메디치소프트 기술연구소

CH19. OpenCV: 이미지 Contour 응용2

MEDICISOFT

19. 이미지 Contour 응용2

- 18.1 테스트 환경

순 번	제 목	설치 버전
1	운영체제	Windows 10 64bit
2	프로그래밍언어	Python3.6.5

19.2 Convex Hull

- 이번 단원에서는 임의의 도형을 볼록체로 변형하기 위한 Convex Hull, 볼록면을 체크하는 방법, 개체에 외접하는 다각형 또는 원을 그리는 방법 등에 대해 살펴본다.

- Convex Hull(볼록체)은 결과물의 모양으로 보면 Contour 근사법과 유사하지만, 엄밀히 말하면 전혀 다른 기법이다.

- cv2.convexHull() 함수는 Contour의 오목한 부분을 체크하고 이를 보정하는 역할을 수행한다.

19.3 [19-01 example.py] – Convex Hull

- 다음은 19-01 example.py 예제의 code이다.

```
import numpy as np
import cv2
def convex():
    img = cv2.imread('images/convexhull.png')

    img1 = img.copy()
    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)
```

19.3 [19-01 example.py] – Convex Hull

- 다음은 19-01 example.py 예제의 code이다.

```
cnt = contours[1]
cv2.drawContours(img, [cnt], 0, (0,255,0), 3)

check = cv2.isContourConvex(cnt)

if not check:
    hull = cv2.convexHull(cnt)
    cv2.drawContours(img1, [hull], 0, (0,255,0), 3)
    cv2.imshow('convexhull', img1)
```

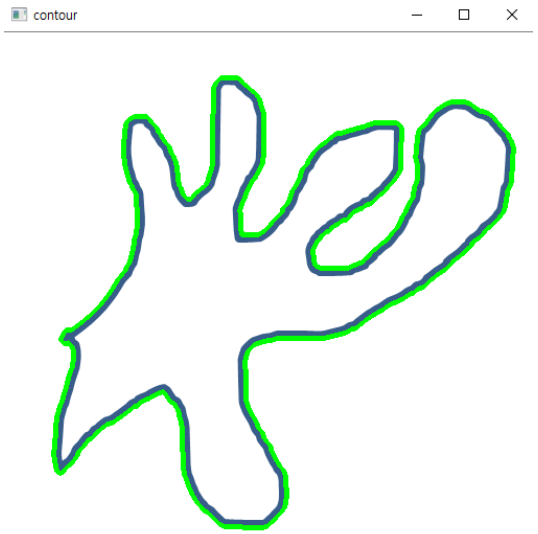
19.3 [19-01 example.py] – Convex Hull

- 다음은 19-01 example.py 예제의 code이다.

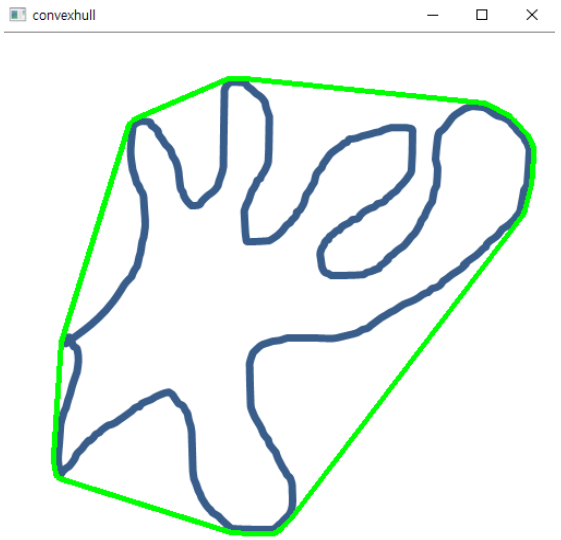
```
cv2.imshow('contour', img)


    cv2.waitKey(0)
    cv2.destroyAllWindows()


convex()
```

▪ 19.3 [19-01 example.py] – Convex Hull

- 다음은 19-01 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 19-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 19-01 example 실행 결과(1)



[그림] 19-01 example 실행 결과 (2)

▪ 19.3 [19-01 example.py] – Convex Hull

>> LINE 17) check = cv2.isContourConvex(cnt)

- cv2.isContourConvex() 함수는 인자로 입력된 Contour가 Convex Hull 인지 체크한다. 만약 Convex Hull이라면 True를 리턴하고 그렇지 않으면 False를 리턴한다. 예시의 그림은 Convex Hull이 아니므로 check 변수값은 False가 된다.

>> LINE 20) hull = cv2.convexHull(cnt)

>> LINE 21) cv2.drawContours(img1, [hull], 0, (0,255,0), 3)

- check 값이 False인 경우, cv2.convexHull() 함수를 이용해 원본이미지의 contours[1]에 대한 convex hull 곡선을 구한다. contours[1]은 원본 이미지에서 찾은 contour들 가운데 2번째 contour이다. 2번째 contou는 도형 바깥쪽을 감싸는 contou이다.

convex hull을 cv2.drawContour() 함수의 인자로 입력하여 출력해보면 contours[1]에 대한 Convex Hull을 나타내고 있음을 볼 수 있다.

19.4 [19-02 example.py] – Convex Hull

- 다음은 19-02 example.py 예제의 code이다.

```
import numpy as np
import cv2


def convex():
    img = cv2.imread('images/lightning.png')
    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)

    cnt = contours[5]
```

19.4 [19-02 example.py] – Convex Hull

- 다음은 19-02 example.py 예제의 code이다.

```
x,y,w,h = cv2.boundingRect(cnt)
cv2.rectangle(img, (x,y), (x+w, y+h), (0, 0, 255), 3)

rect = cv2.minAreaRect(cnt)
box = cv2.boxPoints(rect)
box = np.int0(box)

cv2.drawContours(img, [box], 0, (0,255,0), 3)
```

19.4 [19-02 example.py] – Convex Hull

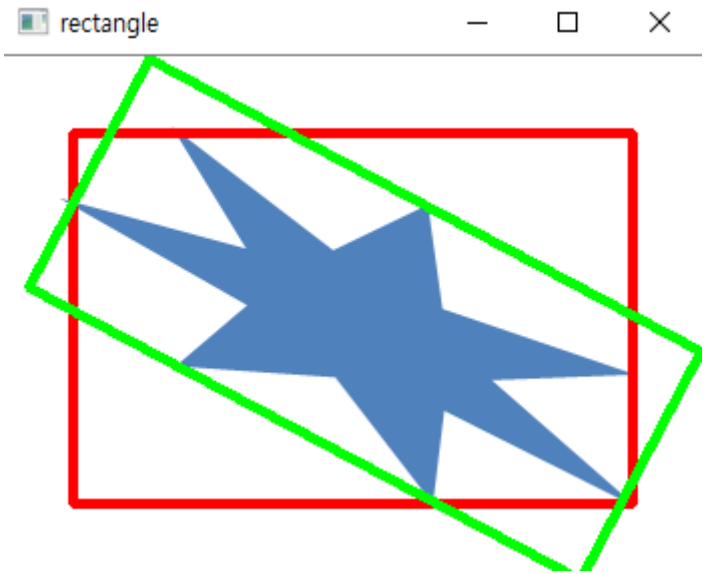
- 다음은 19-02 example.py 예제의 code이다.

```
cv2.imshow('rectangle', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

convex()

▪ 19.4 [19-02 example.py] – Convex Hull

- 다음은 19-02 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 19-02 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 19-02 example 실행 결과

19.4 [19-02 example.py] – Convex Hull

- 본 예제에서 해볼 것은 별표 모양의 그림을 에워싸는 사각형과 원, 타원을 그려보려 한다. 어떤 도형 또는 이미지를 에워싸는 도형의 종류는 아래와 같다.

  1) Bounding Rectangle

  2) Minimum Area Rectangle

  3) Minimum Enclosing Circle

  4) Fitting an Ellipse

▪ 19.4 [19-02 example.py] – Convex Hull

>> LINE 13) x,y,w,h = cv2.boundingRect(cnt)

- 원본 이미지의 6번째 contour인 contours[5]가 단풍잎 모양의 외곽을 에워싸고 있는 contour 이다. contour에 외접하는 똑바로 세워진 사각형을 얻기 위해 cv2.boundingRect() 함수를 이용한다.
- cv2.boudingRect()함수는 인자로 받은 contour에 외접하고 똑바로 세워진 직사각형의 좌상단 꼭지점 좌표 (x, y)와 가로 세로 폭을 리턴한다. 이렇게 얻은 좌표를 이용해 원본 이미지에 빨간색으로 사각형을 그린다.

▪ 19.4 [19-02 example.py] – Convex Hull

>> LINE 17) rect = cv2.minAreaRect(cnt)

>> LINE 17) box = cv2.boxPoints(rect)

>> LINE 17) box = np.int0(box)

>> LINE 17) cv2.drawContours(img, [box], 0, (0, 255, 0), 3)

- cv2.minAreaRect() 함수는 인자로 입력한 contour에 외접하면서 면적이 가장 작은 직사각형을 구하는데 활용된다. 이 함수의 리턴값은 좌상단 꼭지점 좌표 (x, y), 가로 세로 폭과 이 사각형이 기울어진 각도이다.

- cv2.boxPoints() 함수는 cv2.minAreaRect() 함수로 얻은 직사각형의 꼭지점 4개의 좌표를 얻기 위해 사용된다. 좌표는 float형으로 리턴되므로 np.int0()로 정수형 값으로 전환한 후, 원본 이미지에 초록색 사각형을 그린다.

19.5 [19-03 example.py] – Convex Hull

- 다음은 19-03 example.py 예제의 code이다.

```
import numpy as np
import cv2


def convex():
    img = cv2.imread('images/lightning.png')
    imgray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)


    rows, cols = img.shape[:2]


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

▪ 19.5 [19-03 example.py] – Convex Hull

- 다음은 19-03 example.py 예제의 code이다.

```
cnt = contours[5]

(x,y),r = cv2.minEnclosingCircle(cnt)
center = (int(x), int(y))
r = int(r)

cv2.circle(img, center, r, (255,0,0),3)

ellipse = cv2.fitEllipse(cnt)
cv2.ellipse(img,ellipse,(0,255,0),3)
```

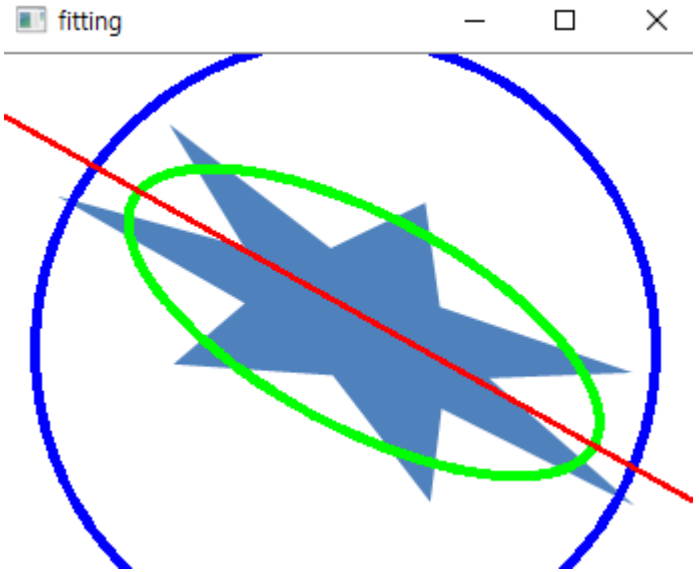
19.5 [19-03 example.py] – Convex Hull

- 다음은 19-03 example.py 예제의 code이다.

```
[vx,vy,x,y] = cv2.fitLine(cnt, cv2.DIST_L2, 0, 0.01, 0.01)
   ly = int((-x*vy/vx)+y)
   ry = int(((cols-x)*vy/vx)+y)

   cv2.line(img, (cols-1, ry), (0, ly), (0,0,255),2)

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

convex()
```

▪ 19.5 [19-03 example.py] – Convex Hull

- 다음은 19-03 example.py 예제의 실행 결과로 나오는 화면이다. 뒷장 부터 19-03 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.



[그림] 19-03 example 실행 결과

▪ 19.5 [19-03 example.py] – Convex Hull

>> LINE 15) (x,y),r = cv2.minEnclosingCircle(cnt)

>> LINE 16) center = (int(x), int(y))

>> LINE 17) r = int(r)

>> LINE 19) cv2.circle(img, center, r, (255,0,0),3)

- cv2.minEnclosingCircle() 함수는 주어진 contour에 외접하는 원을 얻기 위해 사용된다. 이 함수의 리턴값은 원의 중심 좌표와 반지름이다. cv2.minEnclosingCircle() 함수로 얻은 원은 파란색으로 원본이미지에 그려준다.

▪ 19.5 [19-03 example.py] – Convex Hull

>> LINE 21) ellipse = cv2.fitEllipse(cnt)

>> LINE 22) cv2.ellipse(img,ellipse,(0,255,0),3)

- cv2.minEnclosingCircle() cv2.fitEllipse() 함수는 주어진 Contour를 최적으로 둘러싸는 타원을 얻기 위해 사용된다. 이 함수의 리턴값은 타원 그리기 함수인 cv2.ellipse()의 인자로 바로 사용될 수 있다. 계산된 타원은 초록색으로 그려준다.

i^{th} r_i $\rho(r)$ $\sum_i \rho(r_i)$

▪ 19.5 [19-03 example.py] – Convex Hull

>> LINE 24) [vx,vy,x,y] = cv2.fitLine(cnt, cv2.DIST_L2, 0, 0.01, 0.01)

>> LINE 25) ly = int((-x*vy/vx)+y)

>> LINE 26) ry = int(((cols-x)*vy/vx)+y)

>> LINE 28) cv2.line(img, (cols-1, ry), (0, ly), (0,0,255),2)

- cv2.fitLine() 함수는 주어진 Contour에 최적화된 직선을 추출한다. cv2.fitLine() 함수의 2번째 인자는 최적화된 직선을 계산하기 위해 M-estimator라 불리는 알고리즘이 사용할 계산식을 지정하는 역할을 한다.
- 이에 대해 좀 더 자세하게 살펴보면, 구하려는 직선과 Contour의 i번째 점들간의 거리를 ri, p(r)를 거리 함수라고 할 때, p(ri)의 합이 최소가 되는 직선을 구한다.
- cv2.fitLine() 함수는 계산된 직선위의 점 (x, y)와 직선의 단위벡터 (vx, vy)를 리턴한다. (vx, vy)와 (x, y)를 이용해 직선을 이미지에 맞게 연장한 후 빨간색으로 그려준다.

```python
import numpy as np
import cv2


def convex():
    img = cv2.imread('images/convexhull.png')

    img1 = img.copy()
    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)

    cnt = contours[1]
    cv2.drawContours(img, [cnt], 0, (0,255,0), 3)

    check = cv2.isContourConvex(cnt)

    if not check:
        hull = cv2.convexHull(cnt)
        cv2.drawContours(img1, [hull], 0, (0,255,0), 3)
        cv2.imshow('convexhull', img1)

    cv2.imshow('contour', img)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

convex()
```

$\rho(r)$ $\sum_i \rho(r_i)$

```python
import numpy as np
import cv2

def convex():
    img = cv2.imread('images/lightning.png')
    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)

    cnt = contours[5]

    x,y,w,h = cv2.boundingRect(cnt)
    cv2.rectangle(img, (x,y), (x+w, y+h), (0, 0, 255), 3)

    rect = cv2.minAreaRect(cnt)
    box = cv2.boxPoints(rect)
    box = np.int0(box)

    cv2.drawContours(img, [box], 0, (0,255,0), 3)

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

convex()
```

```python
import numpy as np
import cv2

def convex():
    img = cv2.imread('images/lightning.png')
    imgray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    rows, cols = img.shape[:2]

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

    cnt = contours[5]

    (x,y),r = cv2.minEnclosingCircle(cnt)
    center = (int(x), int(y))
    r = int(r)

    cv2.circle(img, center, r, (255,0,0),3)

    ellipse = cv2.fitEllipse(cnt)
    cv2.ellipse(img,ellipse,(0,255,0),3)

    [vx,vy,x,y] = cv2.fitLine(cnt, cv2.DIST_L2, 0, 0.01, 0.01)
    ly = int((-x*vy/vx)+y)
    ry = int(((cols-x)*vy/vx)+y)

    cv2.line(img, (cols-1, ry), (0, ly), (0,0,255),2)

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

convex()
```