

CH03. OpenCV: 비디오 reading과 writing

3. 비디오 reading과 writing

- 3.1 테스트 환경

순 번	제 목	설치 버전
1	운영 체 제	Windows 10 64bit
2	프로그래밍 언어	Python3.6.5

3. 비디오 Reading과 Writing

▪ 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

- 다음은 03-01 example.py 예제의 code이다.

```
1 import numpy as np
2 import cv2
3
4 def showVideo():
5     try:
6         print('카메라를 구동합니다.')
7         cap = cv2.VideoCapture(0)
8     except:
9         print('카메라 구동 실패')
10        return
11
12    cap.set(3,680)
13    cap.set(4,320)
```

[그림] 03-01 example.py의 code (1/3)

```
14
15 while True:
16     ret, frame = cap.read()
17
18     if not ret :
19         print('비디오 읽기 오류')
20         break
21
22     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
23     cv2.imshow('video',gray)
24
25     k = cv2.waitKey(1) & 0xFF
26     if k == 27:
27         break
```

[그림] 03-01 example.py의 code (2/3)

3. 비디오 Reading과 Writing

- 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

```
27  
28     cap.release()  
29     cv2.destroyAllWindows()  
30  
31     showVideo()
```

[그림] 03-01 example.py의 code (3/3)

3. 비디오 Reading과 Writing

- 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이
 - 다음은 03-01 example.py 예제의 code와 실행 결과로 나오는 video 화면이다. 뒷장 부터 03-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 03-01 example.py의 실행 결과

3. 비디오 Reading과 Writing

▪ 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

>> LINE 7) `cap = cv2.VideoCapture(0)`

- 먼저, 비디오 캡처를 위해 VideoCapture 객체를 생성해야 한다. VideoCapture의 인자로는 장치 인덱스 또는 비디오파일 이름을 지정한다. 장치 인덱스는 어떤 카메라를 지정할 것인지에 대한 숫자이다. 컴퓨터에 웹캠이 하나 부착되어 있으면 0을 지정하고, 2개 이상일 경우에는 첫번째 웹캠은 0, 두번째 웹캠은 1 등으로 지정하면 된다.

>> LINE 12) `ret = cap.set(3, 680)`

>> LINE 13) `ret = cap.set(4, 320)`

- 비디오 프레임의 폭과 높이를 설정한다. 폭을 설정 할 때는 3번을, 높이를 설정 할 때는 4번을 통해서 세팅한다.

3. 비디오 Reading과 Writing

- 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

>> LINE 15) while True:

- 라이브로 들어오는 비디오를 프레임별로 캡처하고 이를 화면에 디스플레이 해주어야 하므로, 특정 키를 누를 때 까지 무한 루프를 돌린다.

3. 비디오 Reading과 Writing

▪ 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

>> LINE 16) `ret, frame = cap.read()`

- `cap.read()`는 재생되는 비디오의 한 프레임씩 읽는다. 여기서 프레임은 화면에 뿌려지는 정지 영상의 낱장을 뜻한다. 비디오 프레임을 제대로 읽었다면 `ret` 값이 `True`가 되고, 실패하면 `False`가 된다. 필요한 경우, `ret` 값을 체크하여 비디오 프레임을 제대로 읽었는지 확인할 수 있다.
- 가끔 `cap`이 제대로 초기화되지 않을 수 있는데, 이런 경우 에러코드를 리턴한다. 이 때 `cap.isOpened()` 함수를 이용해 `cap`이 초기화가 제대로 되었는지 확인할 수 있다. 만약 `cap.isOpened()`가 `False`이면 `cap.open()` 함수를 이용해 오픈하면 된다.

3. 비디오 Reading과 Writing

- 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

>> LINE 21) `gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`

- `frame`을 흑백으로 변환해 준다.

>> LINE 22) `cv2.imshow('frame', gray)`

- 흑백으로 변환한 프레임을 화면에 디스플레이 합니다. 비디오는 프레임을 시간에 따라 연속적으로 화면에 보여주는 것이므로, 하나의 프레임은 하나의 이미지이다. 따라서 이미지를 화면에 디스플레이하는 함수인 `cv2.imshow()`를 그대로 사용한다.

3. 비디오 Reading과 Writing

- 3.2 [03-01 example.py] – 카메라 흑백 영상 디스플레이

>> LINE 28) `cap.release()`

- 마지막으로 오픈한 `cap` 객체를 `cap.release()` 함수를 이용해 반드시 해제한다.

>> LINE 29) `cv2.destroyAllWindows()`

- 생성한 모든 윈도우를 제거한다.

3. 비디오 Reading과 Writing

- 3.2 [03-02 example.py] – 비디오 녹화
 - 다음은 03-02 example.py 예제의 code이다.

```
1 import numpy as np
2 import cv2
3
4 def writeVideo():
5     try:
6         print('카메라를 구동합니다.')
7         cap = cv2.VideoCapture(0)
8     except:
9         print('카메라 구동 실패')
10        return
11
12    fps = 20.0
13    width = int(cap.get(3))
14    height = int(cap.get(4))
```

[그림] 03-02 example.py의 code (1/3)

```
15    fcc = cv2.VideoWriter_fourcc('D','T','V','X')
16
17    out = cv2.VideoWriter('mycam.avi', fcc, fps, (width, height))
18    print('녹화를 시작합니다.')
19
20    while True:
21        ret, frame = cap.read()
22
23        if not ret :
24            print('비디오 읽기 오류')
25            break
26
```

[그림] 03-02 example.py의 code (2/3)

3. 비디오 Reading과 Writing

- 3.2 [03-02 example.py] – 비디오 녹화
 - 다음은 03-02 example.py 예제의 code이다.

```
27     cv2.imshow('video',frame)
28     out.write(frame)
29
30     k = cv2.waitKey(1) & 0xFF
31     if k == 27:
32         print('녹화를 종료합니다')
33         break
```

[그림] 03-02 example.py의 code (3/3)

3. 비디오 Reading과 Writing

- 3.2 [03-02 example.py] – 비디오 녹화

```
>> LINE 12) fps = 20.0
```

- 비디오 저장을 위해 초당 프레임 수를 20.0으로 지정한다.

```
>> LINE 13) width = int(cap.get(3))
```

```
>> LINE 14) height = int(cap.get(4))
```

- 원본 크기 그대로 width와 height를 지정한다.

3. 비디오 Reading과 Writing

▪ 3.2 [03-02 example.py] – 비디오 녹화

>> LINE 15) `fcc = cv2.VideoWriter_fourcc('D','I','V','X')`

- DIVX 코덱을 적용하기 위해, `cv2.VideoWriter_fourcc()` 함수의 인자로 'D', 'I', 'V', 'X'를 적용했다. 코덱이란 영상 또는 음성 등의 아날로그 신호를 디지털 방식으로 변환하는 코더와 디지털 신호를 영상이나 음성으로 바꿔주는 디코더의 합성이다. 여기서 적용 가능한 코덱은 "DIVX", "XVID", "MJPG", "X264", "WMV1", "WMV2"이다.

>> LINE 17) `out = cv2.VideoWriter('mycam.avi', fcc, fps, (width, height))`

- 비디오 저장을 위한 객체를 생성한다. 첫번째 인자에 저장할 파일명을 지정해 준다.

3. 비디오 Reading과 Writing

- 3.2 [03-02 example.py] – 비디오 녹화

>> LINE 27) `cv2.imshow('video',frame)`

- 비디오 프레임을 제대로 읽었다면 `cv2.imshow()` 함수로 화면에 해당 프레임을 디스플레이 한다.

>> LINE 28) `out.write(frame)`

- `out.write()` 함수로 'mycam.avi'에 프레임을 저장한다.

3. 비디오 Reading과 Writing

▪ 3.2 [03-02 example.py] – 비디오 녹화

```
>> LINE 35) cap.release()
```

```
>> LINE 36) out.release()
```

```
>> LINE 37) cv2.destroyAllWindows()
```

- ESC 키를 누르면 무한 루프를 빠져나오고 cap과 out 객체를 해제해주고 프로그램을 종료한다.

```
>> 참고 사항) frame = cv2.flip(frame, 0)
```

- 이 코드를 적용 시 재생되는 화면이 거꾸로 나오며, 저장되는 비디오 또한 거꾸로 저장된다.