

컴퓨터 비전

동양미래대학교
인공지능소프트웨어학과
백찬은 교수

OpenCV 실습

01

영상

02

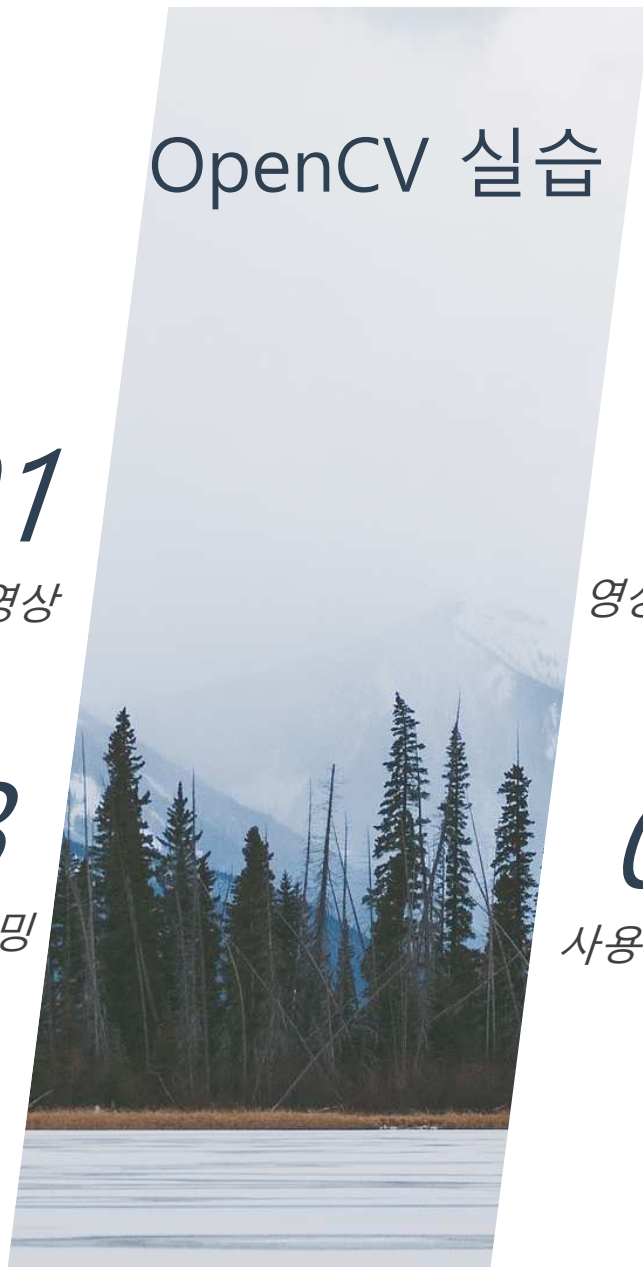
영상입출력, 영상변환

03

웹캠 영상 활용 프로그래밍

04

사용자 인터페이스 구현하기

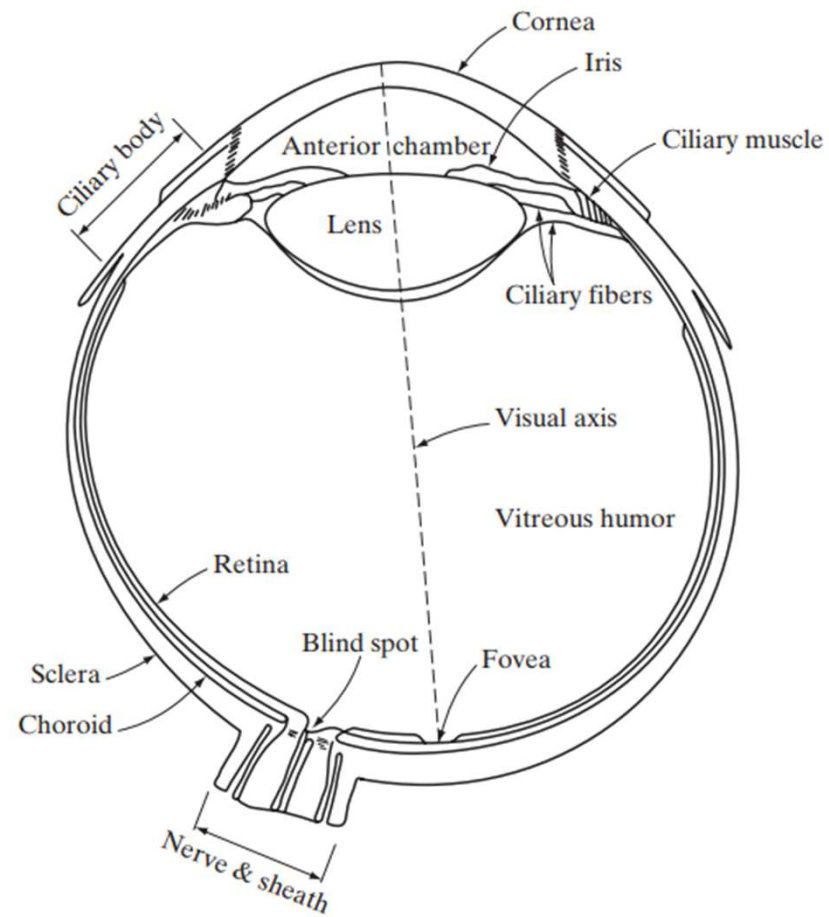




영상

인간의 눈과 카메라

FIGURE 2.1
Simplified
diagram of a cross
section of the
human eye.



인간의 눈과 카메라

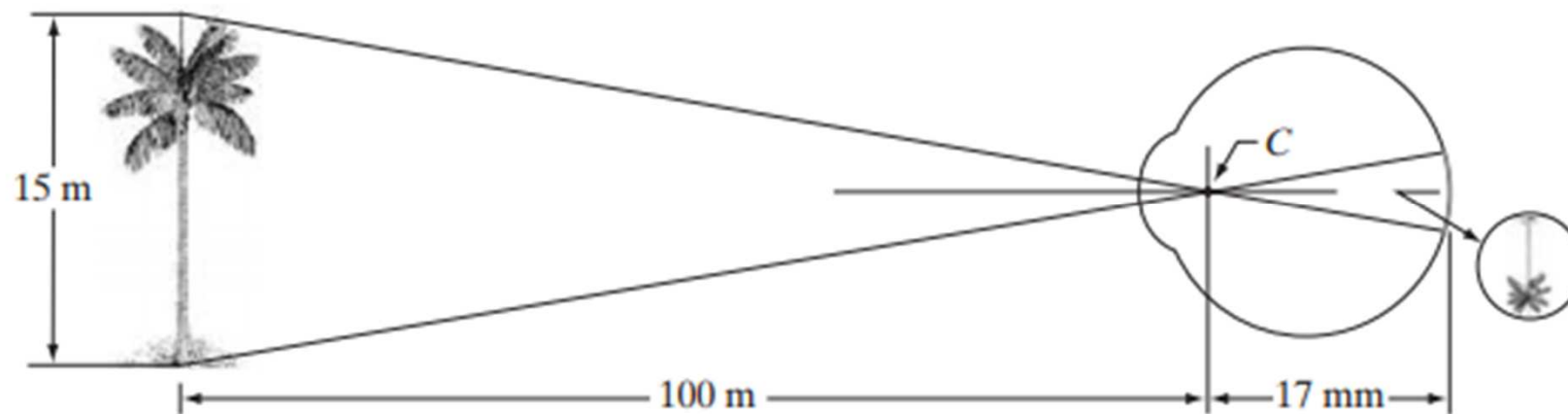
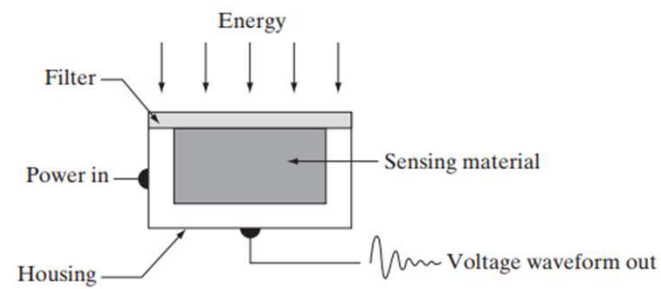


FIGURE 2.3

Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.

인간의 눈과 카메라



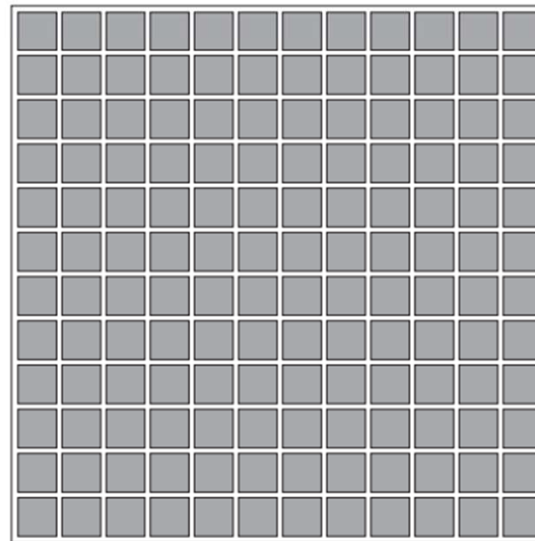
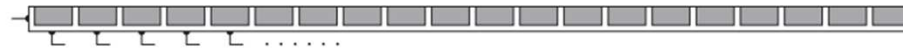
a
b
c

FIGURE 2.12

(a) Single imaging sensor.

(b) Line sensor.

(c) Array sensor.



디지털 영상의 획득

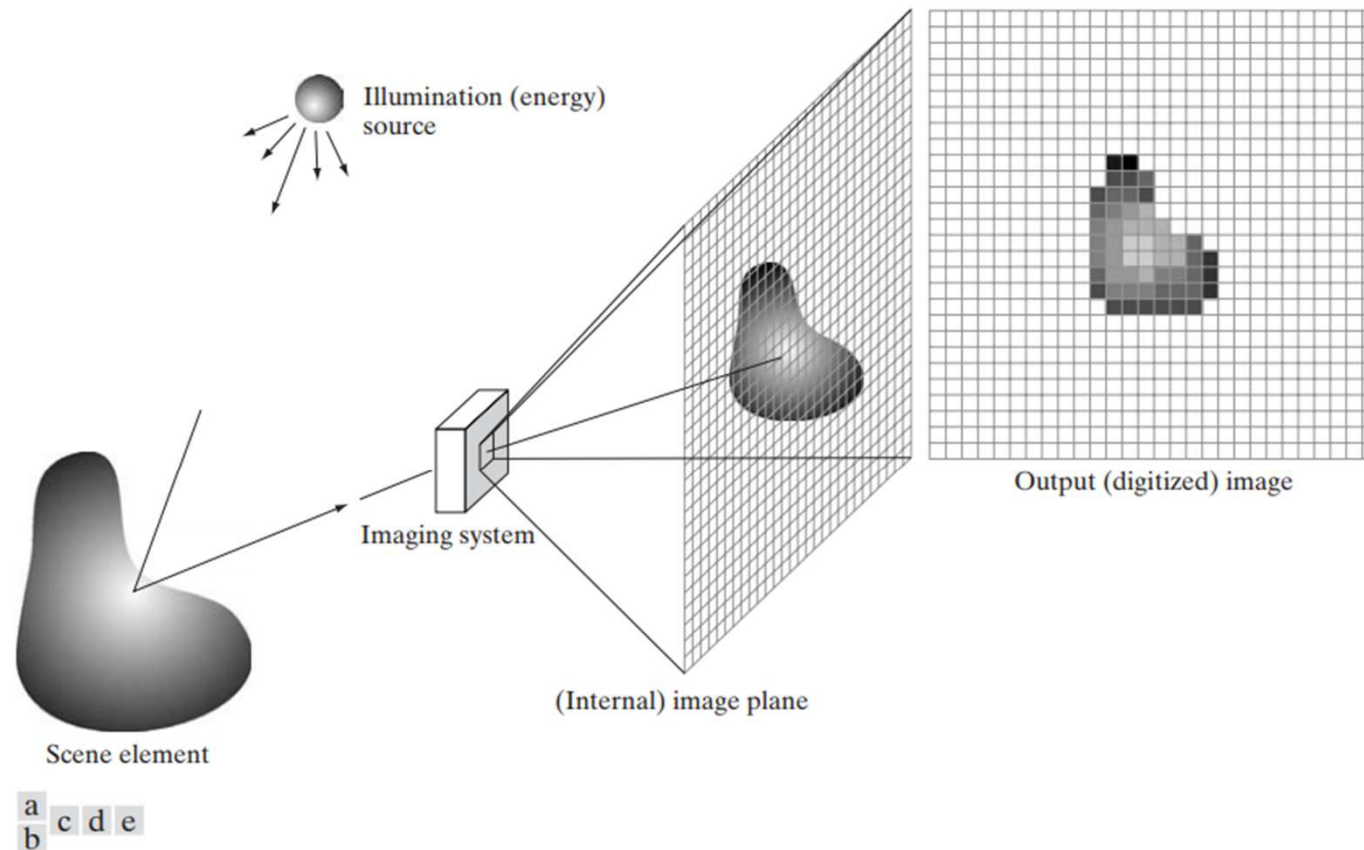
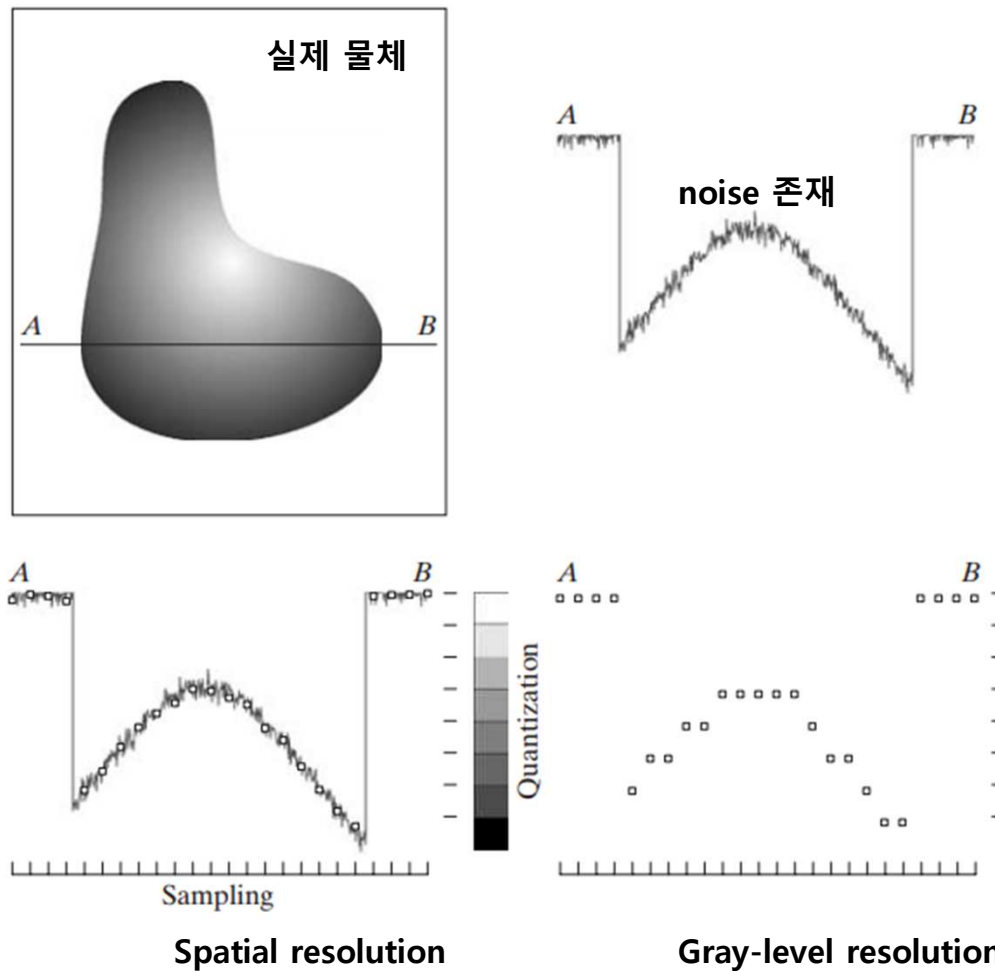


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

디지털 영상의 획득



a b
c d

FIGURE 2.16
Generating a digital image.
(a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

continuous → digitized

Sampling과 Quantization

- 이미지를 몇 픽셀로 나타낼 것인가? = Sampling(샘플링)
- 이미지의 밝기를 몇 단계로 표현할 것인가? = Quantization(양자화)
- **Sampling**
 - 카메라 센서는 이미지를 격자 모양으로 잘라 저장함.
 - 얼마나 촘촘하게 자를 것인가를 결정하는 것이 샘플링.
 - 샘플링 간격이 좁을수록 고해상도 사진이 됨.
- **Quantization**
 - 각 픽셀의 색이나 밝기를 얼마나 다양하게 나타낼 것인가를 결정하는 것.
 - 예: 밝기 단위가 2단계 - 흑백만 있는 이진 이미지.
 - 밝기의 단계가 많을수록 자연스러운 이미지

Sampling과 Quantization

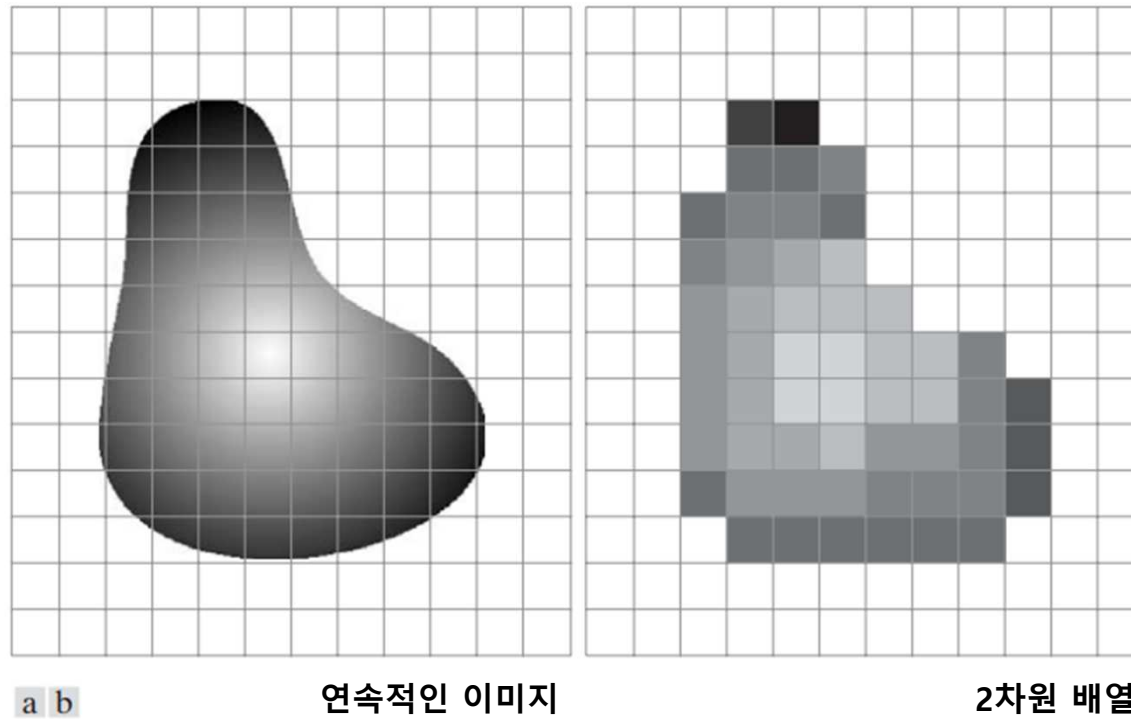


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Sampling 간격이 좁으면 화질이 높아지고,

Quantization level이 높아지면 여러가지 밝기를 표현할 수 있음

Sampling과 Quantization

- **Quantization**
- 밝기 레벨의 수: $L = 2^k$
- 즉, 2, 4, 8, 16, ...

$k = 1$: binary image, Dynamic range: [0,1]

$k = 8$: grayscale image, Dynamic range: [0,255]

$k \geq 10$: medical image

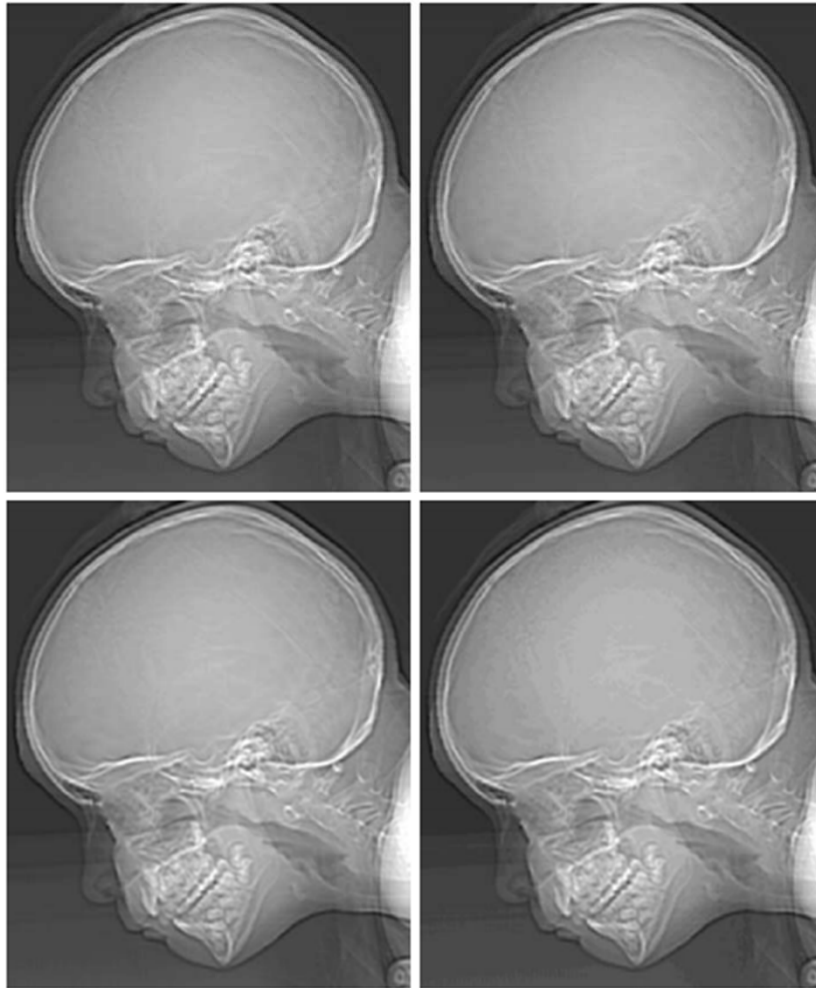


Sampling과 Quantization

- 이미지에 사용되는 밝기의 단계에 따라

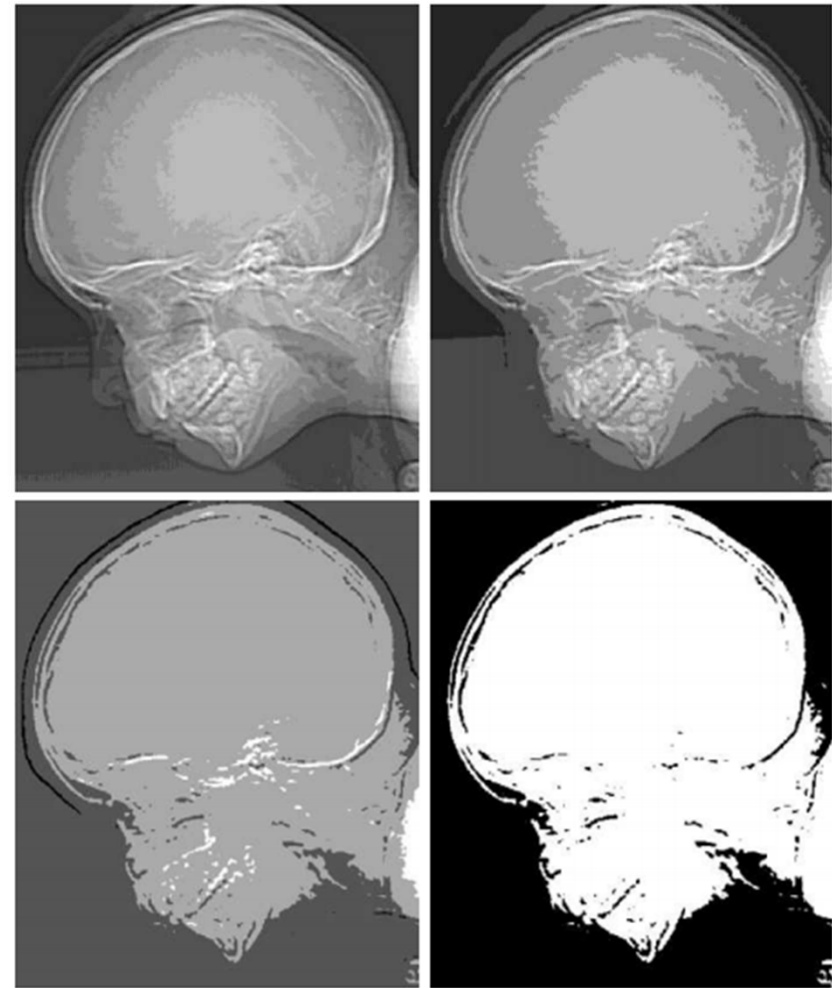
a b
c d

FIGURE 2.21
(a) 452×374 ,
256-level image.
(b)–(d) Image
displayed in 128,
64, and 32
intensity levels,
while keeping the
image size
constant.



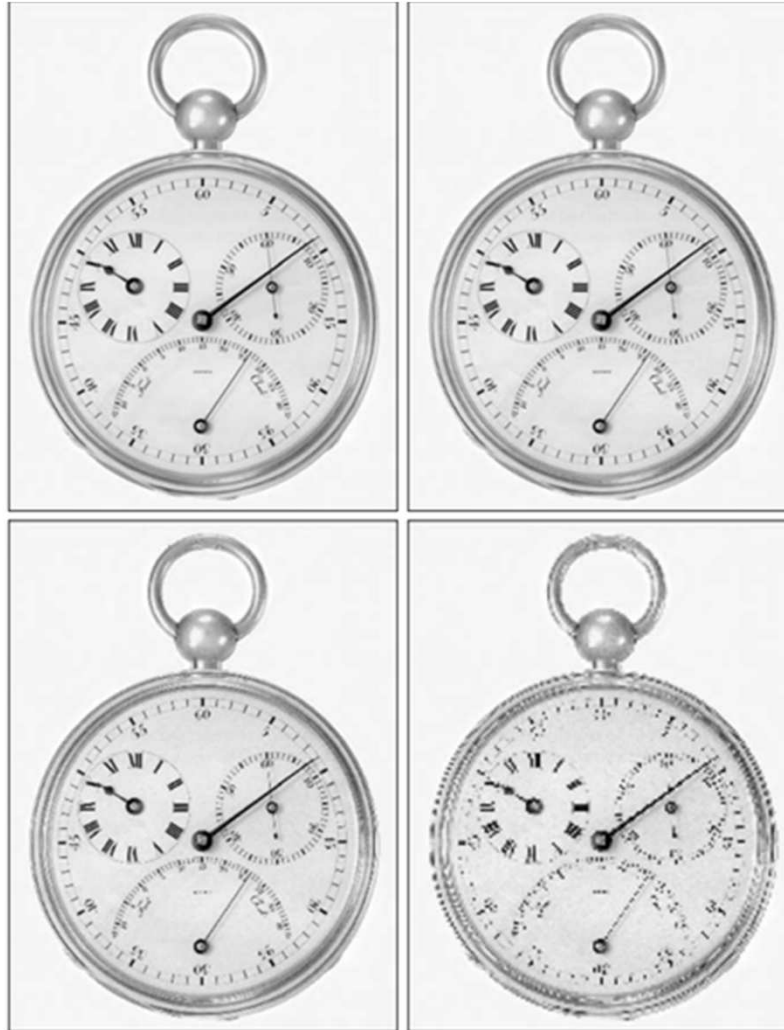
e f
g h

FIGURE 2.21
(Continued)
(e)–(h) Image
displayed in 16, 8,
4, and 2 intensity
levels. (Original
courtesy of
Dr. David R.
Pickens,
Department of
Radiology &
Radiological
Sciences,
Vanderbilt
University
Medical Center.)

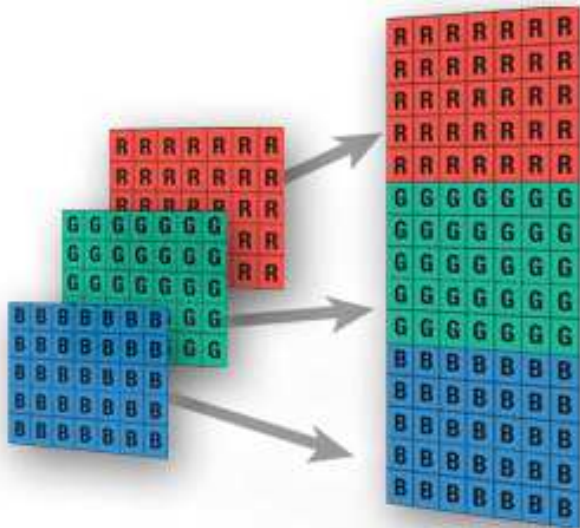


Sampling과 Quantization

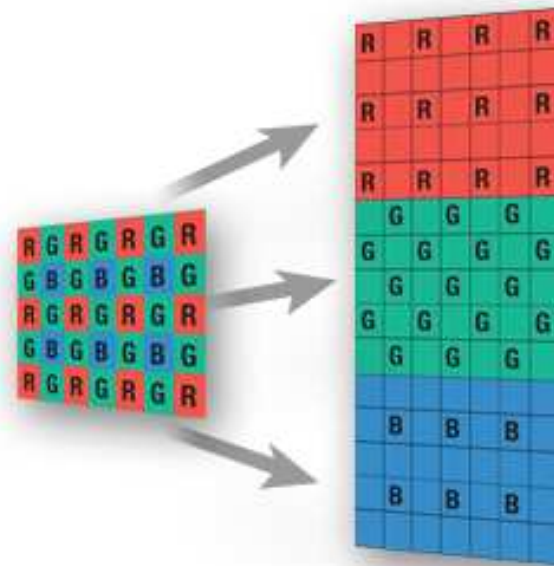
- 이미지의 가로, 세로 해상도에 따라



컬러 이미지의 취득



Three Shot Color Filter Camera



Color Mosaic Camera

컬러 이미지의 취득

● 256개의 밝기값을 나타내는 카메라에서 512*512 픽셀의 영상을 취득하면 용량은?

(Three Shot Color Filter Camera인 경우만 생각한다.)

● 256개의 밝기값을 나타내는 카메라에서
512*512 픽셀의 30분 흑백 동영상을 취득하면 용량은?
단, 동영상은 60fps이다.



영상입출력, 영상변환

객체지향 특성과 강점

- 객체는 능동적
 - '.'을 찍어 자신이 가진 함수를 능동적으로 호출할 수 있음.
 - 객체 내부에 있는 함수를 멤버 함수(member function), 또는 메서드(method)라 함.
 - 객체 내부에는 데이터를 저장할 멤버 변수(member variable)도 있음.
- 필요한 만큼 얼마든지 찍어낼 수 있음



(a) a 객체



(b) b 객체



(c) c 객체

그림 2-8 np.ndarray 클래스로 찍어낸 객체들

(실습)객체지향 특성

프로그램 2-1

numpy.ndarray 클래스 형의 객체를 만들고 멤버 함수 적용하기

```
01 import numpy as np
02
03 a=np.array([4,5,0,1,2,3,6,7,8,9,10,11])
04 print(a) ①
05 print(type(a)) ②
06 print(a.shape) ③
07 a.sort()
08 print(a) ④
09
10 b=np.array([-4.3,-2.3,12.9,8.99,10.1,-1.2])
11 b.sort()
12 print(b) ⑤
13
14 c=np.array(['one','two','three','four','five','six','seven'])
15 c.sort()
16 print(c) ⑥
```

```
[ 4  5  0  1  2  3  6  7  8  9 10 11] ①
<class 'numpy.ndarray'> ②
(12,) ③
[ 0  1  2  3  4  5  6  7  8  9 10 11] ④
[-4.3 -2.3 -1.2  8.99 10.1 12.9 ] ⑤
['five' 'four' 'one' 'seven' 'six' 'three' 'two'] ⑥
```

(실습)객체지향 특성

- type: 객체의 클래스를 알려주는 함수
- dir: 사용 가능한 멤버 함수를 알려주는 함수
- help: 함수가 하는 일에 대한 설명을 해주는 함수

****개발하다 기억이 분명히 떠오르지 않을 때엔 type, dir, help 명령어로 확인할 수 있어야 함****

```
In [1]: type(a)
numpy.ndarray
In [2]: dir(a)
['T', '__abs__', '__add__', '__and__', ..., 'all', 'any', ..., 'max', ..., 'shape', ..., 'sort', ...]
```

```
In [3]: help(a.sort)
Help on built-in function sort:

sort(...) method of numpy.ndarray instance
    a.sort(axis=-1, kind=None, order=None)

    Sort an array in-place. Refer to `numpy.sort` for full documentation.

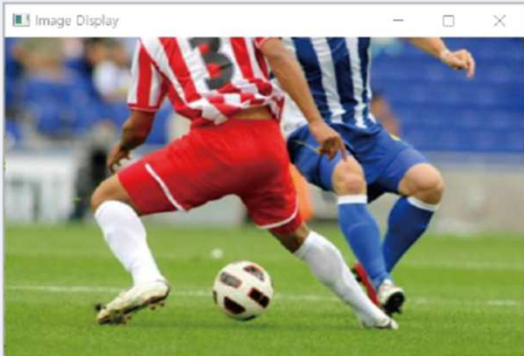
    Parameters
    -----
    axis : int, optional
```

(실습-복습) 영상 표시하기

프로그램 2-2

영상 파일을 읽고 윈도우에 디스플레이하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')    # 영상 읽기
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 cv.imshow('Image Display',img) # 윈도우에 영상 표시
10
11 cv.waitKey()
12 cv.destroyAllWindows()
```



OpenCV에서는 영상을 numpy.ndarray 클래스로 표현함.

그래서 OpenCV가 다루는 영상에는 numpy가 제공하는 다양한 기능(함수)을 사용할 수 있음.

값을 출력해볼 수도 있음

"""5번 줄에 추가"""

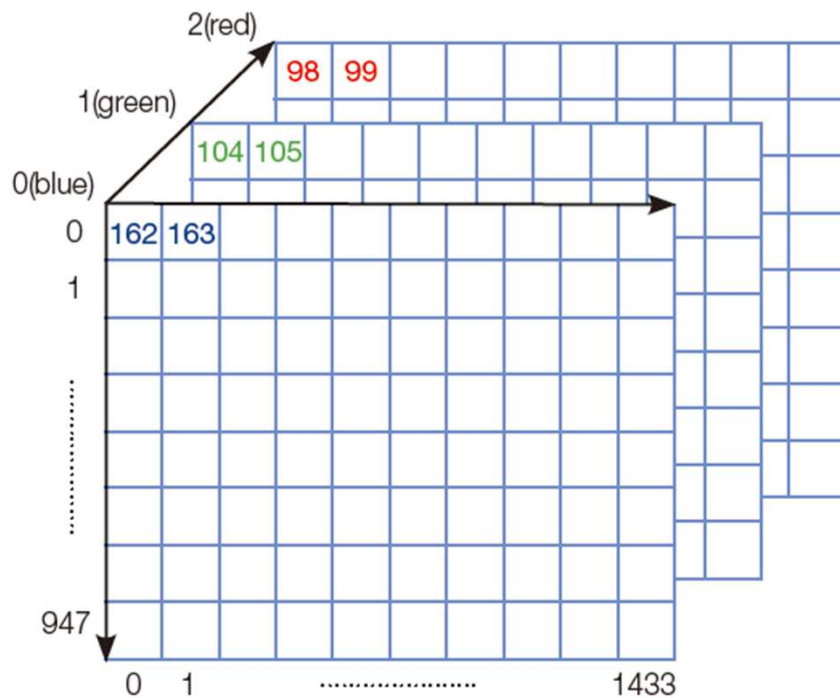
print(img[0,0])

➔ B, G, R 세 가지 값이 나타남

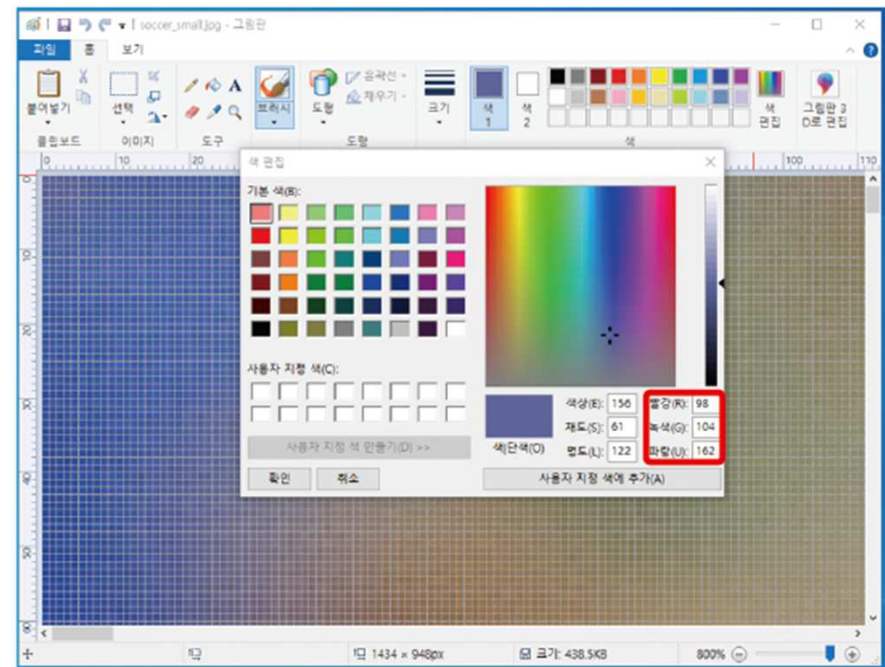
print(img[0,0,0], img[0,0,1], img[0,0,2]) 도 추가해보기

➔ 각 B, G, R 값들을 하나씩 확인할 수 있음

(실습-복습)영상 표시하기



(a) 프로그램으로 조사

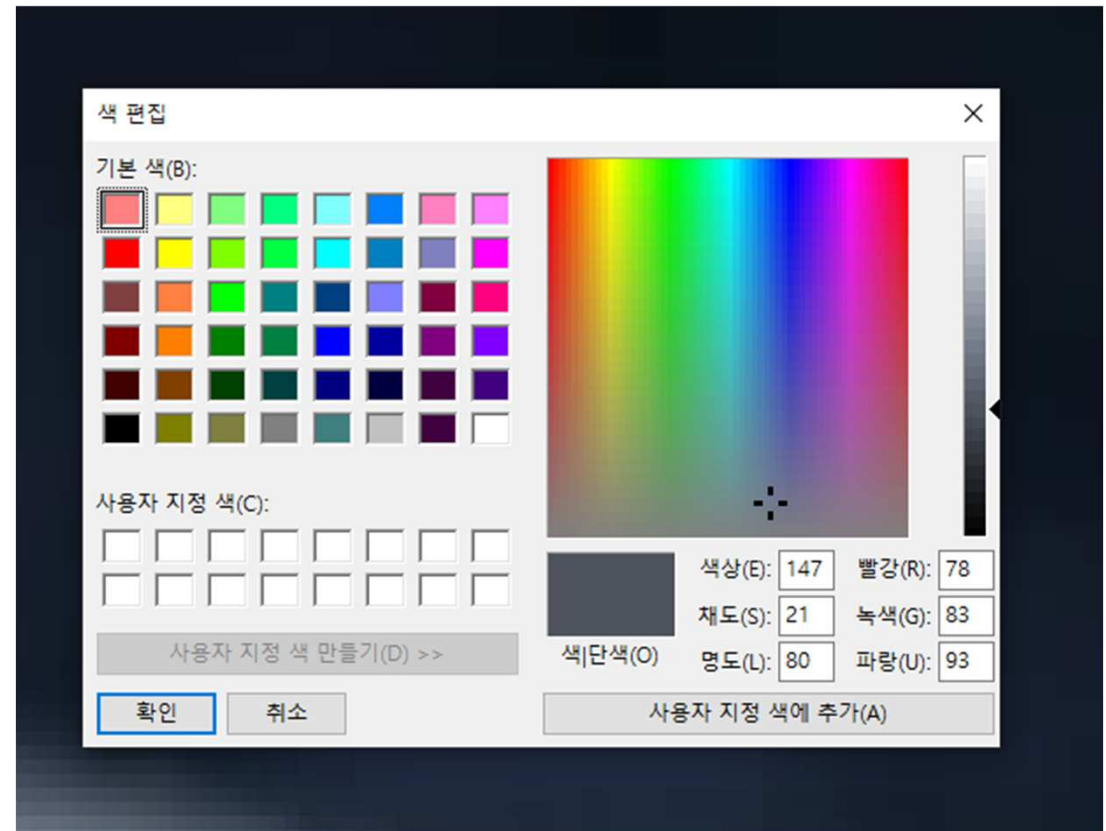


(b) 그림판으로 조사

그림 2-9 img 객체가 표현하는 영상의 구조와 내용

(실습-복습) 영상 표시하기

```
[93 83 78]  
In [5]: %runfile 'F:/class/_컴퓨터비전/code/p2_2.py' --wdir  
[93 83 78]  
In [6]:
```

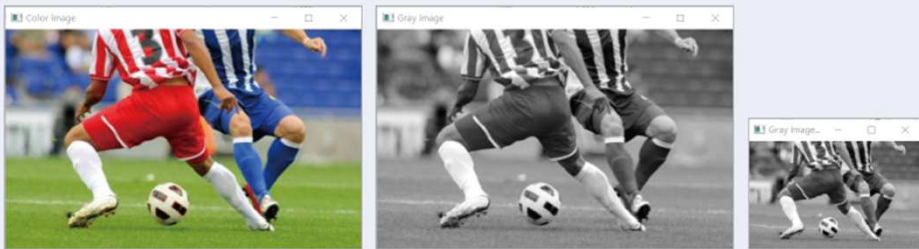


(실습) 영상 변환하기

프로그램 2-3

영상을 명암 영상으로 변환하고 반으로 축소하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY) # BGR 컬러 영상을 명암 영상으로 변환
10 gray_small=cv.resize(gray,dsize=(0,0),fx=0.5,fy=0.5) # 반으로 축소
11
12 cv.imwrite('soccer_gray.jpg',gray) # 영상을 파일에 저장
13 cv.imwrite('soccer_gray_small.jpg',gray_small)
14
15 cv.imshow('Color image',img)
16 cv.imshow('Gray image',gray)
17 cv.imshow('Gray image small',gray_small)
18
19 cv.waitKey()
20 cv.destroyAllWindows()
```



(실습)영상 변환하기

cvtColor 함수가 컬러 영상을 명암 영상으로 바꾸는 방법

$$I = \text{round}(0.299 \times R + 0.587 \times G + 0.114 \times B)$$

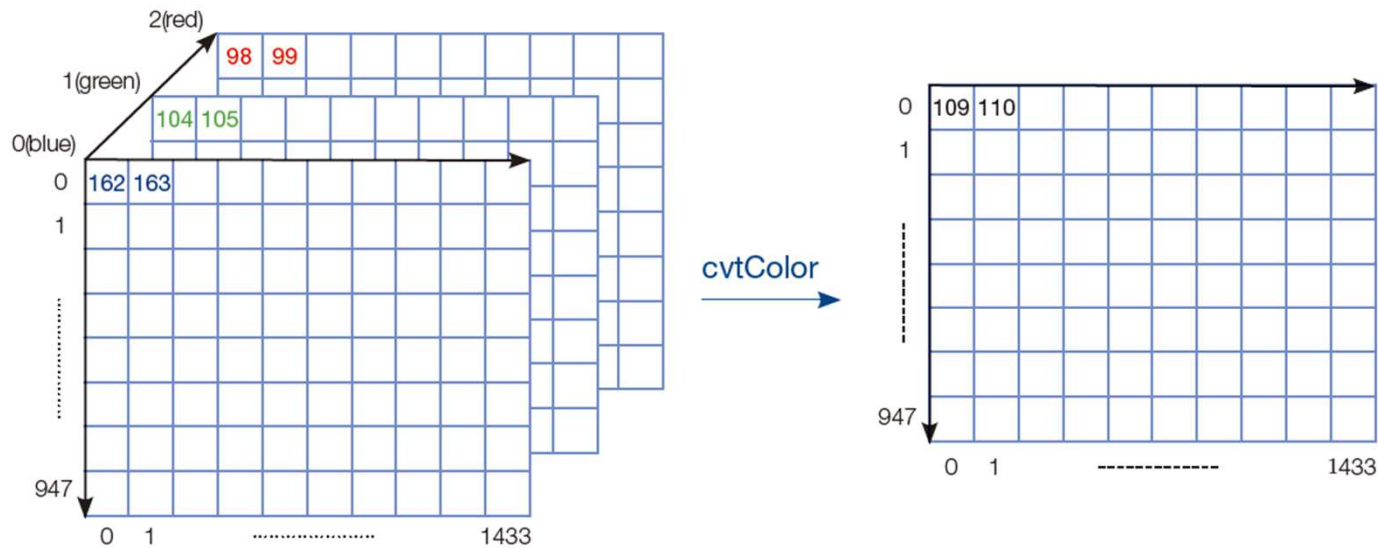


그림 2-10 BGR 컬러 영상을 명암 영상으로 변환



웹캠 영상 활용 프로그래밍

(실습)웹캠에서 비디오 읽기

프로그램 2-4

웹 캠으로 비디오 획득하기

```
01 import cv2 as cv
02 import sys
03
04 cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도
05
06 if not cap.isOpened():
07     sys.exit('카메라 연결 실패')
08
09 while True:
10     ret,frame=cap.read()          # 비디오를 구성하는 프레임 획득
11
12     if not ret:
13         print('프레임 획득에 실패하여 루프를 나갑니다.')
14         break
15
16     cv.imshow('Video display',frame)
17
18     key=cv.waitKey(1)             # 1밀리초 동안 키보드 입력 기다림
19     if key==ord('q'):             # 'q' 키가 들어오면 루프를 빠져나감
20         break
21
22 cap.release()                   # 카메라와 연결을 끊음
23 cv.destroyAllWindows()
```



(실습)비디오에서 영상(이미지) 수집하기

프로그램 2-5

비디오에서 수집한 영상을 이어 붙이기

```
01 import cv2 as cv
02 import numpy as np
03 import sys
04
05 cap=cv.VideoCapture(0,cv.CAP_DSHOW)      # 카메라와 연결 시도
06
07 if not cap.isOpened():
08     sys.exit('카메라 연결 실패')
09
10 frames=[]
11 while True:
12     ret,frame=cap.read()                 # 비디오를 구성하는 프레임 획득
13
14     if not ret:
15         print('프레임 획득에 실패하여 루프를 나갑니다.')
16         break
17
```

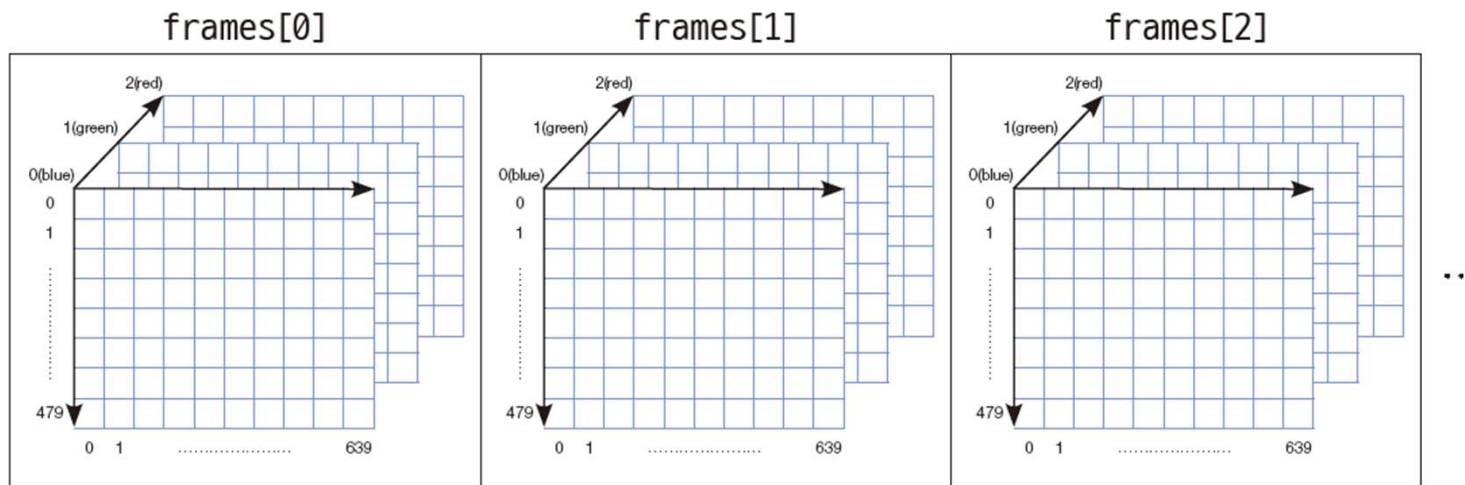
(실습)비디오에서 영상(이미지) 수집하기

```
18 cv.imshow('Video display',frame)
19
20 key=cv.waitKey(1)           # 1밀리초 동안 키보드 입력 기다림
21 if key==ord('c'):          # 'c' 키가 들어오면 프레임을 리스트에 추가
22     frames.append(frame)
23 elif key==ord('q'):        # 'q' 키가 들어오면 루프를 빠져나감
24     break
25
26 cap.release()              # 카메라와 연결을 끊음
27 cv.destroyAllWindows()
28
29 if len(frames)>0:           # 수집된 영상이 있으면
30     imgs=frames[0]
31     for i in range(1,min(3,len(frames))): # 최대 3개까지 이어 붙임
32         imgs=np.hstack((imgs,frames[i]))
33
34     cv.imshow('collected images',imgs)
35
36     cv.waitKey()
37     cv.destroyAllWindows()
```

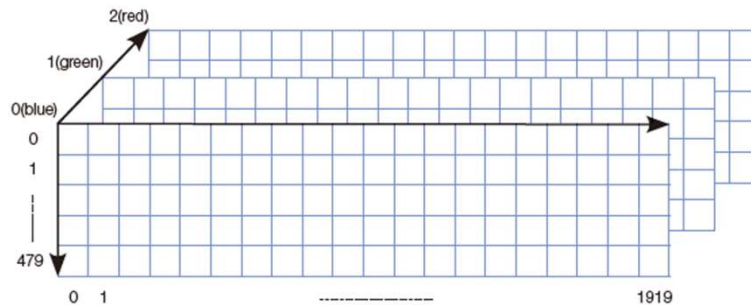
(실습)비디오에서 영상(이미지) 수집하기



(실습)비디오에서 영상(이미지) 수집하기



(a) frames 리스트



(b) imgs 배열

```
In [1]: len(frames)
7
In [2]: frames[0].shape
(480,640,3)
In [3]: type(imgs)
numpy.ndarray
In [4]: imgs.shape
(480,1920,3)
```

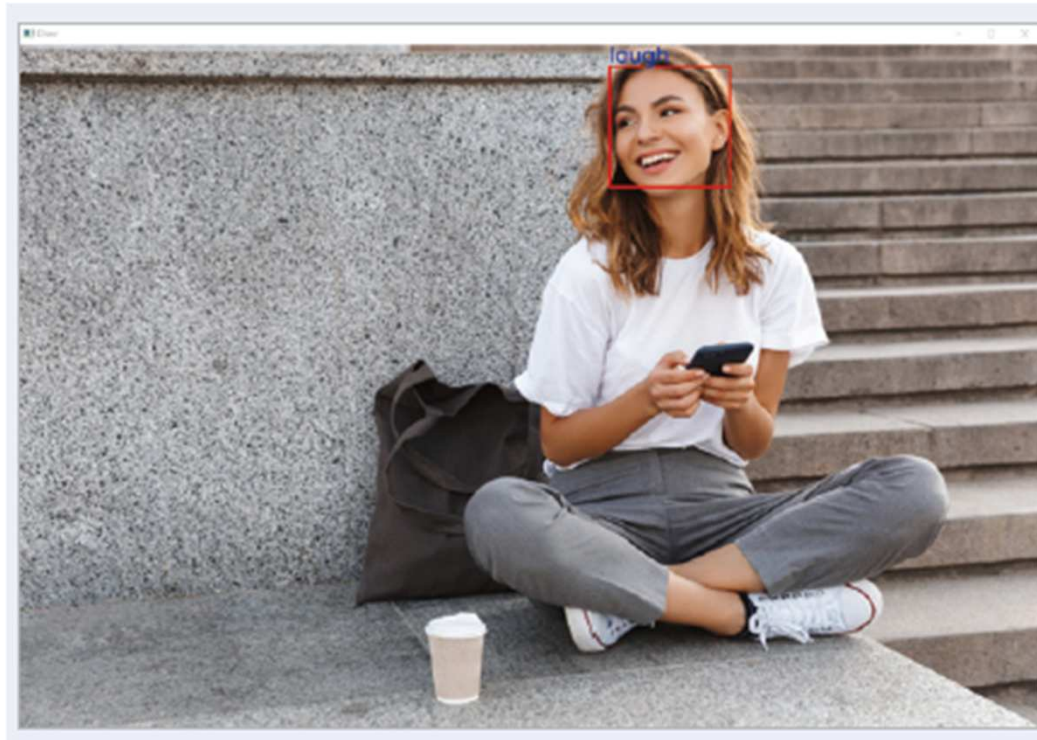
그림 2-11 [프로그램 2-5]의 자료 구조



사용자 인터페이스 구현하기

(실습)영상에 도형 그리고 글자 쓰기

- OpenCV의 그래픽 기능
 - 영상에 글씨나 도형을 넣는데 유용
 - line, rectangle, polylines, circle, ellipse, putText 함수



(실습)영상에 도형 그리고 글자 쓰기



(실습)영상에 도형 그리고 글자 쓰기

◆ rectangle() [1/2]

```
void cv::rectangle ( InputOutputArray img,
                    Point pt1,
                    Point pt2,
                    const Scalar & color,
                    int thickness = 1 ,
                    int lineType = LINE_8 ,
                    int shift = 0
                  )
```

Python:

```
cv.rectangle( img, pt1, pt2, color[, thickness[, lineType[, shift]]] ) -> img
cv.rectangle( img, rec, color[, thickness[, lineType[, shift]]] ) -> img
```

#include <opencv2/imgproc.hpp>

Draws a simple, thick, or filled up-right rectangle.

The function `cv::rectangle` draws a rectangle outline or a filled rectangle whose two opposite corners are `pt1` and `pt2`.

Parameters

img	Image.
pt1	Vertex of the rectangle.
pt2	Vertex of the rectangle opposite to <code>pt1</code> .
color	Rectangle color or brightness (grayscale image).
thickness	Thickness of lines that make up the rectangle. Negative values, like <code>FILLED</code> , mean that the function has to draw a filled rectangle.
lineType	Type of the line. See LineTypes
shift	Number of fractional bits in the point coordinates.

그림 2-12 OpenCV 공식 사이트가 제공하는 rectangle 함수의 선언(https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga07d2f74cadcf8e305e810ce8eed13bc9)

(실습)영상에 도형 그리고 글자 쓰기

프로그램 2-6

영상에 도형을 그리고 글자 쓰기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 cv.rectangle(img,(830,30),(1000,200),(0,0,255),2) # 직사각형 그리기
10 cv.putText(img,'laugh',(830,24),cv.FONT_HERSHEY_SIMPLEX,1,(255,0,0),2) # 글씨 쓰기
11
12 cv.imshow('Draw',img)
13
14 cv.waitKey()
15 cv.destroyAllWindows()
```



red

굵기

blue

(실습)마우스를 통한 상호작용

- 콜백함수(Callback function)
- : 어떤 이벤트가 발생했을 때 자동으로 불러지는 함수

[이벤트의 예]

- cv2.EVENT_LBUTTONDOWN : 마우스 왼쪽 버튼 눌림
- cv2.EVENT_LBUTTONUP : 마우스 왼쪽 버튼 땀
- cv2.EVENT_RBUTTONDOWN : 마우스 오른쪽 버튼 눌림
- cv2.EVENT_MOUSEMOVE : 마우스 이동
- cv2.EVENT_LBUTTONDBLCLK : 왼쪽 버튼 더블클릭

(실습)마우스를 통한 상호작용

프로그램 2-7

마우스로 클릭한 곳에 직사각형 그리기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 def draw(event,x,y,flags,param):      # 콜백 함수
10     if event==cv.EVENT_LBUTTONDOWN:  # 마우스 왼쪽 버튼 클릭했을 때
11         cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
12     elif event==cv.EVENT_RBUTTONDOWN: # 마우스 오른쪽 버튼 클릭했을 때
13         cv.rectangle(img,(x,y),(x+100,y+100),(255,0,0),2)
14
15     cv.imshow('Drawing',img)
16
17 cv.namedWindow('Drawing')
18 cv.imshow('Drawing',img)
19
20 cv.setMouseCallback('Drawing',draw)  # Drawing 윈도우에 draw 콜백 함수 지정
21
22 while(True):
23     if cv.waitKey(1)==ord('q'):
24         cv.destroyAllWindows()
25     break
```



(실습)마우스 드래그로 도형 크기 조절하기

프로그램 2-8

마우스 드래그로 직사각형 그리기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 def draw(event,x,y,flags,param):
10     global ix,iy
11
12     if event==cv.EVENT_LBUTTONDOWN: # 마우스 왼쪽 버튼 클릭했을 때 초기 위치 저장
13         ix,iy=x,y
14     elif event==cv.EVENT_LBUTTONUP: # 마우스 왼쪽 버튼 클릭했을 때 직사각형 그리기
15         cv.rectangle(img,(ix,iy),(x,y),(0,0,255),2)
16
17     cv.imshow('Drawing',img)
18
19 cv.namedWindow('Drawing')
20 cv.imshow('Drawing',img)
21
22 cv.setMouseCallback('Drawing',draw)
23
24 while(True):
25     if cv.waitKey(1)==ord('q'):
26         cv.destroyAllWindows()
27         break
```



(실습)페인팅 기능

프로그램 2-9

빨간색 붓과 파란색 붓으로 페인팅하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 BrushSiz=5                                # 붓의 크기
10 LColor,RColor=(255,0,0),(0,0,255)         # 파란색과 빨간색
```

(실습)페인팅 기능

```
11
12 def painting(event,x,y,flags,param):
13     if event==cv.EVENT_LBUTTONDOWN:
14         cv.circle(img,(x,y),BrushSiz,LColor,-1)    # 마우스 왼쪽 버튼 클릭하면 파란색
15     elif event==cv.EVENT_RBUTTONDOWN:
16         cv.circle(img,(x,y),BrushSiz,RColor,-1)    # 마우스 오른쪽 버튼 클릭하면 빨간색
17     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_LBUTTON:
18         cv.circle(img,(x,y),BrushSiz,LColor,-1)    # 왼쪽 버튼 클릭하고 이동하면 파란색
19     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_RBUTTON:
20         cv.circle(img,(x,y),BrushSiz,RColor,-1)    # 오른쪽 버튼 클릭하고 이동하면 빨간색
21
22     cv.imshow('Painting',img)                        # 수정된 영상을 다시 그림
23
24 cv.namedWindow('Painting')
25 cv.imshow('Painting',img)
26
27 cv.setMouseCallback('Painting',painting)
28
29 while(True):
30     if cv.waitKey(1)==ord('q'):
31         cv.destroyAllWindows()
32         break
```


(실습)페인팅 기능

