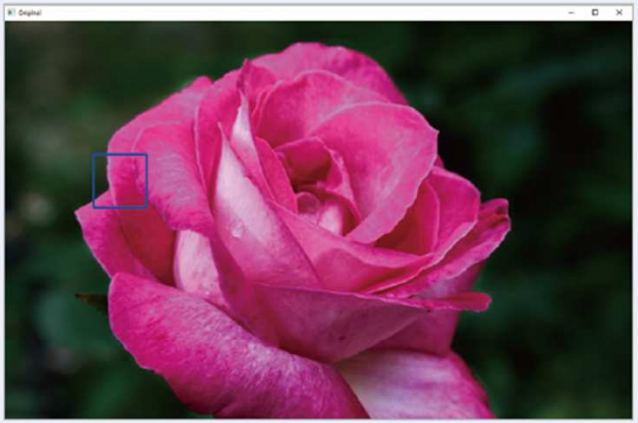
https://vincmazet.github.io/bip/interpolation/interpolation.html

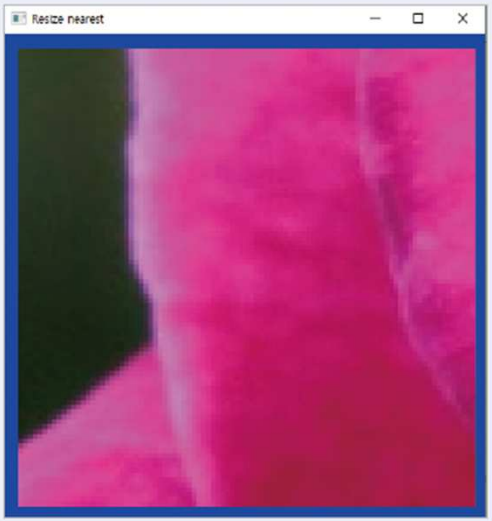
(실습) 기하 연산 - 보간

프로그램 3-8 보간을 이용해 영상의 기하 변환하기

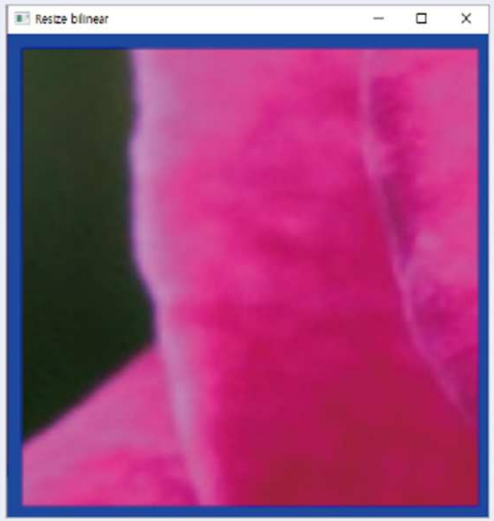
```
01  import cv2 as cv
02
03  img=cv.imread('rose.png')
04  patch=img[250:350,170:270,:]
05
06  img=cv.rectangle(img,(170,250),(270,350),(255,0,0),3)
07  patch1=cv.resize(patch,dsize=(0,0),fx=5,fy=5,interpolation=cv.INTER_NEAREST)
08  patch2=cv.resize(patch,dsize=(0,0),fx=5,fy=5,interpolation=cv.INTER_LINEAR)
09  patch3=cv.resize(patch,dsize=(0,0),fx=5,fy=5,interpolation=cv.INTER_CUBIC)
10
11  cv.imshow('Original',img)
12  cv.imshow('Resize nearest',patch1)
13  cv.imshow('Resize bilinear',patch2)
14  cv.imshow('Resize bicubic',patch3)
15
16  cv.waitKey()
17  cv.destroyAllWindows()
```

(실습) 기하 연산 - 보간

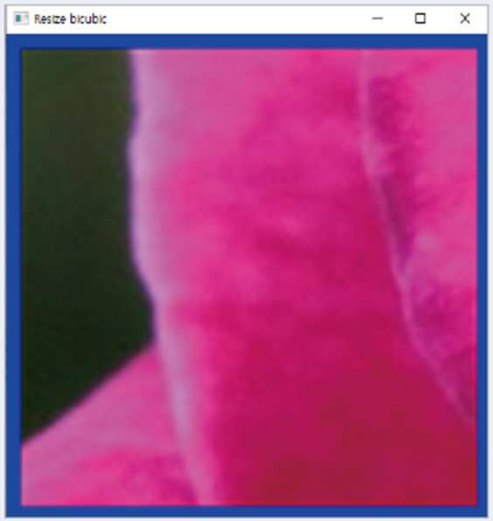




최근접 이웃



양선형 보간



양3차 보간