

# Fitting Ellipse

`ellipse()`: 타원을 그리는 함수

`fitEllipse()`: 점들을 타원으로 근사화하는 함수

2024년 1학기

서경대학교 김진헌

Structural Analysis and Shape Descriptors

# 차례

- 1. axes, angle 지정 타원 그리기
  - ▣ 실습 1, 2-회전, 일부만 그리기
  - ▣ 실습 3-같은 도형의 다른 표현
  - ▣ 맞춰보세요.
- 2. rotated rectangle로 타원 그리기
- 3. 타원형으로의 피팅
  - ▣ 사례1: 원본과 같은 값
  - ▣ 사례2: 다른 값이지만 사실상 같은 타원

타원 그리기

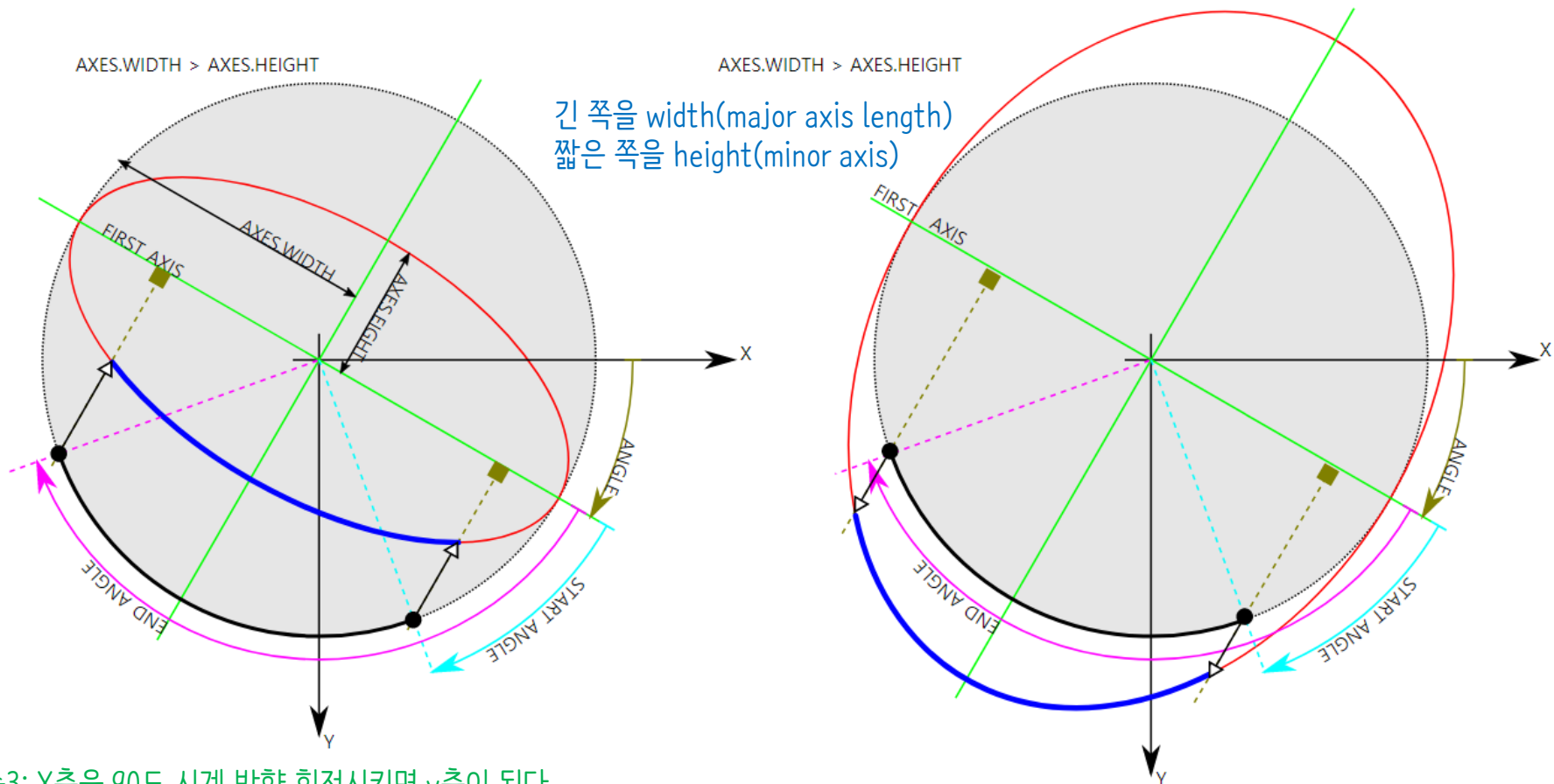
타원 둘레를 따라 약간의  
자음 추가하여 점들을 분포  
하게 만들고, 이를 기반으로  
로 타원 그리기

# 1. axes, angle 지정 타원 그리기

[https://docs.opencv.org/master/d6/d6e/group\\_imgproc\\_draw.html#ga28b2267d35786f5f890ca167236cbc69](https://docs.opencv.org/master/d6/d6e/group_imgproc_draw.html#ga28b2267d35786f5f890ca167236cbc69)

1\_ellipse(axes, angle).py

- `img = cv.ellipse(img, center, axes, angle, startAngle, endAngle, color[, thickness[, lineType[, shift]]])`
- Draws a simple or thick elliptic arc or fills an ellipse sector.



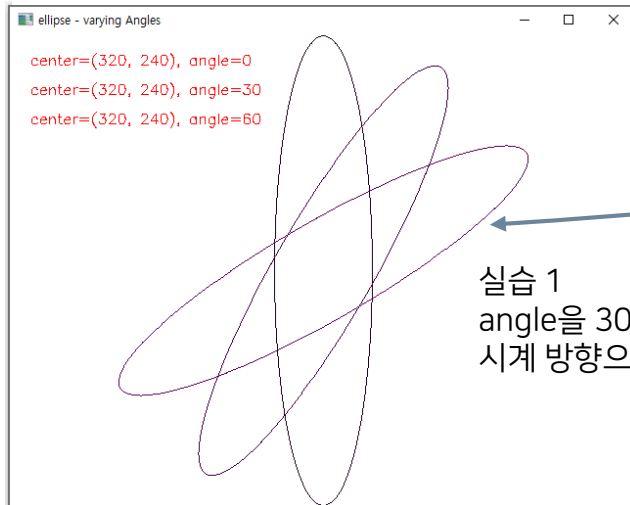
실습3: X축을 90도 시계 방향 회전시키면 y축이 된다.  
angle을 90증가시키고, axes를 바꾸면 모양이 유지된다.

Parameters of Elliptic Arc

# 실습 1, 2 - 회전, 일부만 그리기

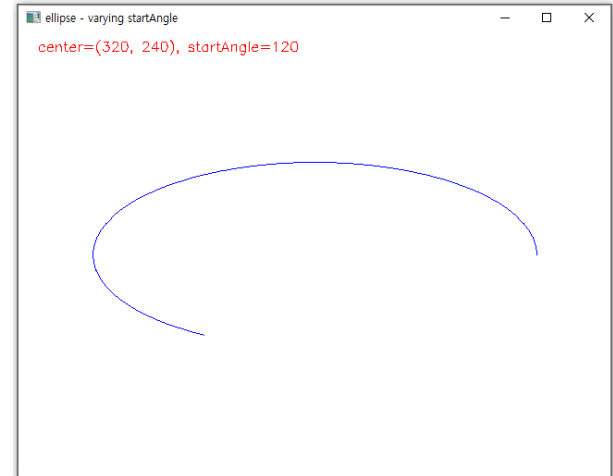
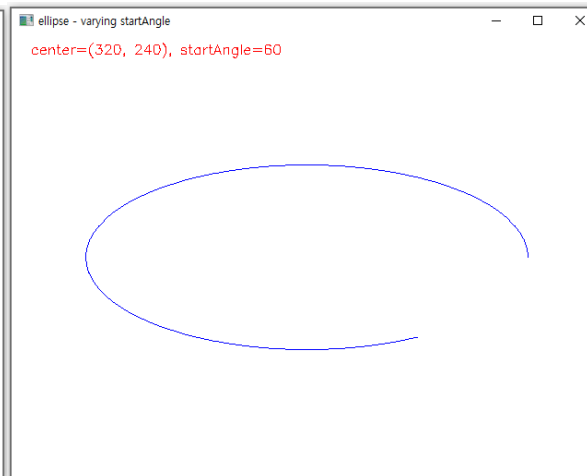
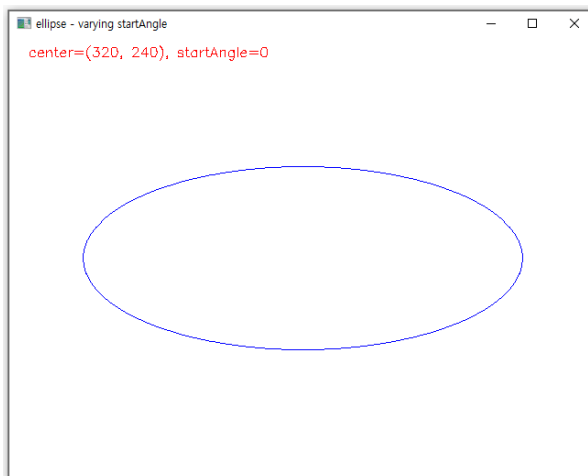
1\_ellipse(axes, angle).py

```
dst = cv.ellipse(img, center, axes, angle, startAngle, endAngle, color)
```



실습 1  
angle을 30씩 증가시켜 보았을 때...  
시계 방향으로 회전한다.

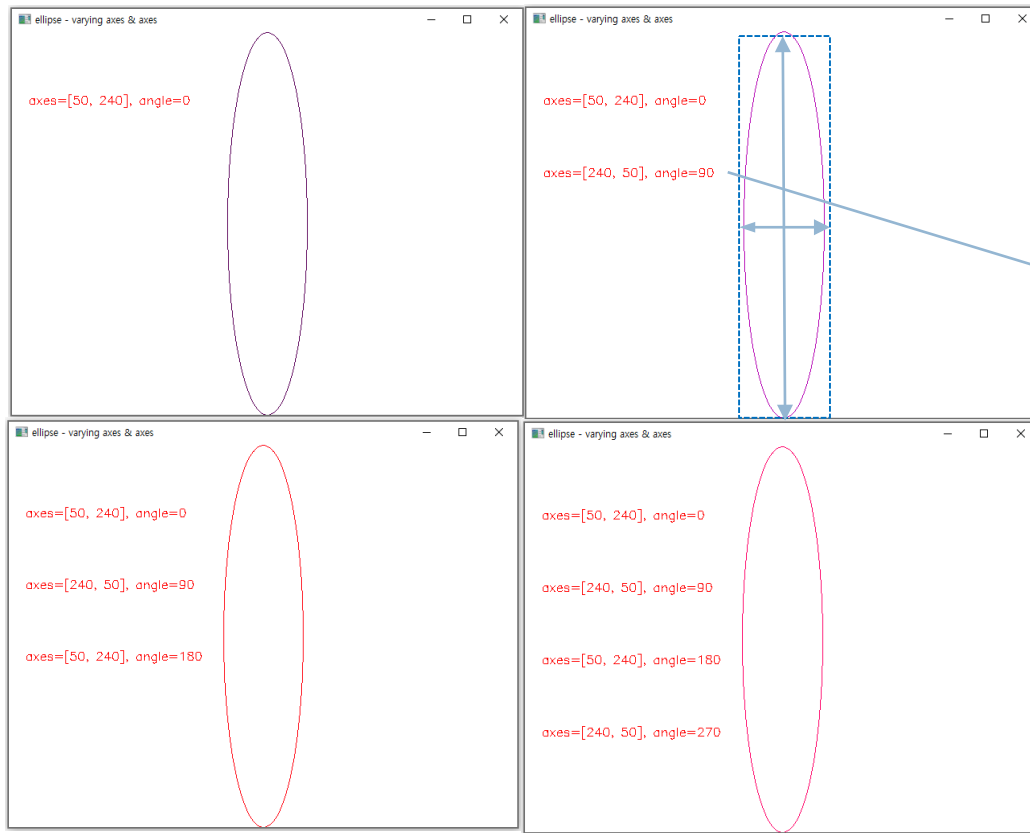
실습 2  
startAngle을 60도씩 증가시켜 보았을 때..  
타원의 시작 지점이 시계방향으로 늦어진다.  
종료 각도는 360도로 고정



# 실습 3 - 같은 도형의 다른 표현

1\_ellipse(axes, angle).py

```
dst = cv.ellipse(img, center, axes, angle, startAngle, endAngle, color)
```



angle를 90도 증가시킬 때마다 axes의 원소값의 위치를 서로 토글시킨다. → 이렇게 하면 모양이 바뀌지 않는다.

폭 420, 높이 50의 박스를 90도 회전

axes는 main 축의 길이 tuple로서 (제일 긴 축, 제일 짧은 축) 혹은 그 반대로 (제일 짧은 축, 제일 긴 축)을 의미한다.  
회전은 angle 담당한다.

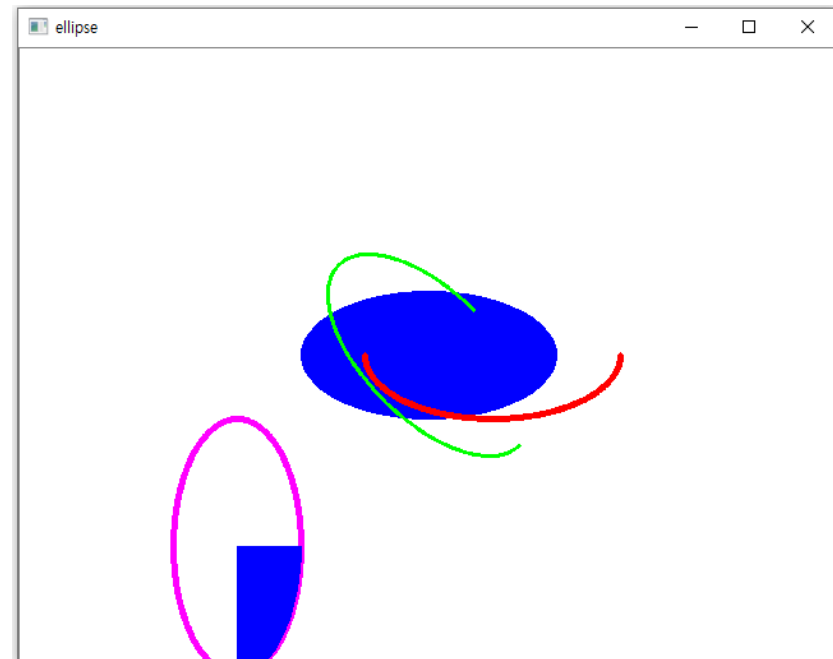
실습3: X축을 90도 시계 방향 회전시키면 y축이 된다.

angle을 90증가시키고, axes의 원소들의 위치를 토글시키면 모양이 유지된다.

# 맞춰보세요.

1\_ellipse(axes, angle).py

```
w=640; h=480; # 가로, 세로
img = np.full((h, w, 3), (255, 255, 255), np.uint8)
cv.ellipse(img, (int(w/2),int(h/2)), (100,50), 0, 0,360, (255, 0,0), -1) # 0=기울기. 0,360=0~360. -1=채우기
cv.ellipse(img, (int(w/2)+50,int(h/2)), (100,50), 0, 0,180, (0, 0,255), 3) # 0=기울기. 0,180=0~180. 3=선두께
cv.ellipse(img, (int(w/2),int(h/2)), (100,50), 45, 0,270, (0, 255, 0), 2) # 45=기울기. 0,270=0~270. 2=선두께
cv.ellipse(img, (int(w/2)-150,int(h/2)+150), (50, 100), 0, 0,360, (255, 0, 255), 3) # 0=기울기. 0,360=0~360. 3=선두께
cv.ellipse(img, (int(w/2)-150,int(h/2)+150), (50, 100), 0, 0,90, (255, 0, 0), -1) # 0=기울기. 0,90=0~90. -1=채우기
```

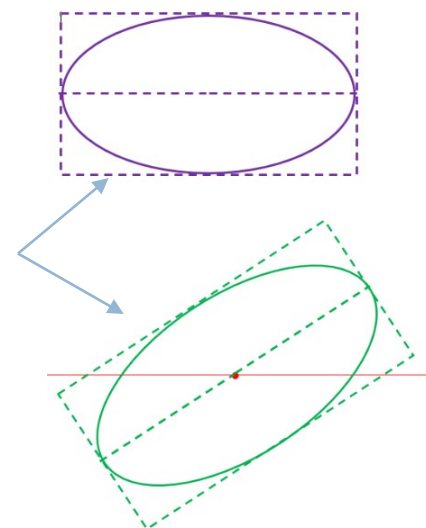


## 2. rotated rectangle로 타원 그리기

1\_ellipse(box).py

- rotated rectangle 형의 box 데이터(RotatedRect)는 원래 기울어진 4각형으로 어떤 객체를 둘러쌀 때 사용하는 데이터 형이다. ellipse() 함수는 이 박스안에 들어갈 수 있는 타원형을 그린다.
  - ▣ `img = cv.ellipse(img, box, color[, thickness[, lineType]]`
- box 데이터는 3개의 원소로 이루어진 튜플(중심점, 가로x세로, 각도)이다. 이 데이터는 부동소수를 지원한다.
  - ▣ `center = (x, y)` # 타원의 중심
  - ▣ `size = (width, height)` # 타원의 크기. 반지름 성격이 아님. 전체 길이. 가로x세로.
  - ▣ `angle = box(타원)가 x좌표에 대해 얼마나(degree) 기울었는가?`
- 사례: 아래 box 정보를 ellipse() 함수에 제공한다
  - ▣ `box = (center, size, angle)`
  - ▣ `box = ((x, y), (width, height), angle)`

2개의 타원을 한 화면에  
그려보세요.



# 3. 타원형으로의 피팅

2\_fitEllipse.py

- `return_value = fitEllipse()` 함수로 반환받은 3개의
- 데이터를 기반으로 타원 혹은 선 모델을 피팅하여 해당 모델과의 유사성을 검토한다.
  - ▣ 타원과 직선을 점으로 모델링하고 여기에 잡음을 첨가한다.
  - ▣ `fitEllipse()` 함수는 잡음이 첨가된 점들을 이용하여 이를 타원으로 피팅하여 타원을 재현하기 위한 값(`ret_val`)을 반환받는다. 이는 2가지 중의 하나의 형태로 활용한다.
    - `ret_box = fitEllipse(영상)`                      # rotated rectangle 정보 반환
    - `center, axis, angle = fitEllipse(영상)`   # rotated rectangle을 unpacking
  - ▣ 모델링한 결과를 2가지 방법으로 그림을 그린다.
    - `img = cv.ellipse(img, ret_box, color[, thickness[, lineType]]`
    - `img = cv.ellipse(img, center, axes, angle, startAngle, endAngle, color[, thickness[, lineType[, shift]]]`
  - ▣ 원래의 형상 파라미터와 얼마나 유사한지 검토한다.



# 사례1: 원본과 같은 값

2\_fitEllipse.py

fitEllipse() 함수는 axes 정보를 ellipse() 함수의 ases 정보에 비해 2배(즉 지름과 같은 개념)으로 반환한다.  
rotated rectangle 형의 box 데이터를 반환받은 것을 ellipse() 함수로 그릴 때는 그대로 입력하여 써도 된다.  
그러나, axes정보로 바꾸어 ellipse() 함수에 넣어 그림을 그릴 때는 1/2로 줄여야 한다.  
**ellipse() = fitEllipse(영상) # 3개의 원소로 이루어진 tuple 정보 반환**

```
ellipse= ((255.90187072753906, 256.15869140625), (259.75714111328125, 498.7635192871094), 121.27838897705078)  
type(ellipse)=<class 'tuple'>, len(ellipse)=3
```

unpacked values: center=(255, 256), axes=(259, 498), angle=121

\*\*\* original: center=(256, 256), axes=(128, 248), angle=122

\*\*\* fitting: center=(255, 256), axes=(129, 249), angle=121.28

2배의 길이로 반환

각도가 같을 때는 axes 정보도 같다.



## 사례2: 다른 값이지만 사실상 같은 타원

2\_fitEllipse.py

반환받은 값이 원본과 다르지만, 실제로 그림을 그려보면 같은 도형이다.

Fitting한 파라미터의 angle에 90도를 더하고, axes 튜플 원소를 자리를 바꾸어 1/2배하면 같은 original과 같은 결과가 나온다.

```
ellipse= ((255.41900634765625, 256.25531005859375), (141.4468536376953, 234.73533630371094), 40.5925407409668)
type(ellipse)=<class 'tuple'>, len(ellipse)=3
unpacked values: center=(255, 256), axes=(141, 234), angle=40
*** original: center=(256, 256), axes=(116, 66), angle=130
*** fitting: center=(255, 256), axes=(70, 117), angle=40.59
```

2배의 길이로 반환

