

# 컴퓨터 비전

**Hanjin Cho**

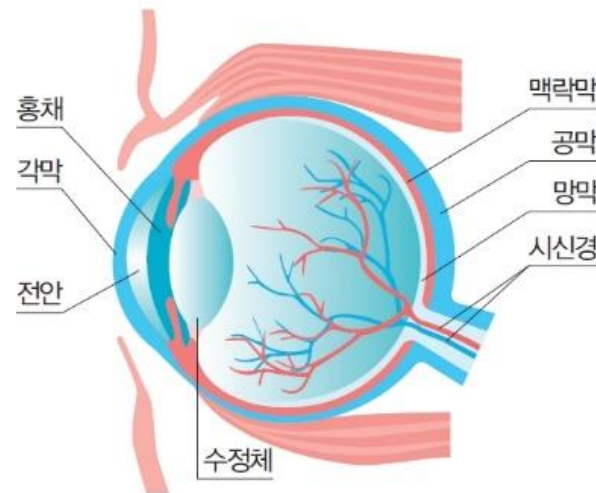


Electronic & Electrical Convergence Engineering  
Hongik University  
Republic of Korea

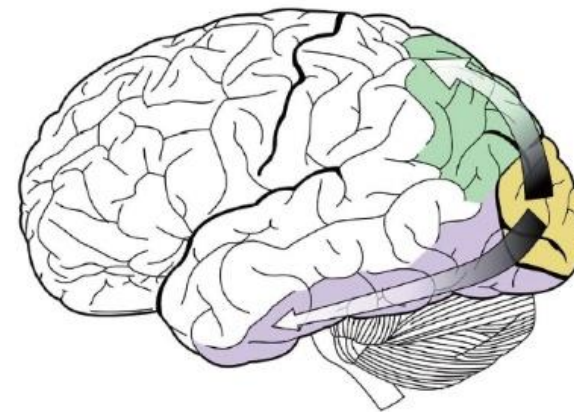
- 인간의 시각
- 컴퓨터 비전
- 디지털 영상
- 영상 처리

## ■ 인간의 시각이 지닌 놀라운 능력

- 인간은 시각 정보를 이용해 인식, 추론, 예측, 상상 등의 고차원적 사고를 수행.
- 이는 인간 시각이 단순한 감각을 넘어 복합적인 정보 처리 능력을 갖고 있음을 보여줌.
- 눈은 빛을 감지하고 시신경을 통해 정보를 대뇌 피질로 전달.
- 뇌에서는 다양한 부위가 협력하여 시각 정보를 종합 처리.



(a) 눈의 구조



(b) 시각 정보 처리를 담당하는 시각 피질

그림 1-2 인간의 시각 시스템

## ■ 시각의 장점

- 분류, 검출, 분할, 추적, 행동 분석 등에 능숙.
- 3차원 공간 정보를 빠르게 획득.
- 처리 속도가 빠르고 결과가 안정적.
- 추론, 협업, 계획 등의 고차원 인지 능력과 결합 가능.
- 환경 변화에 능동적으로 대응하여 유연하게 적응.



그림 1-1 인간이 쉽고 정확하게 해석할 수 있는 영상

## ■ 시각의 한계

- 착시 현상으로 인해 실제와 다르게 인식되는 경우 발생.
  - 예: 그림자, 배경, 대비 효과 등으로 인한 시각적 왜곡.
- 정확한 수치나 정밀한 측정에는 오차가 존재.
- 시야 범위 제한적이며 동시에 볼 수 있는 정보에 한계 있음.
- 장시간 집중 시 피로도 증가 및 인식 능력 저하.

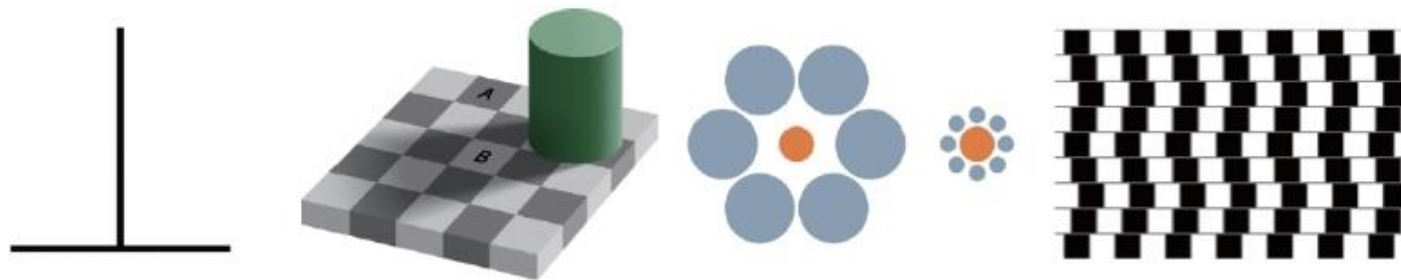
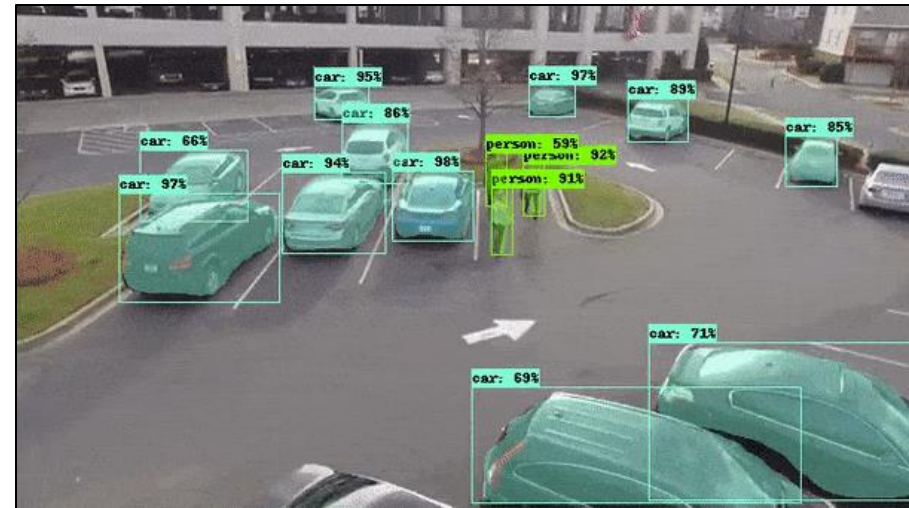
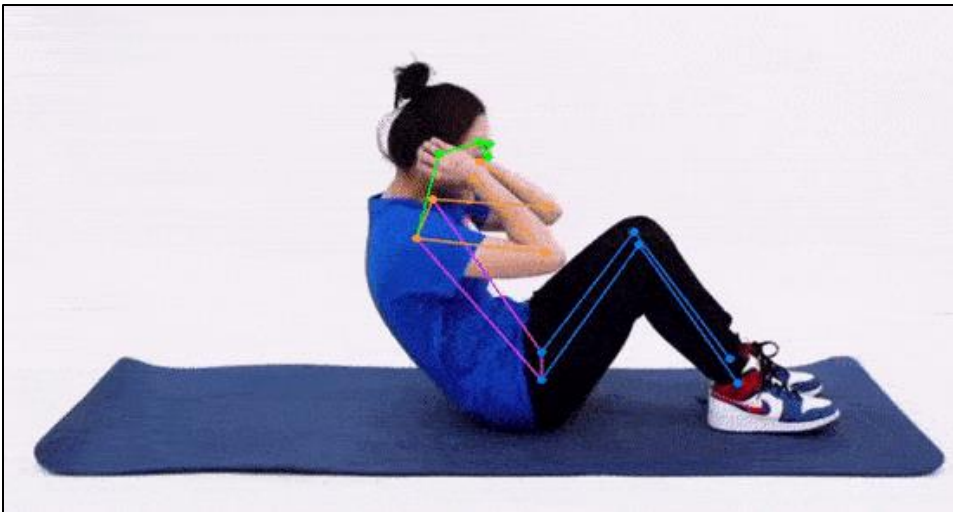


그림 1-3 인간 시각의 착시 현상(출처: 영문 위키피디아 'optical illusion')

## ■ 컴퓨터 비전이란?

- 컴퓨터 비전은 인간의 시각을 모방하는 컴퓨터 프로그램.
- 현재 기술로 인간 수준의 시각을 완벽히 구현하는 것은 불가능하지만 인간 시각의 한계를 극복하는 분야도 많이 존재. 특히, 반복되는 작업으로 한정할 경우 인간 수준에 근접하거나 뛰어넘는 성능 발휘 가능.



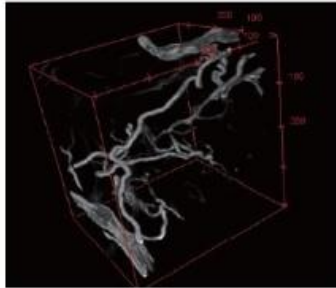


## ■ 대표적인 응용 사례

- 농업, 의료, 교통, 스마트 공장, 스포츠, 유통.
- 보안, 에너지, 엔터테인먼트, 환경, 우주과학, 감시, 예술, 가사, 휴머노이드 로봇.



(a) 과일 수확 드론



(b) 혈관 분할



(c) 자율주행



(d) 불량 검사



(e) 선수의 행동 분석



(f) 고객의 동선 분석



(g) 얼굴 인식 보안



(h) 태양광 모니터링



(i) 게임 플레이(알파스타)



(j) 지형 모니터링



(k) 화성 탐사선



(l) 광장 감시



(m) 에드몽 벨라미



(n) 청소 로봇

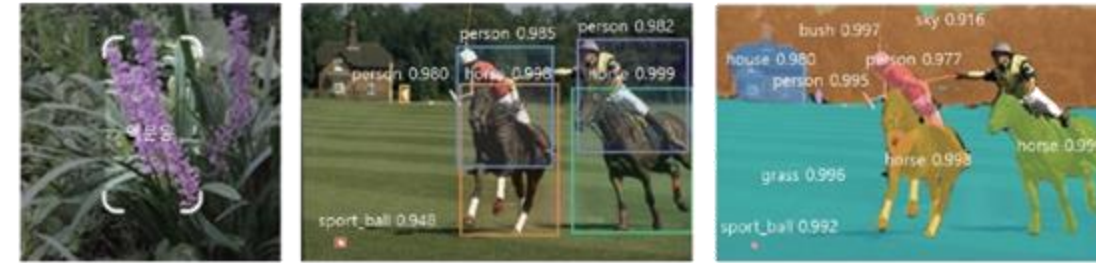


(o) 휴머노이드 로봇

그림 1-5 컴퓨터 비전의 응용 사례

## ■ 컴퓨터 비전의 주요 과제

- 분류(Classification): 객체가 어떤 종류인지 판단.
- 검출(Detection): 이미지나 영상에서 객체의 위치를 찾음.
- 분할(Segmentation): 객체의 픽셀 단위 경계 구분.
- 추적(Tracking): 시간에 따라 객체의 위치를 지속적으로 파악.
- 행동 분석(Activity Recognition): 객체의 행위나 움직임 파악.



(a) 분류

(b) 검출

(c) 분할



(d) 추적(<https://motchallenge.net/vis/MOT17-09-SDP>)



(e) 행동 분석(<https://github.com/mostafa-saad/deep-activity-rec#dataset>)

그림 1-13 컴퓨터 비전이 풀어야 할 문제



## ■ 영상 획득의 기본 원리

- 실제 영상 획득 과정은 복잡하므로 개념 설명을 위해 핀홀 카메라 모델을 사용함.
- 핀홀을 통과한 광선이 상을 맺으며, 이 상이 센서(예: CCD)에 기록됨.
- 카메라의 구조와 영상 형성 원리를 이해하는 데 유용한 모델임.

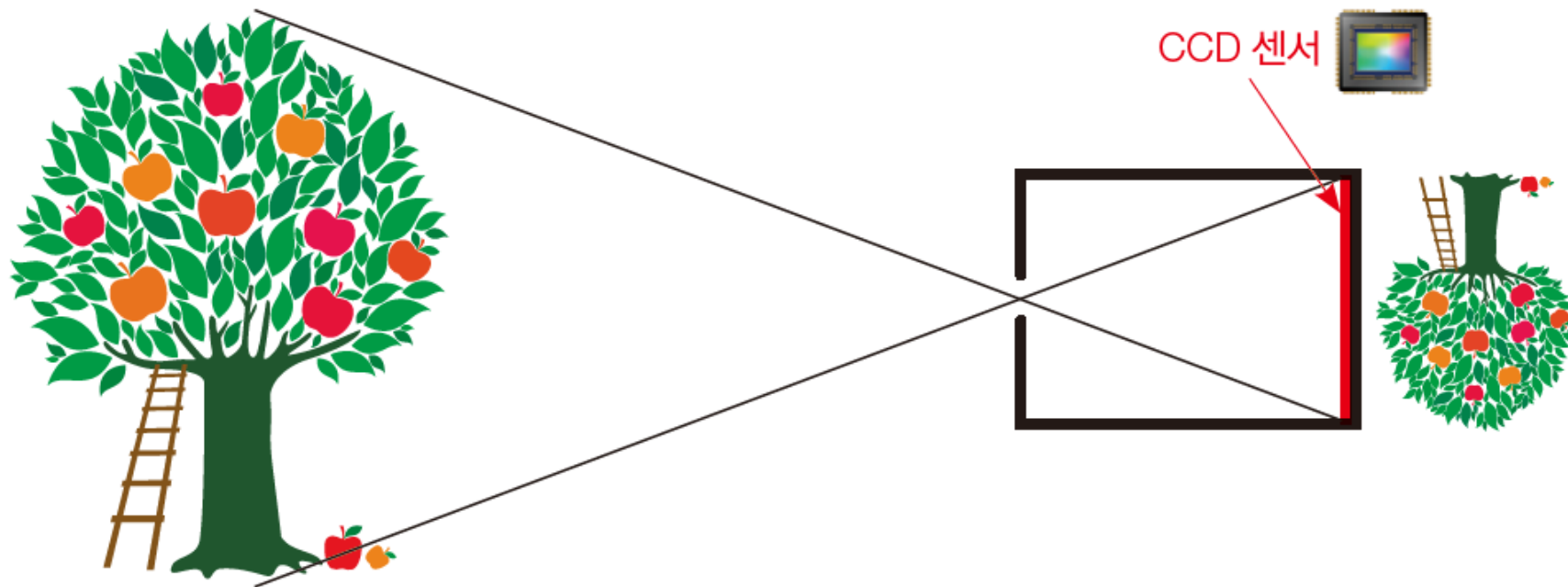
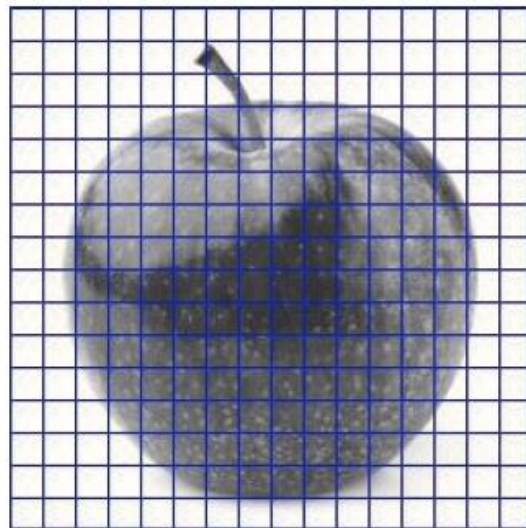


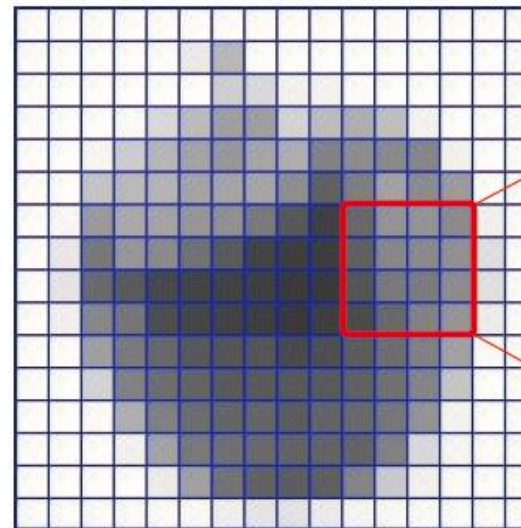
그림 3-2 핀홀 카메라 모델과 CCD 센서

## ■ 샘플링과 양자화를 통한 디지털화

- 샘플링: 연속적인 영상을  $M \times N$ 의 격자 구조로 나눠 각 위치의 밝기 정보를 추출함.
- 양자화: 샘플링된 밝기 값을  $L$ 단계로 이산화하여 숫자(픽셀 값)로 표현함.
- 이 과정을 통해 영상은 컴퓨터가 처리 가능한 디지털 영상으로 변환됨.
  - TIP: 해상도는 단위 공간당 픽셀 수를 의미.



(a) 샘플링



(b) 양자화

105	149	149	141
97	137	139	146
86	123	126	142
76	106	132	150

그림 3-3 피사체가 반사하는 빛 신호를 샘플링과 양자화를 통해 디지털 영상으로 변환

## ■ 영상 좌표계

- 좌측 상단이 원점이며  $(y, x)$  형태로 좌표를 표현함.
- 사용하는 함수(OpenCV 등)에 따라  $(x, y)$  형태로 사용되기도 함.

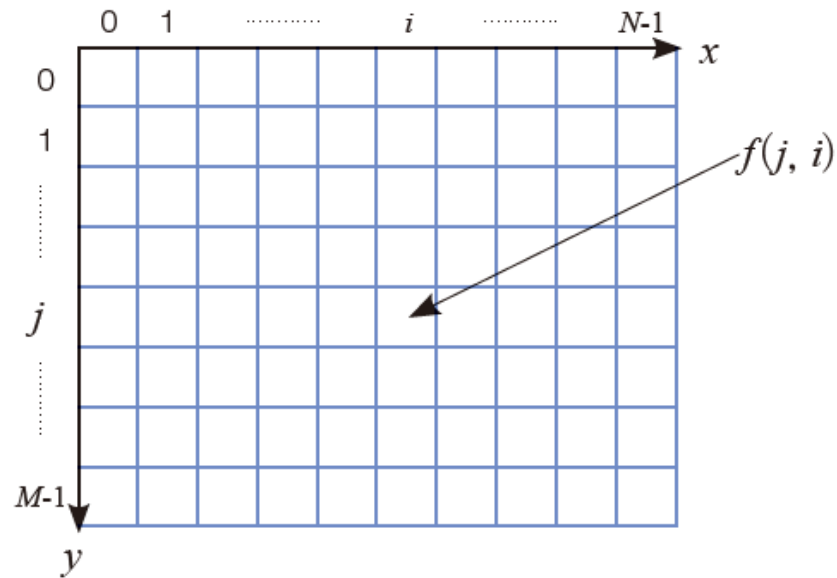


그림 3-4 디지털 영상의 좌표계

## 영상 데이터 구조와 텐서 개념

- 영상은 텐서(tensor)로 표현.

(a) 명암 영상

→ 2차원 텐서

(b) 컬러 영상(RGB)

→ 3차원 텐서

(c) 컬러 동영상

→ 시간축이 추가된 4차원 텐서

(d) 다분광/초분광/MR/CT

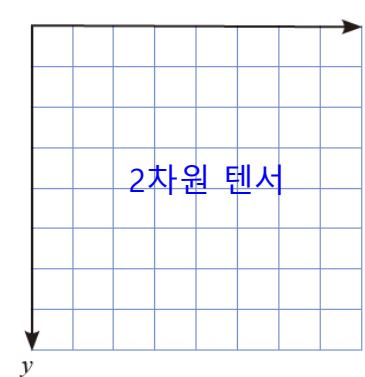
→ 다채널 스택 형태

(e) RGB-D 영상

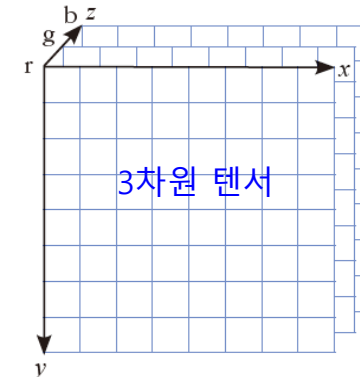
→ 깊이 정보 추가 → 조명 변화에 강함

(f) 점 구름(Point Cloud) 영상

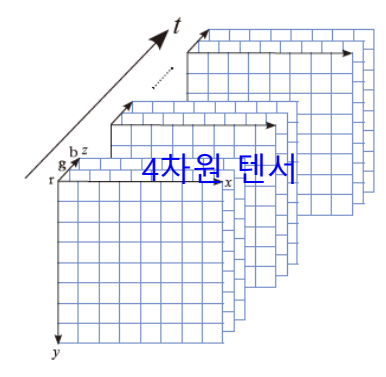
→ 라이다 등으로 획득한 3D 점 데이터



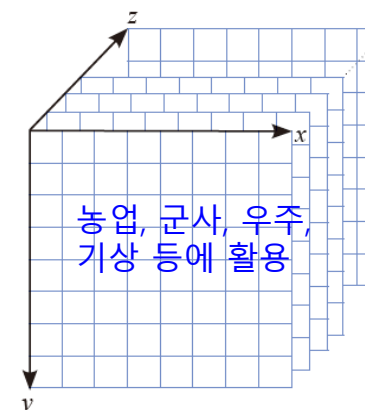
(a) 명암 영상



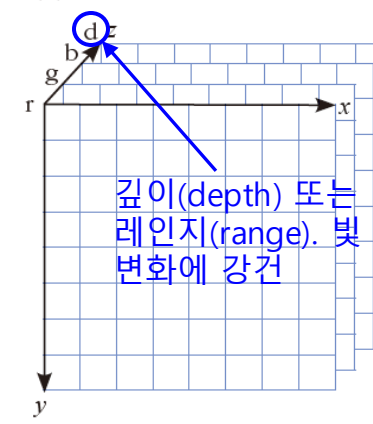
(b) 컬러 영상



(c) 컬러 동영상



(d) 다분광/초분광/MR/CT 영상



(e) RGB-D 영상

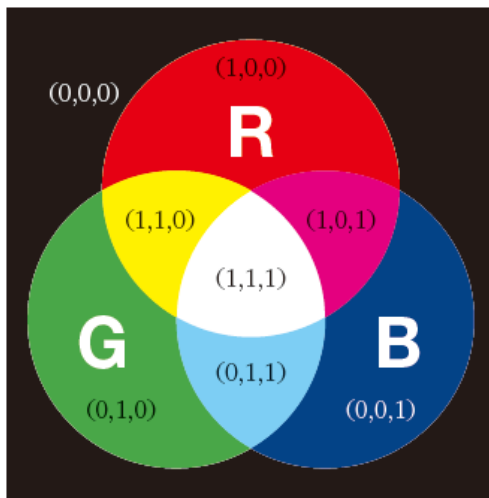


(f) 점 구름 영상

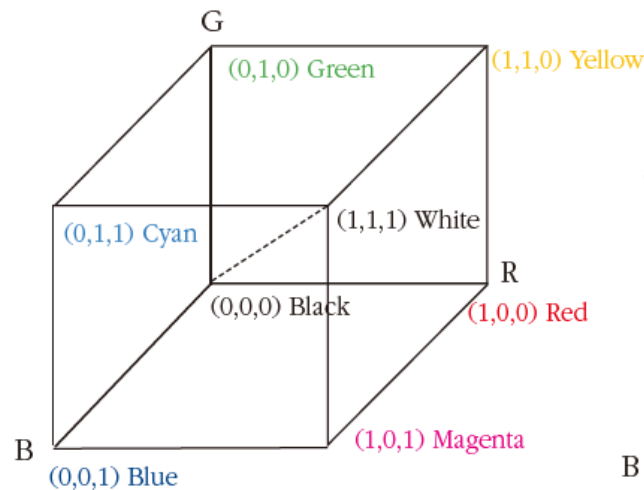
그림 3-5 다양한 형태의 디지털 영상

## ■ RGB 컬러 모델의 구조와 표현

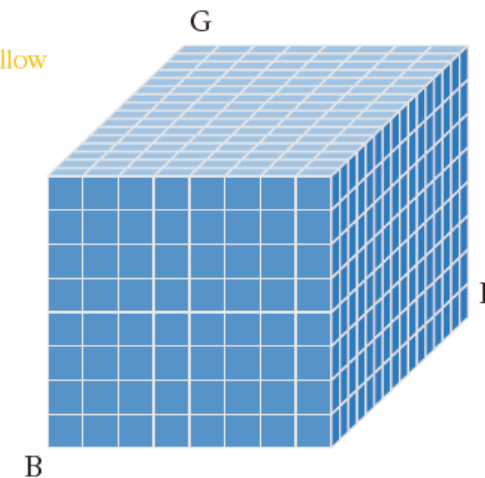
- 디지털 컬러 영상은 Red, Green, Blue 삼원색 혼합으로 색을 표현.
  - (a) RGB 삼원색 원: 색 혼합 개념 시각화
  - (b) RGB 큐브: 컬러 공간을 입체적으로 표현, 각 점이 특정 색을 의미
  - (c) 양자화된 RGB 큐브: 각 축이 정수 단위로 양자화된 컬러 데이터 구조



(a) RGB 삼원색의 혼합



(b) RGB 큐브



(c) 양자화된 RGB 큐브

그림 3-6 RGB 컬러 공간



## ■ numpy 슬라이싱을 통한 채널 분리

- numpy는 다차원 배열을 효율적으로 처리하고 연산할 수 있게 해주는 파이썬 수치 계산 라이브러리.
- 영상은 shape (높이, 너비, 채널) 구조의 3차원 배열로 표현.
- 각 채널(R/G/B)을 numpy 슬라이싱으로 분리 가능.

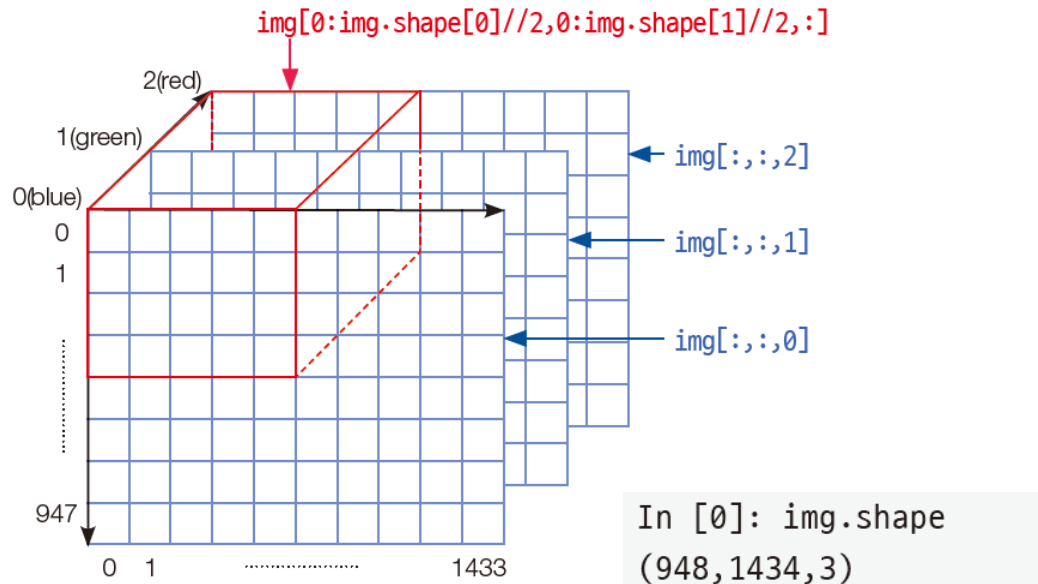
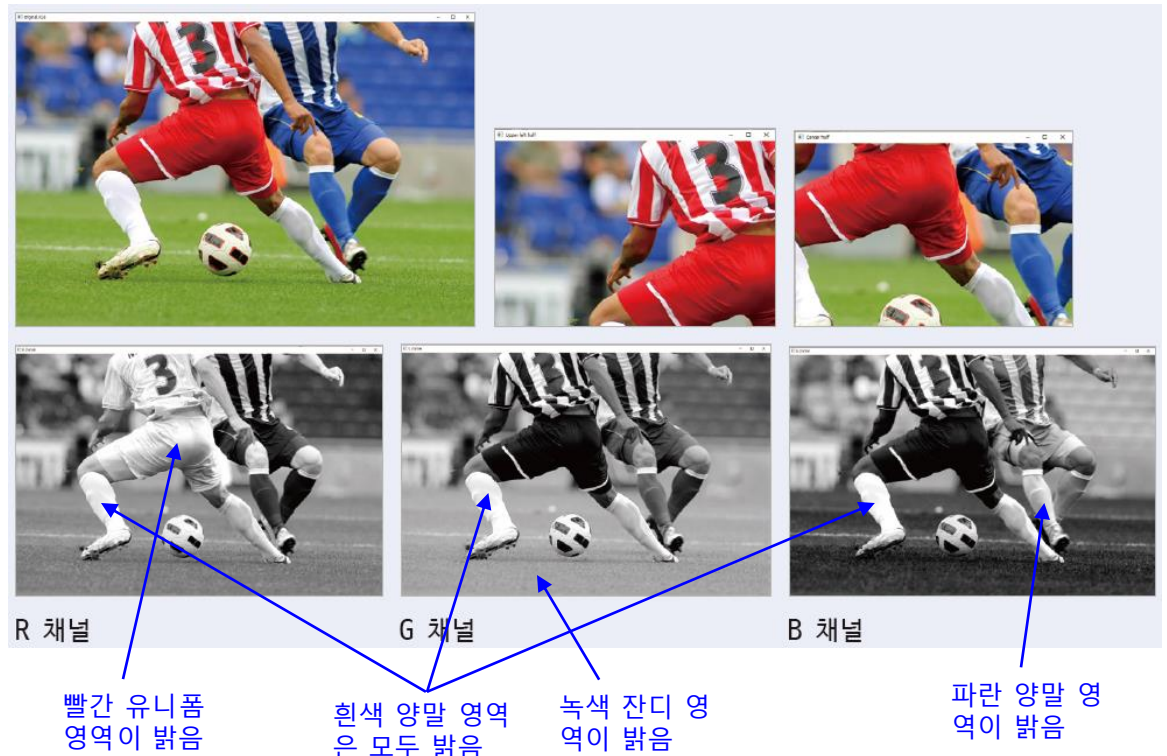


그림 3-8 numpy.ndarray의 슬라이싱을 이용한 영상 일부만 자르기([프로그램 3-1]의 10행)



## ■ 영상 처리란?

- 특정 목적을 위해 원래의 영상을 개선하거나 변형하는 작업.
  - 예: 흐린 영상에 히스토그램 평활화를 적용하여 명료도 향상.
- 화질 개선 자체가 최종 목적이 될 수도 있음.
  - 예: 차량 번호판 인식, 병변 위치 찾기 등.
- 컴퓨터 비전은 영상 처리를 전처리 단계로 활용하여 목표 성능을 향상.



(a) 안개 낀 도로 영상



(b) 히스토그램 평활화로 개선한 영상

그림 3-1 영상 처리로 화질 개선

## 영상처리의 주요 기법: 이진화

- 이진화는 임계값  $T$ 를 기준으로 픽셀 값을 0 또는 1로 변환하는 과정.
- 임계값 결정은 결과에 큰 영향을 주며, 히스토그램을 활용해 적절한 값 선택 가능.
- 엣지 및 물체 검출, 영역 분할 등에서 객체와 배경을 구분하는 용도로 사용됨.

$$b(j,i) = \begin{cases} 1, & f(j,i) \geq T \\ 0, & f(j,i) < T \end{cases}$$

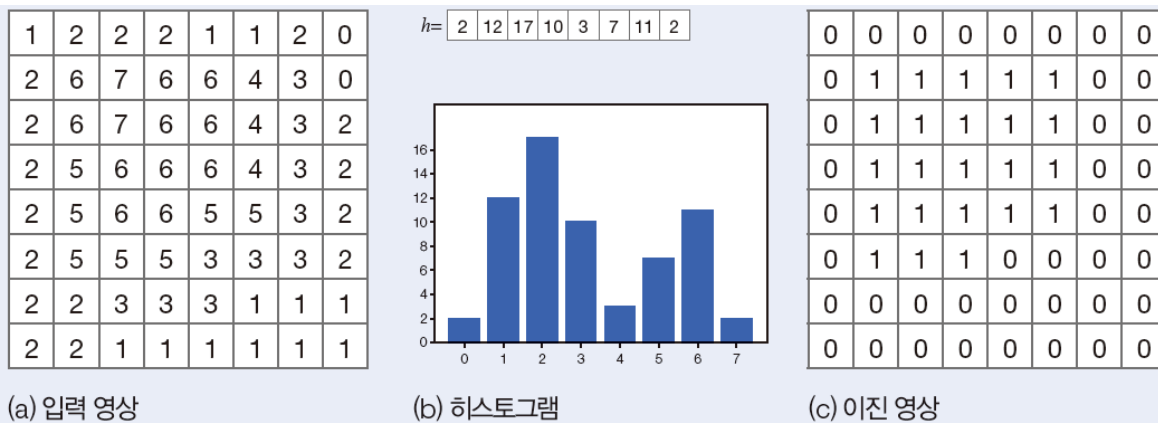


그림 3-9 히스토그램을 이용한 이진화



## 영상처리의 주요 기법: 모폴로지

- 모폴로지는 구조 요소(structuring element)를 이용하여 이진 영상의 형태를 조작하는 기법.
- 기준 픽셀 주변의 연결성을 바탕으로 픽셀 간의 인접 관계를 정의하여 영상을 처리.
  - 4-연결성: 상하좌우만 연결로 판단.
  - 8-연결성: 대각선까지 포함한 모든 방향을 연결로 판단.

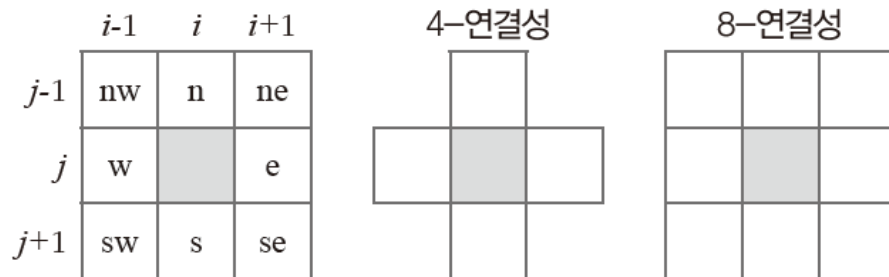


그림 3-10 화소의 연결성

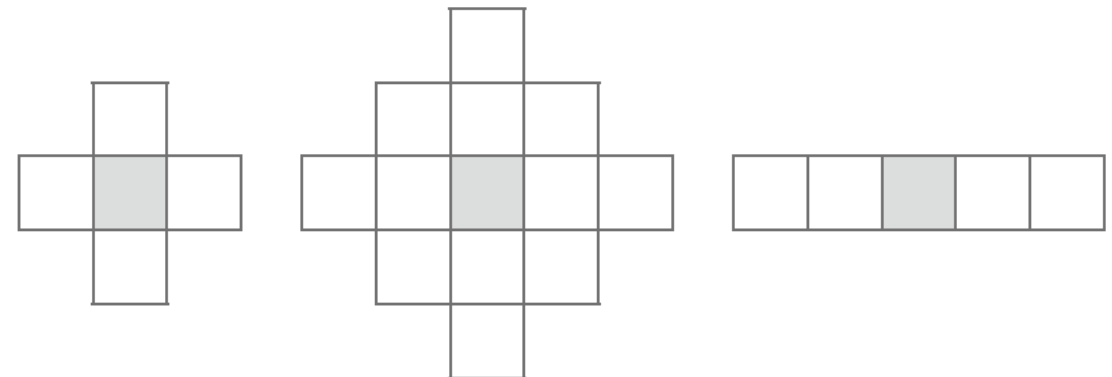
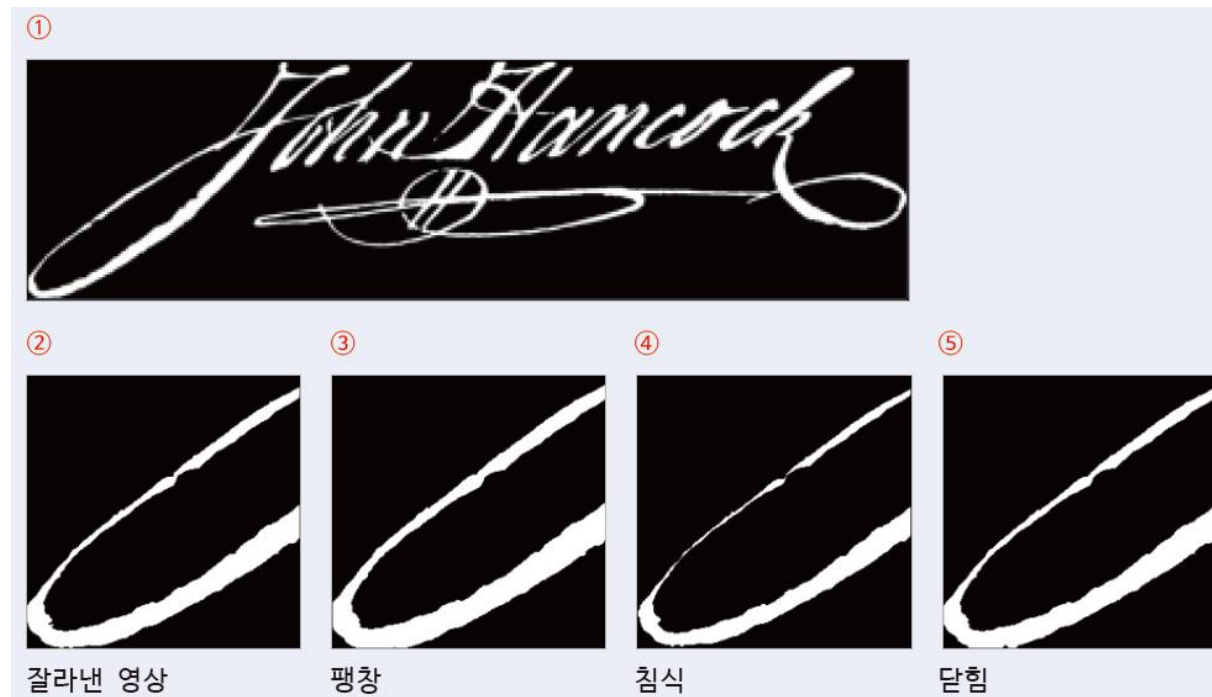


그림 3-12 모폴로지가 사용하는 구조 요소

## ■ 영상처리의 주요 기법: 모폴로지 (계속)

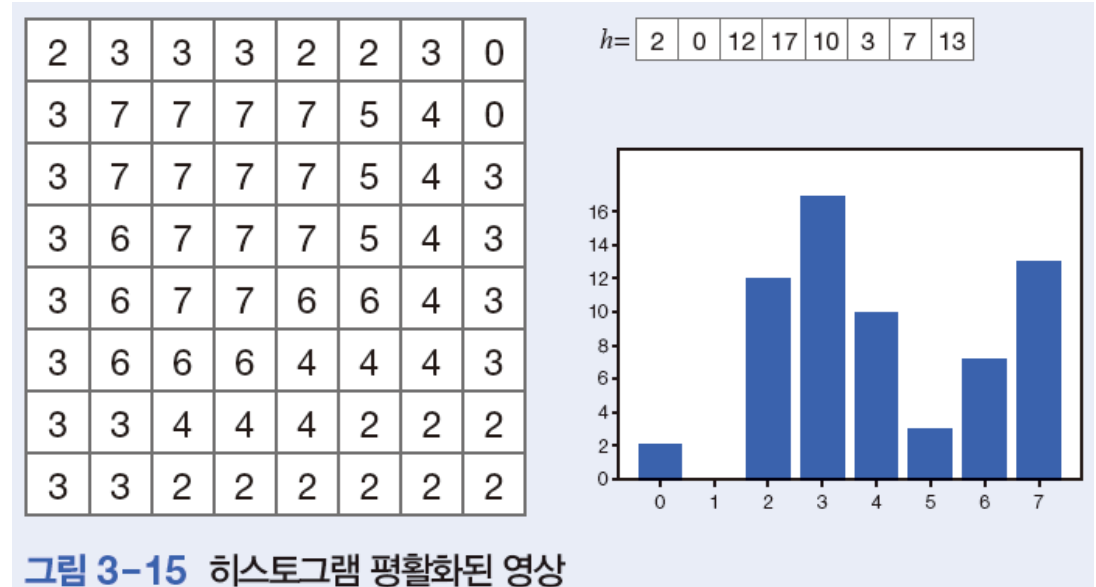
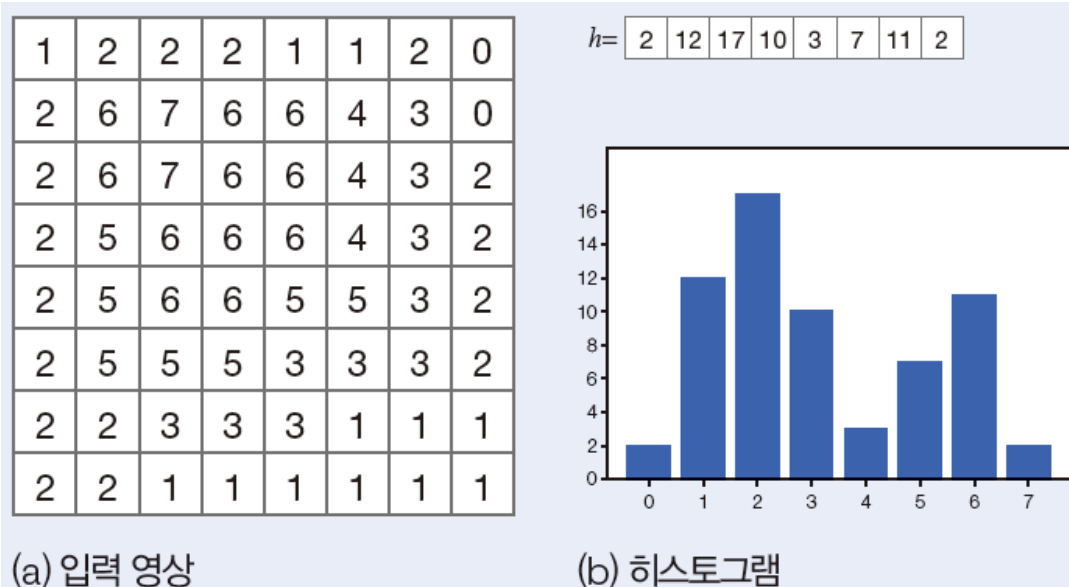
- 팽창(dilation): 작은 구멍을 메우고 단절된 영역을 연결하여 전체 영역을 확장.
- 침식(erosion): 경계에 있는 돌출부를 제거하여 영역을 축소.
- 열림(opening): 침식 후 팽창 적용. 작은 잡음 제거에 효과적.
- 닫힘(closing): 팽창 후 침식 적용. 내부의 작은 구멍 제거에 효과적.





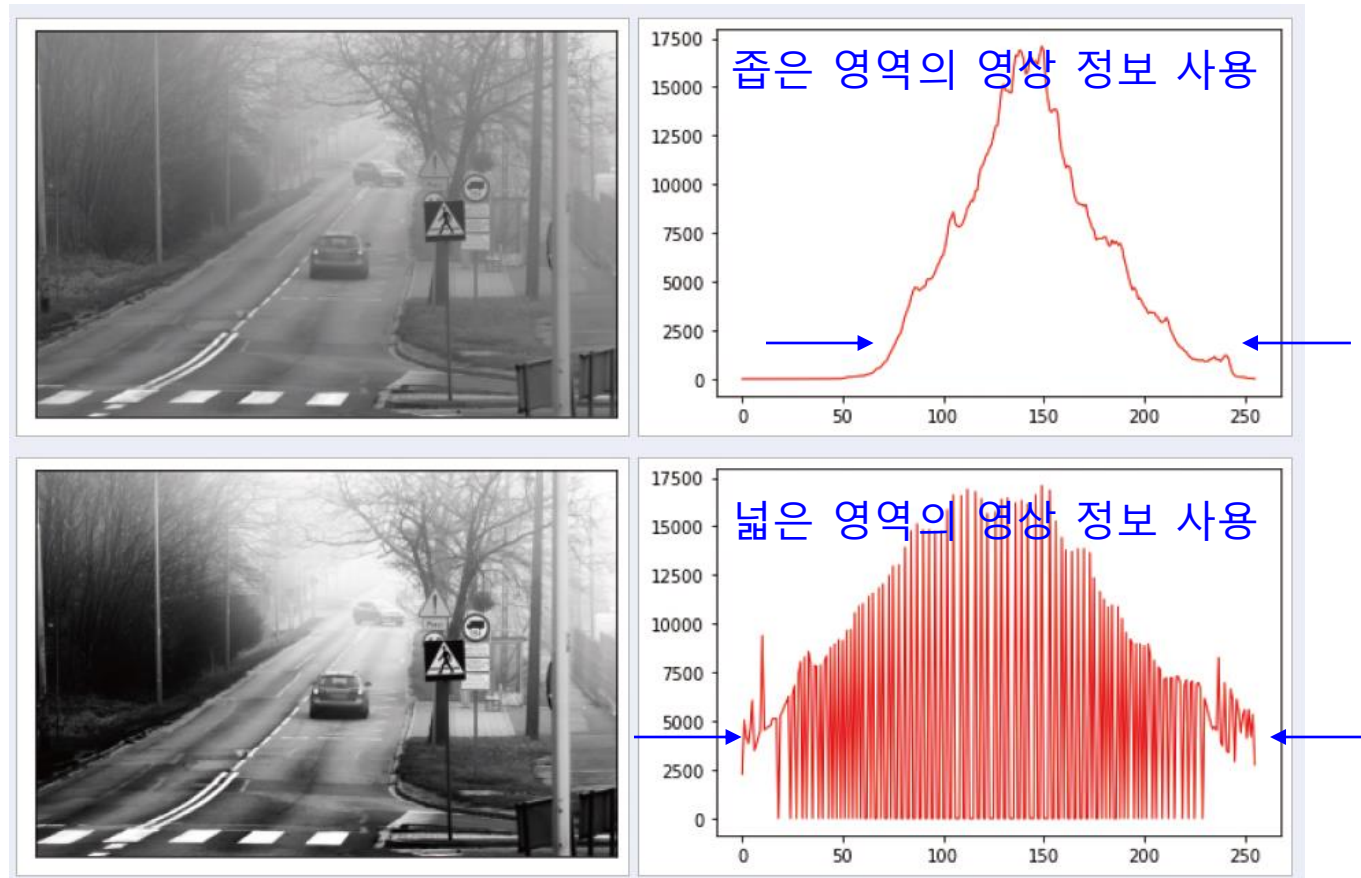
## 영상처리의 주요 기법: 히스토그램 평활화

- 히스토그램은 각 밝기 값이 영상에 몇 번 등장하는지 나타내는 분포도를 의미.
- 영상의 전체적인 밝기 구성과 대비 상태를 직관적으로 파악할 수 있음.
- 히스토그램 평활화는 밝기 분포를 넓게 퍼뜨려 영상의 명암 대비를 향상시키는 기법.



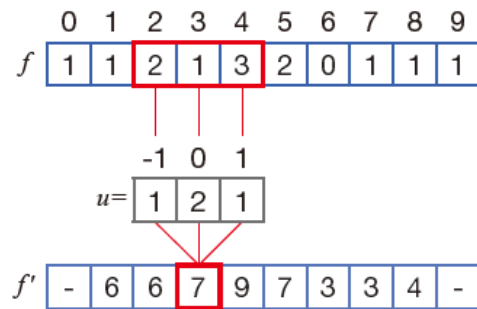
## ■ 영상처리의 주요 기법: 히스토그램 평활화 (계속)

- 히스토그램의 분포를 분석하여 평활화를 하면 영상의 시인성을 향상시킬 수 있음.
  - (예) 도로의 시인성 개선 및 객체 검출 성능 향상 가능



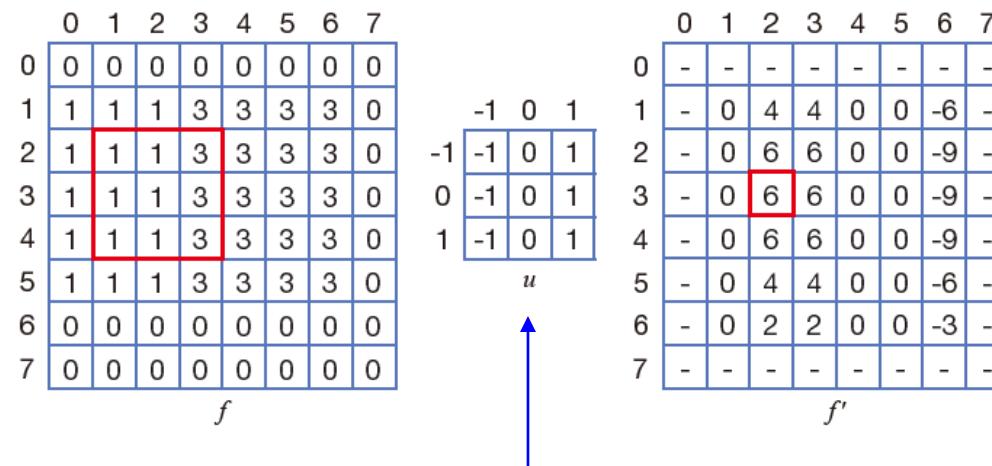
## 영상처리의 주요 기법: 필터를 적용한 영상 처리 연산

- 필터란 Edge 검출, Smoothing, Sharpening 등 다양한 영상 처리를 위해 사용됨.
- 디지털 영상에서 필터를 각 픽셀에 적용하기 위해서는 컨볼루션 연산을 통해 진행.
- 컨볼루션이란 필터를 영상의 각 픽셀에 적용해 곱의 합을 계산하는 연산을 의미.
  - 1차원 컨볼루션은 필터  $u$ 를 중심으로 입력 신호와 곱셈 후 합산.
  - 2차원 컨볼루션은 커널을 영상에 슬라이딩하면서 대응 위치의 곱을 더함.



(a) 1차원 영상에 컨볼루션 적용

그림 3-16 컨볼루션의 원리



(b) 2차원 영상에 컨볼루션 적용

수직방향 엣지를 검출하는 필터

## 영상처리의 주요 기법: 필터를 적용한 영상 처리 연산 (계속)

- 컨볼루션 필터는 목적에 따라 다양한 형태로 설계 가능.
  - Smoothing Filter : 평균 필터로 노이즈 제거 및 영상 부드럽게 처리.
  - Sharpening Filter : 경계를 강조해 세부 구조를 뚜렷하게 만듦.
  - Embossing Filter: 음영 차이를 이용해 입체감을 부여.
- ※ 필터는 커널 값의 조합을 통해 원하는 효과로 조절 가능함.

1/9	1/9	1/9	0.0030	0.0133	0.0219	0.0133	0.0030
1/9	1/9	1/9	0.0133	0.0596	0.0983	0.0596	0.0133
1/9	1/9	1/9	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

(a) 스무딩 필터

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

(b) 샤프닝 필터

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

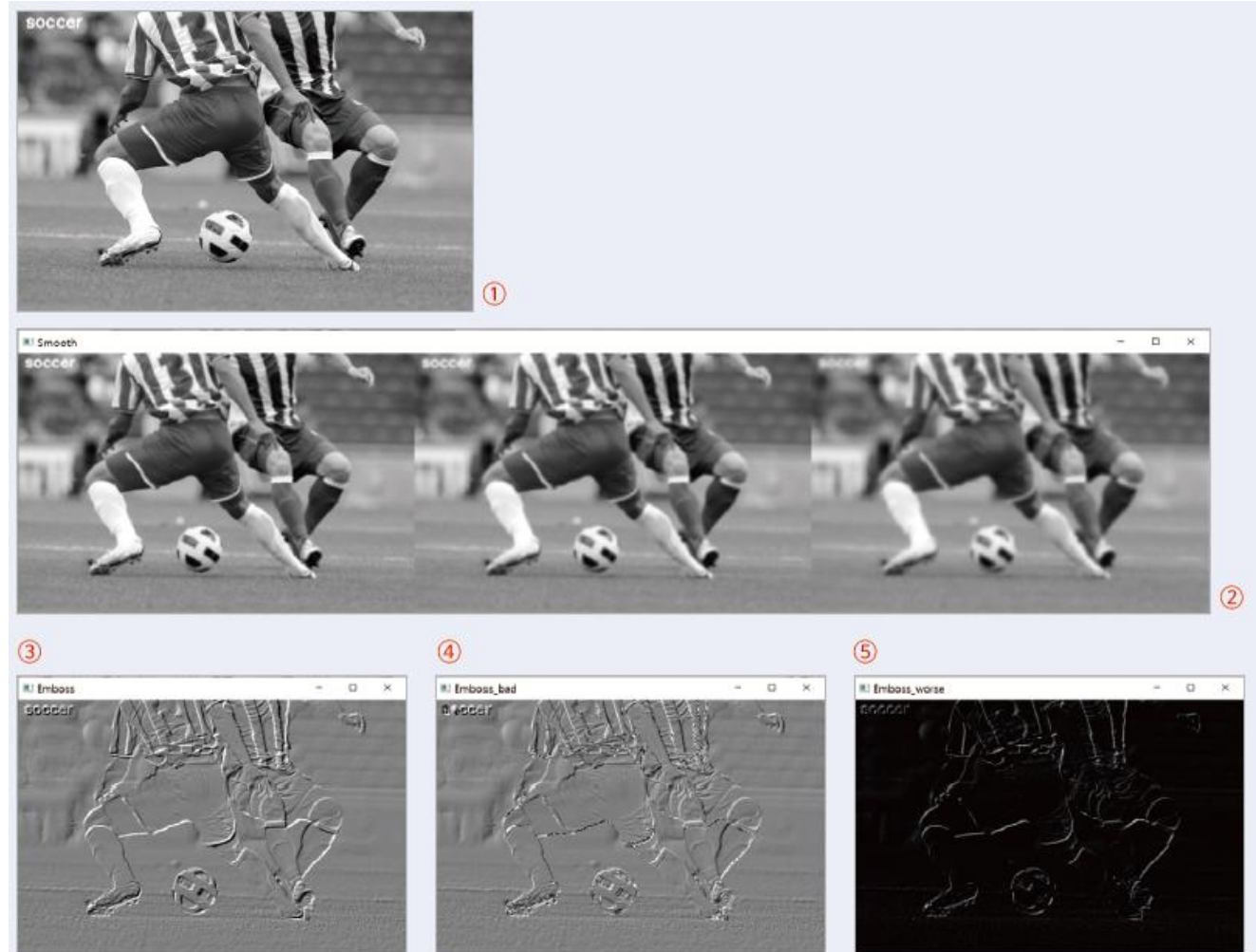
(c) 엠보싱 필터

그림 3-17 잡음 제거와 대비 향상을 위한 필터

## ■ 영상처리의 주요 기법: 필터를 적용한 영상 처리 연산 (계속)

- 필터를 적용한 영상처리 결과

- ① 원본 영상
- ② Smoothing 영상
- ③, ④ Embossing 영상
- ⑤ Edge 영상





## ■ 영상처리의 주요 기법: 기하학적 연산

- 기하학적 연산은 정의된 위치에 따라 픽셀 값을 재배치하는 처리 방식.
- 주로 이동, 회전, 크기 조절과 같은 형태 변환을 포함.
- 물체 정렬, 정규화, 정합 등에 활용.

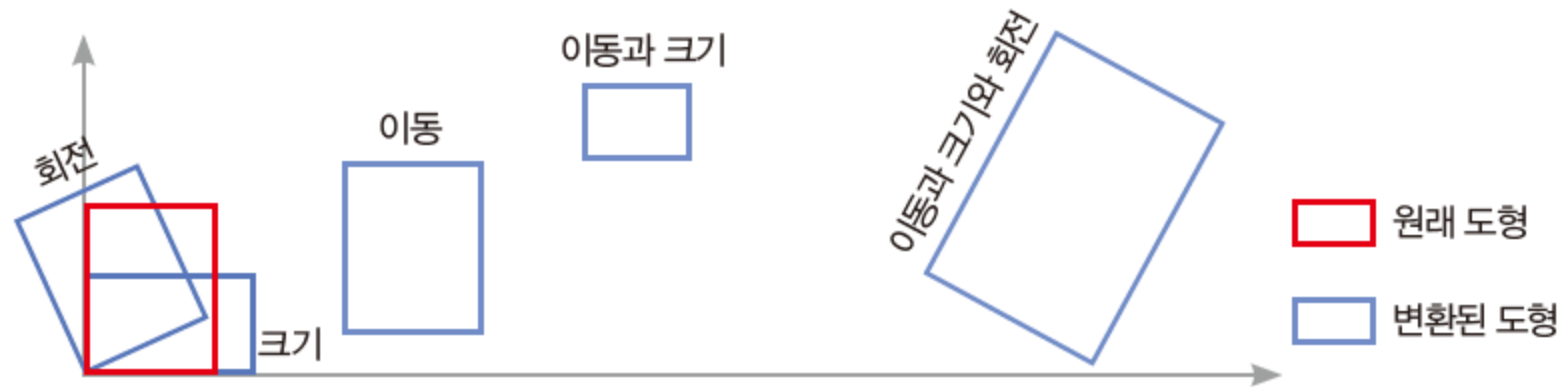


그림 3-19 여러 가지 기하 변환

## 영상처리의 주요 기법: 기하학적 연산

- 동차 행렬(homogeneous matrix)
  - 기하 연산은 동차 행렬을 사용해 일괄적으로 처리.
  - 이동 행렬: x, y 방향으로 지정된 거리만큼 이동.
  - 회전 행렬: 원점을 중심으로  $\theta$ 만큼 회전.
  - 크기 조절 행렬: x, y 방향으로 각각  $s_x, s_y$ 배 크기 변환.
  - 이들 행렬은 모두 어파인 변환(affine transform)에 속하며, 평행 관계를 유지.

표 3-1 3가지 기하 변환

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

Thank you!