

메디치소프트 기술연구소

CH17. OpenCV: 이미지 Contour

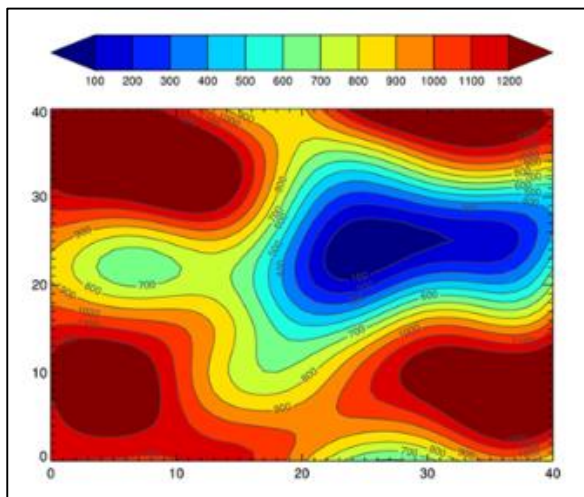
17. 이미지 Contour

- 17.1 테스트 환경

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

■ 17.2 이미지 Contour란?

- Canny Contour란 같은 값을 가진 곳을 연결한 선이라고 생각하면 됩니다. 우리 주위에 자주 접할 수 있는 Contour의 예로 지도에서 같은 높이를 가진 지점을 연결한 등고선, 기상도에서 같은 기압을 가진 곳을 연결한 등압선 등이 있다. 아래 그림은 Contour의 예시 이다.



[그림] Contour 예시

■ 17.2 이미지 Contour란?

- 이미지 Contour란 동일한 색 또는 동일한 색상 강도(Color Intensity)를 가진 부분의 가장 자리 경계를 연결한 선이다. 이미지 Contour는 이미지에 있는 물체의 모양 분석이나 객체 인식 등에 유용하게 활용되는 도구이다.
- 정확한 이미지 Contour를 확보하기 위해 바이너리 이미지를 사용한다. 즉, 이미지에서 Contour를 찾기 전에 앞에서 배웠던 threshold나 canny edge detection을 적용하는 것이 좋다.
- OpenCV의 `cv2.findContours()` 함수는 Suzuki85라는 알고리즘을 이용해서 이미지에서 Contour를 찾는 함수이다. 이 함수는 원본 이미지를 변경시키기 때문에 추후 원본이미지를 활용하기 위해서는 원본 이미지의 복사본을 가지고 Contour를 찾으려 해야한다.

■ 17.3 [17-01 example.py] – Contour

- 다음은 17-01 example.py 예제의 code이다.

```
import numpy as np
import cv2
```

```
def contour():
    img = cv2.imread('images/globe.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)
```

```
cv2.drawContours(img, contours, -1, (0, 0, 255), 1)
cv2.imshow('thresh', thr)
cv2.imshow('contour', img)
```

■ 17.3 [17-01 example.py] – Contour

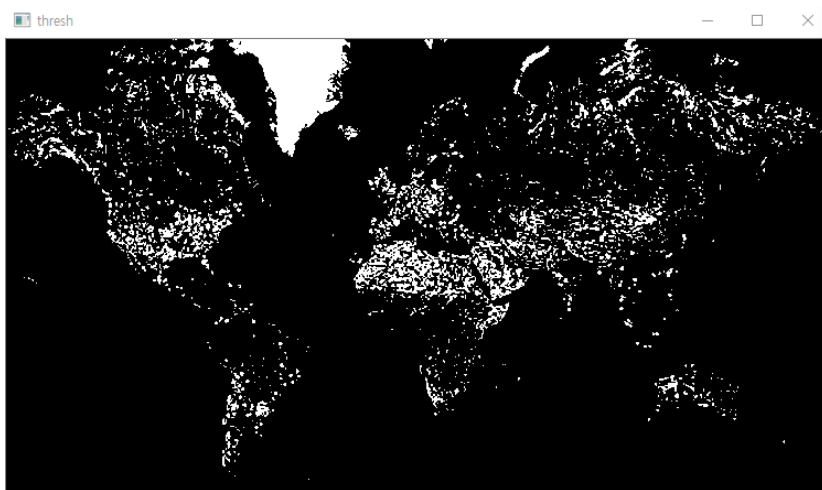
- 다음은 17-01 example.py 예제의 code이다.

```
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

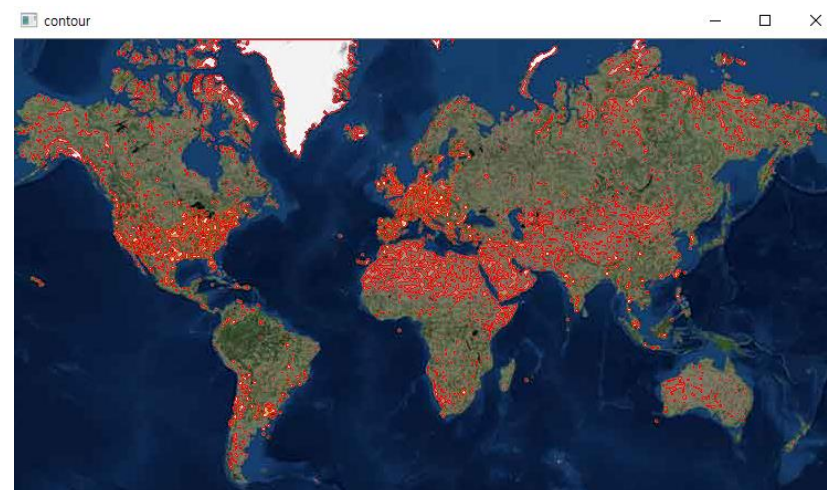
```
contour()
```

■ 17.3 [17-01 example.py] – Contour

- 다음은 17-01 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 17-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] Thresh



[그림] Contour

■ 17.3 [17-01 example.py] – Contour

- OpenCV에서 Contour 찾기는 검정색 배경에서 흰색 물체를 찾는 것과 비슷하다. Contour를 찾
고자 하는 대상은 흰색으로, 배경은 검정색으로 변경해야 함을 꼭 기억하여라

```
>> LINE 8) ret, thr = cv2.threshold(imgray, 127, 255, 0)
```

```
>> LINE 9) _, contours, _ = cv2.findContours(thr, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

- Gray 스케일로 변환시킨 imgray를 thresholding 하여 그 값을 thresh로 한다. 이를
cv2.findContours() 함수에 넘겨 contour를 찾는다.

■ 17.3 [17-01 example.py] – Contour

>> `cv2.findContours(thr, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)`

- `cv2.findContours()` 함수는 `thr`, 이미지에서 찾은 `contour`와 `contour`들 간의 계층 구조를 리턴한다. 그런데 우리는 찾은 `contour`만 관심이 있으므로 리턴값 3개 중 필요없는 녀석들은 '_' 로 리턴 받는다. 이 함수는 세 개의 인자를 가지고 있다.

1) `thresh` : `contour` 찾기를 할 소스이미지. `thresholding`을 통해 변환된 바이너리 이미지여야 한다.

2) `cv2.RETR_TREE` : 두 번째 인자는 `contour` 추출 모드이며, 2번째 리턴값인 `hierarchy`의 값에 영향을 준다.

- `cv2.RETR_EXTERNAL`: 이미지의 가장 바깥쪽의 `contour`만 추출
- `cv2.RETR_LIST`: `contour` 간 계층구조 상관관계를 고려하지 않고 `contour`를 추출
- `cv2.RETR_CCOMP`: 이미지에서 모든 `contour`를 추출한 후, 2단계 `contour` 계층 구조로 구성함
1단계 계층에서는 외곽 경계 부분을, 2단계 계층에서는 구멍(hole)의 경계 부분을 나타내는 `contour`로 구성됨

■ 17.3 [17-01 example.py] – Contour

3) cv2.CHAIN_APPROX_SIMPLE : 세 번째 인자는 contour 근사 방법

- cv2.CHAIN_APPROX_NONE: contour를 구성하는 모든 점을 저장
- cv2.CHAIN_APPROX_SIMPLE: contour의 수평, 수직, 대각선 방향의 점은 모두 버리고 끝 점만 남겨둔다.
- cv2.CHAIN_APPROX_TC89_1: Teh-Chin 연결 근사 알고리즘

■ 17.3 [17-01 example.py] – Contour

>> LINE 11) `cv2.drawContours(img, contours, -1, (0, 0, 255), 1)`

- `cv2.drawContours()` 함수는 우리가 찾은 contour를 실제로 그리는 함수이며, 5개의 인자를 가지고 있다.

- 1) `img`: contour를 나타낼 대상 이미지
- 2) `contours`: `img`에 그릴 contour. 이 값은 `cv2.findContours()` 함수의 2번째 리턴 값으로 리스트형 자료. `i`번째 contour의 첫 번째 픽셀 좌표는 `contours[i][0]`과 같이 접근 가능
- 3) `-1`: `img`에 실제로 그릴 contour 인덱스 파라미터. 이 값이 음수이면 모든 contour를 그림
- 4) `(0, 255, 0)`: contour 선의 BGR 색상값. 여기서는 Green으로 지정함
- 5) `1`: contour 선의 두께

```
import numpy as np
import cv2

def contour():
    img = cv2.imread('images/globe.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)

    cv2.drawContours(img, contours, -1, (0, 0, 255), 1)
    cv2.imshow('thresh', thr)
    cv2.imshow('contour', img)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

contour()
```