

4각형 혹은 원으로 객체 둘러싸기

`boundingRect()`, `minEnclosingCircle()`,
`minAreaRect()` ← `boxPoints()`,

2024년 1학기

서경대학교 김진헌

Structural Analysis and Shape Descriptors

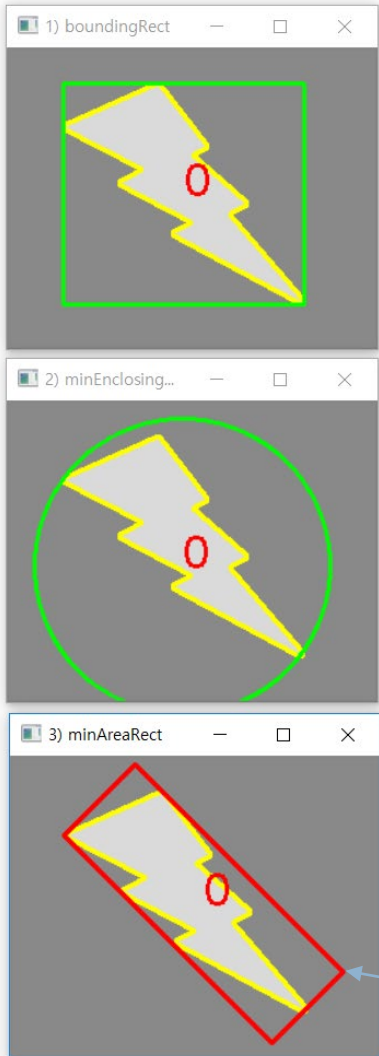
차례

1

- 1. 둘러싸는 함수와 반환 값 요약
- 2. BoundingRect()
- 3. minEnclosingCircle()
- 4. minAreaRect()
 - ▣ boxPoints()
 - ▣ 실험사례

1. 함수와 반환 값 요약

2

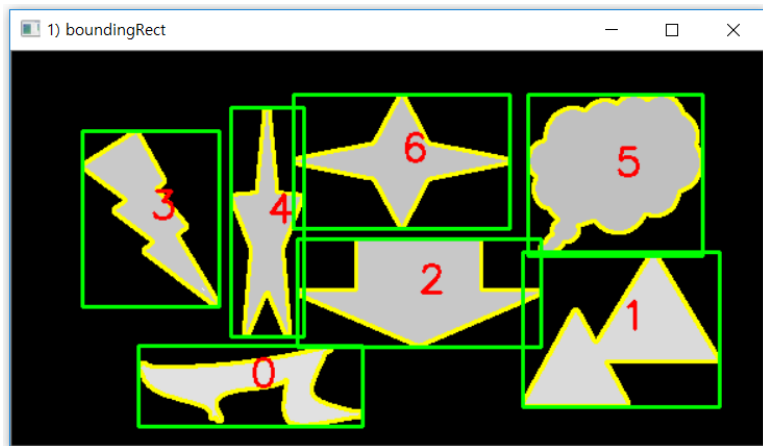


- $x, y, w, h = cv2.boundingRect(c)$
 - ▣ 윤곽선 c 를 둘러싼 4각형의 좌측 상단 좌표(x, y)와 폭(w), 높이(h)를 반환한다.
 - ▣ 그림으로 표현할 때는 좌측상단, 우측하단의 점을 지정하여;
 - $Rectangle(canvas, (x, y), (x+w, y+h), color, thickness)$
 - $(x, y), radius = cv2.minEnclosingCircle(c)$
 - ▣ 윤곽선 c 를 둘러싸는 원의 중심 $center=(x, y)$ 과 반지름을 반환한다.
 - ▣ 그림으로 표현할 때는;
 - $circle(canvas, center, radius, (0,255,0),2)$
 - $rect = cv2.minAreaRect(c)$
 - ▣ **rotated rectangle** 이라는 회전 4각형으로 근사화한다.
 - ▣ 반환 값 = 중심, 크기, 각도 혹은 4각형 3개 꼭지점 좌표
 - ▣ 그림으로 표현할 때는;
 - $boxPoints()$ 함수로 4개의 점으로 변환, 이는 윤곽어와 같은 배열. => $drawContours()$ 함수로 그린다.
- 검토 사항: 타원으로 $fitEllipse()$ 한 결과와 비교해 보자.!!

2. BoundingRect()

3

Rects Circle.py



1) boundingRect()

객체별로 4개의 점을 반환하는데 x, y, width, height로 해석한다.

0: ret= (97, 226, 172, 62)

1: ret= (392, 154, 151, 119)

2: ret= (219, 144, 187, 83)

3: ret= (54, 61, 105, 135)

4: ret= (168, 43, 56, 176)

5: ret= (396, 33, 134, 124)

6: ret= (216, 33, 166, 103)

1) boundingRect()로 근사화 => rectangle() 함수로 그린다.

```
print('\n1) boundingRect()')
```

```
img2 = base.copy()
```

```
i = 0
```

```
for c in contours:
```

```
# 윤곽선 c를 둘러싼 4각형의 좌측 상단 좌표(x, y)와 폭(w), 높이(h)를 반환한다.
```

```
ret = cv2.boundingRect(c)
```

```
print('ret=', ret)
```

```
x, y, w, h = ret
```

```
cv2.rectangle(img2, (x,y), (x+w, y+h), (0, 255, 0), 2)
```

```
cv2.putText(img2, str(i), (x+int(w/2), y+int(h/2)),
```

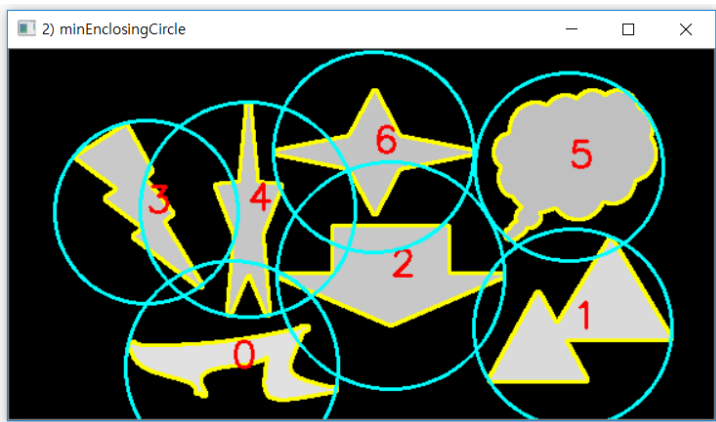
```
cv.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
```

```
i += 1
```

3. minEnclosingCircle()

4

Rects Circle.py



2) minEnclosingCircle()

2개의 원소를 가진 튜플 반환: ((중심점 x, 중심점 y), 반지름)

0: ret= ((182.5, 260.5), 87.47866821289062)

1: ret= ((461.26507568359375, 228.6989288330078), 81.16065216064453)

2: ret= ((312.0, 185.0), 93.00547790527344)

3: ret= ((112.9386978149414, 133.9056854248047), 75.32548522949219)

4: ret= ((195.5, 131.2771453857422), 88.27865600585938)

5: ret= ((458.0, 96.0), 77.98727416992188)

6: ret= ((298.5, 84.0), 82.50615692138672)

```
i = 0
for c in contours:
    # 윤곽선 c를 둘러싸는 원의 중심(x,y)과 반지름을 반환한다.
    ret = cv2.minEnclosingCircle(c)
    print('ret=', ret)
    (x, y), radius = ret
    center = (int(x), int(y))
    radius = int(radius)
    cv2.circle(img2, center, radius, (255, 255, 0), 2)
    cv2.putText(img2, str(i), center, cv.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    i += 1
```

4. minAreaRect()

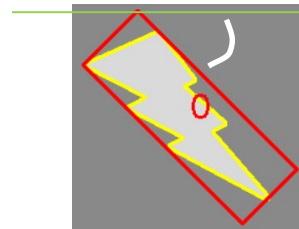
5

§ minAreaRect()

RotatedRect cv::minAreaRect (**InputArray** points)

Python:

```
retval = cv.minAreaRect( points )
```



- Finds a *rotated rectangle* of the minimum area enclosing the input 2D point set.
 - ▣ The function calculates and returns the minimum-area bounding rectangle (possibly rotated) for a specified point set.
- Input Parameters
 - ▣ points: Input vector of 2D points, stored in `std::vector<>` or Mat
 - ▣ 입력 파라미터로 contour를 활용할 수 있음.
- Return Value
 - ▣ 반환 값=무게중심(x, y), 크기($width, height$), 각도 혹은 4각형의 3개 꼭지점 좌표
 - ▣ 각도: returns the rotation angle. When the angle is 0, 90, 180, 270 etc., the rectangle becomes an up-right rectangle.
 - ▣ boxPoints() 함수에 전달하여 바로 선 사각형 정보를 반환받는다.

boxPoints()

6

§ boxPoints()

```
void cv::boxPoints ( RotatedRect box,  
                    OutputArray points  
                    )
```

Python:

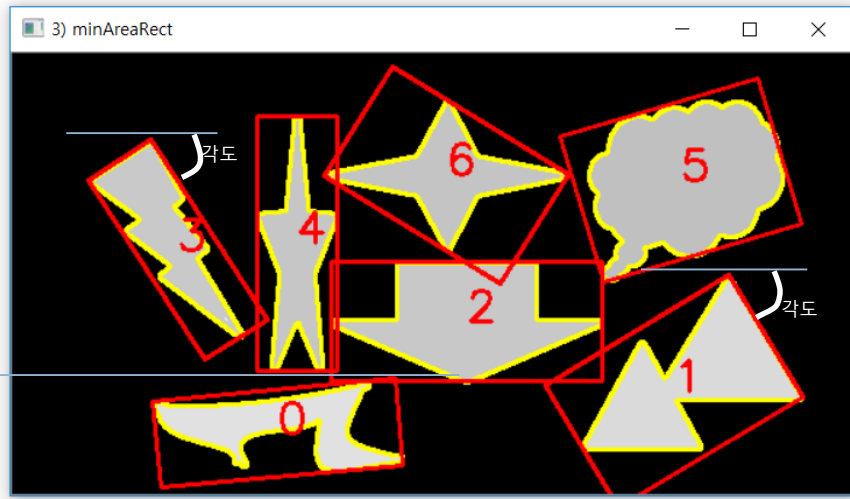
```
points = cv.boxPoints( box[, points] )
```

- rotated rect의 4점을 찾는다. rotated rectangle를 그릴 때 유용하다. .
 - ▣ The function finds the four vertices of a rotated rectangle. This function is useful to draw the rectangle.
- Input Parameters
 - ▣ minAreaRect()로 전달받은 **rotated rectangle** 정보
- Return Value: points
 - ▣ **4각형의 꼭지점 4개의 좌표**(x, y)를 ndarray 타입으로 반환한다.
 - ▣ Type(points)= <class ' numpy.ndarray ' >. Points.shape= (4, 2)
 - ▣ drawContours(image, [points], -1, ...) 이런 식으로 그림을 그릴 수 있다.

실험사례

7

Rects Circle.py



```
for i, c in enumerate(contours):
    ret = cv2.minAreaRect(c)
    pprint_rr(i, ret)    # print rotated rectangle
    (x, y), size, angle = ret
    # '4각형의 4개의 꼭지점 좌표(x,y)를 ndarray 타입으로 반환한다.
    box = cv2.boxPoints(ret) # box.shape= (4, 2). 부동소수
    box1 = np.intp(box)    # box의 각 요소 값을 정수로 바꾼다.
    print('box1=', box1)
    cv2.drawContours(img2, contours=[box1], contourIdx: 0,
                     color: (0, 0, 255), thickness: 2)
    cv2.putText(img2, str(i), org: (int(x), int(y)), cv.FONT_HERSHEY_SIMPLEX,
                 fontScale: 1, color: (0, 0, 255), thickness: 2)
```

3) minAreaRect()

무게 중심(x, y), 크기(w, h), 각도 -> 이른바 rotated rectangle 정보를 반환한다.

```
def pprint_rr(ret):    # print rotated rectangle
    (x, y), size, angle = ret
    print(f'center(x,y)=({int(x):d}, {int(y):d})')
    print(f'size(w,h)=({int(w):d}, {int(h):d})')
    print(f'angle={angle:.1f}')

```

Shape 0:

center(x,y)=(182, 261)

size(w,h)=(166, 103)

angle=84.7

box1= [[96 239]

[263 223]

[268 283]

[102 298]]

Shape 1:

center(x,y)=(454, 232)

size(w,h)=(166, 103)

angle=59.0

box1= [[366 228]

[492 152]

[543 236]

[416 312]]

Shape 3:

center(x,y)=(113, 135)

size(w,h)=(166, 103)

angle=57.3

box1= [[52 87]

[95 59]

[175 183]

[132 210]]