OpenCV 키보드/마우스 이벤트 제어

한림대학교 소프트웨어융합대학
박섭형

2022년 1학기

1 OpenCV 이벤트 처리 함수

- 이벤트: 프로그램이 감지하고 처리할 수 있는 동작이나 사건
  - 사용자가 키보드의 키를 누르는 이벤트
  - 마우스를 움직이거나 마우스 버튼을 누르는 이벤트
  - 타이머(timer)나 통신 포트와 같은 하드웨어 장치가 발생시키는 이벤트
  - 사용자가 자체적으로 정의하는 이벤트
- 일반적으로 이벤트를 처리하기 위해 콜백(callback) 함수 사용
  - 개발자가 이벤트 별로 함수를 등록하면, 이벤트가 발생하거나 특정 시점에 도달했을 때 시스템이 개발자가 등록한 함수 호출
- OpenCV에서도 기본적인 이벤트 처리 함수 지원
  - 키보드 이벤트, 마우스 이벤트, 트랙바(trackbar) 이벤트

2 키보드 이벤트 처리리

- cv2.waitKey([, delay]) –> retval
  - 최소한 delay (ms) 시간 동안 키 입력을 대기하고, 키 이벤트가 발생하면 해당 키의 코드 반환
  - delay ≤ 0
    - 키 이벤트 발생까지 무한 대기
    - 키 이벤트가 발생하면 해당 키의 코드 반환 후 종료
  - dealy > 0: 지연시간 동안 키 입력을 대기하다가
    - 지연 시간 안에 키 이벤트가 발생하면 해당 키의 코드 반환 후 종료

∗ 지연 시간 안에 키 이벤트가 발생하지 않으면면 -1 반환 후 종료

- cv2.waitKeyEx([, delay]) –> retval
  - cv2.waitKey 함수와 동일하지만, 전체 키 코드 (full key code)를 반환

[1]:
```
import sys
sys.version
```

[1]:
```
'3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]'
```

[2]:
```
import cv2
cv2.__version__
```

[2]:
```
'4.5.2'
```

[4]:
```
img = cv2.imread("flower512.png")
cv2.imshow("Image", img)
while True:
    ret = cv2.waitKey()
    retEx = cv2.waitKeyEx()
    print(ret, retEx)
    if ret == ord('q'):
        break
cv2.destroyAllWindows()
```

```
39 39
182 182
172 172
37 37
39 39
38 38
40 40
18 45
45 45
45 45
46 46
36 36
45 45
36 36
```

```
16 16
97 16
97 97
16 97
16 97
16 97
16 97
16 97
97 16
18 18
18 97
91 32
8 97
46 97
35 97
34 97
33 97
20 97
20 97
20 97
98 118
20 100
20 20
113 113
```

3 OpenCV 마우스 이벤트 처리 함수

- OpenCV가 제공하는 콜백(callback) 함수로 처리
  - 마우스 이벤트를 처리하는 콜백 함수를 사용자가 작성
  - 이 함수를 cv2.setMouseCallback() 함수를 통해서 시스템에 등록
  - 프로그램 실행 과정에서 시스템이 마우스 이벤트를 감지했을 때 사용자가 등록한 콜백 함수를 호출
- cv2.setMouseCallback(winname, onMouse, param=None) –> None
  - 파라미터
    - winname: 마우스 이벤트 발생을 확인할 윈도우의 이름 (문자열)

    * onMouse: 마우스 이벤트를 처리하는 콜백 함수 이름
    * param: 이벤트 처리 함수로 전달할 추가적인 사용자 정의 인수

- cv2.onMouse(event, x, y, flags, param=None) –> retval
  - 파라미터
    * event: 발생한 마우스 이벤트의 종류. 아래 중의 하나

Enumerator	마우스 이벤트	설명
0	cv2.EVENT_MOUSEMOVE	윈도우 위로 마우스가 움직임
1	cv2.EVENT_LBUTTONDOWN	마우스 왼쪽 버튼 누르기
2	cv2.EVENT_RBUTTONDOWN	마우스 오른쪽 버튼 누르기
3	cv2.EVENT_MBUTTONDOWN	마우스 중간 버튼 누르기
4	cv2.EVENT_LBUTTONUP	마우스 왼쪽 버튼 떼기
5	cv2.EVENT_RBUTTONUP	마우스 오른쪽 버튼 떼기
6	cv2.EVENT_MBUTTONUP	마우스 중간 버튼 떼기
7	cv2.EVENT_LBUTTONDBLCLK	마우스 왼쪽 버튼 더블클릭
8	cv2.EVENT_RBUTTONDBLCLK	마우스 오른쪽 버튼 더블클릭
9	cv2.EVENT_MBUTTONDBLCLK	마우스 중간 버튼 더블클릭
10	cv2EVENT_MOUSEWHEEL	양수는 마우스 휠 전방 스코롤, 음수는 마우스 휠 후방 스크롤
11	cv2EVENT_MOUSEHWHEEL	양수는 마우스 가로 휠 오른쪽 방향 스코롤, 음수는 마우스 가로

- `x`: 마우스 이벤트의 `x` 좌표
- `y`: 마우스 이벤트의 `y` 좌표
- `flags`: 아래 플래그 중의 하나

Enumerator	flag 상수	설명
1	cv2.EVENT_FLAG_LBUTTON	왼쪽 마우스 버튼 눌림
2	cv2.EVENT_FLAG_RBUTTON	오른쪽 마우스 버튼 눌림
4	cv2.EVENT_FLAG_MBUTTON	중간 마우스 버튼 눌림
8	cv2.EVENT_FLAG_CTRLKEY	CTRL 키 눌림
16	cv2.EVENT_FLAG_SHIFTKEY	SHIFT 키 눌림
32	cv2.EVENT_FLAG_ALTKEY	ALT 키 눌림

- `param`: 콜백 함수로 전달하 추가적인 사용자 정의 인수

[1]:
```
import cv2
```

[3]:
```
def onMouse(event, x, y, flags, param):
    global img
    if event == cv2.EVENT_LBUTTONDOWN:
        rows, cols, _ = img.shape
        rows = int(rows * 1.1)
        cols = int(cols * 1.1)
        img = cv2.resize(img, (cols, rows))
        cv2.imshow("bird", img)
    elif event == cv2.EVENT_RBUTTONDOWN:
        rows, cols, _ = img.shape
        rows = int(rows / 1.1)
        cols = int(cols / 1.1)
        img = cv2.resize(img, (cols, rows))
        cv2.imshow("bird", img)


img = cv2.imread("bird.jpg")
cv2.namedWindow("bird")
cv2.imshow("bird", img)

cv2.setMouseCallback("bird", onMouse)
cv2.waitKey(0)
cv2.destroyAllWindows()
```