

OpenCV 트랙바 이벤트 제어

한림대학교 소프트웨어융합대학
박섭형

2022년 1학기

1 트랙바 이벤트 제어

- Trackbar는 일종의 슬라이더바(slider bar)
- `cv2.createTrackbar(trackbarname, winname, value, count, onChange) -> None`
 - 파라미터
 - * trackbarname: 트랙바의 이름
 - * winname: 트랙바가 포함되어 있는 윈도우의 이름
 - * value: 슬라이더의 초기 위치를 나타내는 정수
 - * count: 슬라이더의 최댓값 (최솟값은 항상 0)
 - * onChange: 트랙바의 슬라이더 변경 이벤트를 처리할 때 사용하는 콜백 함수
 - 반환값: None
- `onChange(pos) -> None`
 - 파라미터
 - * pos: 슬라이더의 위치를 나타내는 정수
 - 반환값: None
- `cv2.getTrackbarPos(trackbarname, winname) -> retval`
 - 지정한 트랙바의 슬라이더 위치 반환
 - 파라미터
 - * trackbarname: 트랙바의 이름
 - * winname: 트랙바가 포함되어 있는 윈도우의 이름
 - 반환값: 슬라이더의 위치
- `cv2.setTrackbarPos(trackbarname, winname) -> None`

- 지정한 트랙바의 슬라이더 위치 설정
- 파라미터
 - * trackbarname: 트랙바의 이름
 - * winname: 트랙바가 포함되어 있는 윈도우의 이름
- 반환값: None

```
[1]: import numpy as np
import cv2
```

```
[3]: def onChange(val):
    global img
    img[:, :] = val
    cv2.imshow(winname, img)

img = np.zeros((300,400), np.uint8)
winname = 'Gray color with a Trackbar'
img[:, :] = 128
cv2.imshow(winname, img)

cv2.createTrackbar("gray", winname, 128, 255, onChange)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
[4]: def onChangeRed(val):
    global img
    img[:, :, 2] = val
    cv2.imshow(winname, img)

def onChangeGreen(val):
    global img
    img[:, :, 1] = val
    cv2.imshow(winname, img)

def onChangeBlue(val):
    global img
    img[:, :, 0] = val
    cv2.imshow(winname, img)
```

```

img = np.zeros((300,400,3), np.uint8)
winname = 'Color with a Trackbar'
cv2.imshow(winname, img)

cv2.createTrackbar("red", winname, 0, 255, onChangeRed)
cv2.createTrackbar("green", winname, 0, 255, onChangeGreen)
cv2.createTrackbar("blue", winname, 0, 255, onChangeBlue)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

2 마우스와 트랙바 이벤트 사용하기

```

[6]: def onChange(val):
    global img
    img[:, :] = val
    cv2.imshow(winname, img)

def onMouse(event, x, y, flags, param):
    global img

    if event == cv2.EVENT_LBUTTONDOWN:
        if img[0,0] >= 10:
            img = img - 10
        else:
            img[:, :] = 0
    elif event == cv2.EVENT_RBUTTONDOWN:
        if img[0,0] <= 245:
            img = img + 10
        else:
            img[:, :] = 255

    cv2.setTrackbarPos("gray", winname, img[0,0])
    cv2.imshow(winname, img)

```

```

img = np.zeros((300,400), np.uint8)
winname = 'Gray color with a Trackbar'
img[:,:] = 128
cv2.imshow(winname, img)

cv2.createTrackbar("gray", winname, 128, 255, onChange)
cv2.setMouseCallback(winname, onMouse)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

```

[10]: def onChange(val):
    global img
    img[:,:] = val
    cv2.imshow(winname, img)

def onMouse(event, x, y, flags, param):
    global img

    if event == cv2.EVENT_LBUTTONDOWN and flags & cv2.EVENT_FLAG_ALTKEY:
        if img[0,0] >= 10:
            img = img - 10
        else:
            img[:,:] = 0
    elif event == cv2.EVENT_RBUTTONDOWN and flags & cv2.EVENT_FLAG_ALTKEY:
        if img[0,0] <= 245:
            img = img + 10
        else:
            img[:,:] = 255

    cv2.setTrackbarPos("gray", winname, img[0,0])
    cv2.imshow(winname, img)

img = np.zeros((300,400), np.uint8)
winname = 'Gray color with a Trackbar'

```

```
img[:, :] = 128
cv2.imshow(winname, img)

cv2.createTrackbar("gray", winname, 128, 255, onChange)
cv2.setMouseCallback(winname, onMouse)
cv2.waitKey(0)
cv2.destroyAllWindows()
```