

메디치소프트 기술연구소

## CH18. OpenCV: 이미지 Contour 응용1

# 18. 이미지 Contour 응용1

- 18.1 테스트 환경

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

## ■ 18.2 이미지 모멘트

- 이번 단원에서는 원이나 타원과 같이 닫혀 있는 폐곡선인 Contour의 면적, Contour의 둘레 길이, Contour의 중심 등과 같이 Contour의 특성을 살펴본다.

- 이미지 모멘트(Image Moments)

이미지 모멘트는 객체의 무게중심, 객체의 면적 등과 같은 특성을 계산할 때 유용하다.

OpenCV의 `cv2.moments()` 함수는 이미지 모멘트를 계산하고 이를 사전형 자료에 담아 리턴한다. 모멘트 종류는 3가지이며 아래와 같이 총 24개의 값을 가진다.

1) 공간 모멘트 :  $m_{00}$ ,  $m_{10}$ ,  $m_{01}$ ,  $m_{20}$ ,  $m_{11}$ ,  $m_{02}$ ,  $m_{30}$ ,  $m_{21}$ ,  $m_{12}$ ,  $m_{03}$

2) 중심 모멘트 :  $\mu_{20}$ ,  $\mu_{11}$ ,  $\mu_{02}$ ,  $\mu_{30}$ ,  $\mu_{21}$ ,  $\mu_{12}$ ,  $\mu_{03}$

3) 평준화된 중심 모멘트 :  $\nu_{20}$ ,  $\nu_{11}$ ,  $\nu_{02}$ ,  $\nu_{30}$ ,  $\nu_{21}$ ,  $\nu_{03}$

## ■ 18.2 이미지 모멘트

- `cv2.moments()` 함수의 인자는 1개로,  $1 \times N$  또는  $N \times 1$  크기의 Numpy array 이다.  
`cv2.findContours()` 함수는 이미지에서 contour를 찾은 후 리스트형 자료에 담아 리턴한다.  
하나의 Contour는  $1 \times N$  크기의 Numpy Array 이다.
- 1) `cv2.findContours()` 함수로 이미지 contour들을 찾음
  - 2) 찾은 contour에서 이미지 모멘트를 구하고자 하는 contour 1개를 정함
  - 3) 이 contour를 `cv2.moments()` 함수의 인자로 전달하여 이미지 모멘트를 구함

## ■ 18.3 [18-01 example.py] – Contour

- 다음은 18-01 example.py 예제의 code이다.

```
import numpy as np
```

```
import cv2
```

```
def moment():
```

```
    img = cv2.imread('images/model.jpg')
```

```
    imgray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
    ret, thr = cv2.threshold(imgray, 127, 255, 0)
```

```
    _, contours, _ = cv2.findContours(thr, cv2.RETR_TREE,  
                                     cv2.CHAIN_APPROX_SIMPLE)
```

## ■ 18.3 [18-01 example.py] – Contour

- 다음은 18-01 example.py 예제의 code이다.

```
contour = contours[0]
```

```
    mmt = cv2.moments(contour)
```

```
for key, val in mmt.items():
```

```
    print('%s: %f' % (key, val))
```

```
cx = int(mmt['m10']/mmt['m00'])
```

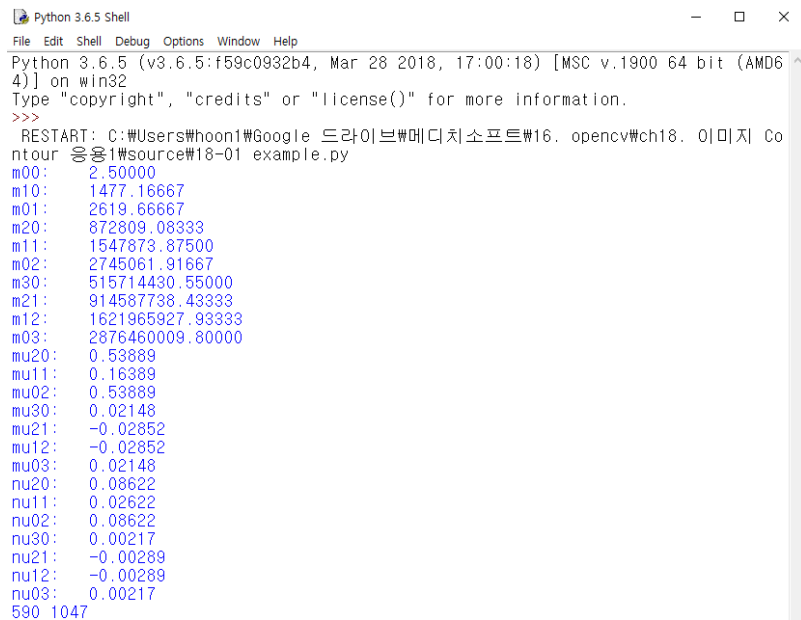
```
cy = int(mmt['m01']/mmt['m00'])
```

```
print(cx, cy)
```

```
moment()
```

## ■ 18.3 [18-01 example.py] – Contour

- 다음은 18-01 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 18-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\hhoon1\Google 드라이브\메디치소프트\16. opencv\ch18. 이미지 Co
ntour 응용1\source\18-01 example.py
m00: 2.50000
m10: 1477.16667
m01: 2619.66667
m20: 872809.08333
m11: 1547873.87500
m02: 2745061.91667
m30: 515714430.55000
m21: 914587738.43333
m12: 1621965927.93333
m03: 2876460009.80000
mu20: 0.53889
mu11: 0.16389
mu02: 0.53889
mu30: 0.02148
mu21: -0.02852
mu12: -0.02852
mu03: 0.02148
nu20: 0.08622
nu11: 0.02622
nu02: 0.08622
nu30: 0.00217
nu21: -0.00289
nu12: -0.00289
nu03: 0.00217
590 1047
```

[그림] 18-01 example 실행 결과

## ■ 18.3 [18-01 example.py] – Contour

```
>> LINE 12) contour = contours[0]
```

```
>> LINE 13) mmt = cv2.moments(contour)
```

- cv2.findContours() 함수로 찾은 contour들 중, 첫번째 contour에 대한 이미지 모멘트를 구한다. 그후, 계산된 이미지 모멘트를 화면에 출력해보면 이미지 모멘트 종류에 따른 여러가지 모멘트 값들이 화면에 출력된다.

```
>> LINE 18) cx = int(mmt['m10']/mmt['m00'])
```

```
>> LINE 19) cy = int(mmt['m01']/mmt['m00'])
```

- 기억해야 할 것은 Contour의 무게중심을 구하는 방법이다. 위 코드는 무게 중심의 x좌표와 y좌표를 구하는 식이다. 이 식은 무게중심을 구하는 경우 항상 변함없이 같다.



## ■ 18.4 [18-02 example.py] – Contour

- 다음은 18-02 example.py 예제의 code이다.

```
import numpy as np
```

```
import cv2
```

```
def contour():
```

```
    img = cv2.imread('images/model.jpg')
```

```
    imgray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
    ret, thr = cv2.threshold(imgray, 127, 255, 0)
```

```
    _, contours, _ = cv2.findContours(thr, cv2.RETR_TREE,  
                                     cv2.CHAIN_APPROX_SIMPLE)
```

## ■ 18.4 [18-02 example.py] – Contour

- 다음은 18-02 example.py 예제의 code이다.

```
cnt = contours[164]
    area = cv2.contourArea(cnt)
    perimeter = cv2.arcLength(cnt, True)
    cv2.drawContours(img, [cnt], 0, (255, 255, 0), 1)
    print('contour 면적: ', area)
    print('contour 길이: ', perimeter)
    cv2.imshow('contour', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

contour()

## ■ 18.4 [18-02 example.py] – Contour

- 다음은 18-02 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 18-02 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 18-02 example 실행 결과(1)

```

*Python 3.6.5 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18)
4)] on win32
Type "copyright", "credits" or "license()" for more info
>>>
RESTART: C:\Users\Whoon1\Google 드라이브\메디치소프트\16
ntour 응용1\source\18-02 example.py
contour 면적: 7.5
contour 길이: 11.071067690849304
  
```

[그림] 18-02 example 실행 결과(2)

## ■ 18.4 [18-02 example.py] – Contour

```
>> LINE 16) cv2.drawContours(img, [cnt], 0, (255, 255, 0), 1)
```

- 위 코드는 `cnt = contours[164]` 즉, 165번째 Contour의 모든 픽셀을 두께1, 하늘색 선으로 `img` 위에 드로잉하라는 의미이다.

```
>> LINE 18) cx = int(mmt['m10']/mmt['m00'])
```

```
>> LINE 19) cy = int(mmt['m01']/mmt['m00'])
```

- 기억해야 할 것은 Contour의 무게중심을 구하는 방법이다. 위 코드는 무게 중심의 x좌표와 y좌표를 구하는 식이다. 이 식은 무게중심을 구하는 경우 항상 변함없이 같다.

```
import numpy as np
import cv2

def moment():
    img = cv2.imread('images/model.jpg')
    imggray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    ret, thr = cv2.threshold(imggray, 127, 255, 0)
    _, contours, _ = cv2.findContours(thr, cv2.RETR_TREE,
                                     cv2.CHAIN_APPROX_SIMPLE)

    contour = contours[0]
    mmt = cv2.moments(contour)

    for key, val in mmt.items():
        print('%s: %t%.5f' % (key, val))

    cx = int(mmt['m10']/mmt['m00'])
    cy = int(mmt['m01']/mmt['m00'])

    print(cx, cy)

moment()
```

```
import cv2

def contour():
    img = cv2.imread('images/model.jpg')
    imggray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    ret, thr = cv2.threshold(imggray, 127, 255, 0)
    _, contours, _ = cv2.findContours(thr, cv2.RETR_TREE,
                                     cv2.CHAIN_APPROX_SIMPLE)

    cnt = contours[164]
    area = cv2.contourArea(cnt)
    perimeter = cv2.arcLength(cnt, True)

    cv2.drawContours(img, [cnt], 0, (255, 255, 0), 1)

    print('contour 면적: ', area)
    print('contour 길이: ', perimeter)

    cv2.imshow('contour', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

contour()
```