

OpenCV-Python

02. Object Detection



2021-W
MECHA Seminar

Contents :

- 이미지 경계점 검출
- 이미지 경계 및 사각형 그리기
- HSV Palette
- 과제: example2



이미지 경계점 검출 :

drawcontours.py

- cv2.findcontours(image, mode, method)
- mode : contour를 찾는 방법
 - RETR_EXTERNAL : 바깥쪽 line만 찾기
 - RETR_LIST : 모든 line을 찾지만, hierarchy를 구성하지 않음
 - RETR_TREE : 모든 line을 찾고 hierarchy를 구성
- Method: contour를 찾는 근사치 방법
 - CHAIN_APPROX_SIMPLE: contour line을 그릴 수 있는 point만 저장
- Usage
 - contours, hierarchy = cv2.findContours(redmask_HSV, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)



이미지 경계 및 사각형 그리기 :

drawcontours.py

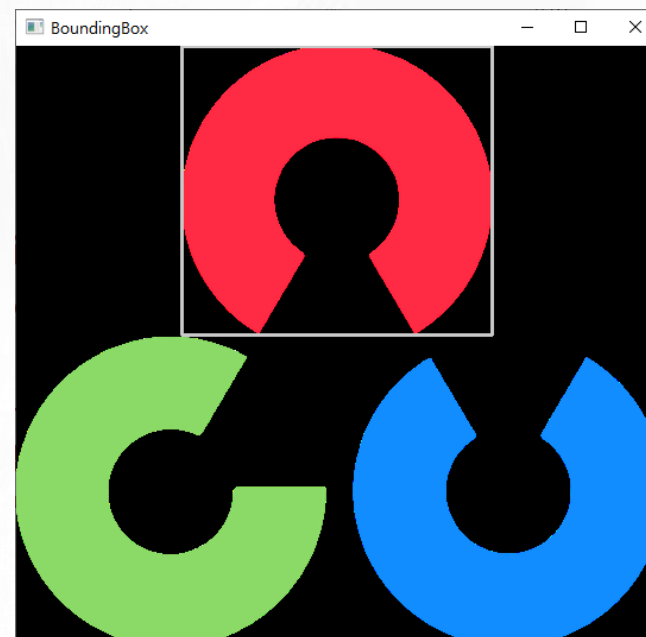
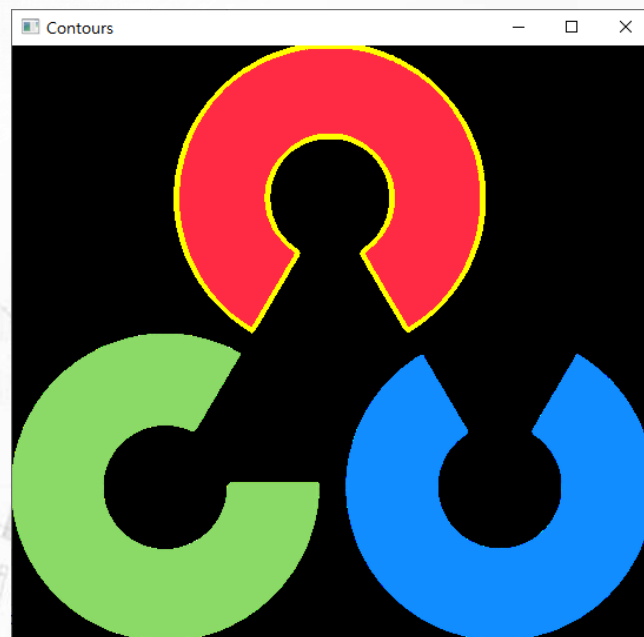
- `cv2.drawcontours(image, contours, contour_index, color, thickness)`
- `contour_index`
 - 각 점들에 대해 index 부여됨. 전부 다 그릴 때에는 `index=-1`
- `x,y,w,h = cv2.boundingrect(contours[index])`
 - 폐곡선으로 둘러싸인 contour에 대해 직사각형을 생성
 - `x,y`: 직사각형 좌측 상단 좌표, `w,h`: 직사각형 가로 세로 길이
- `cv2.rectangle(image, 좌측 상단 좌표, 우측 하단 좌표, color, thickness)`
- Usage
 - `cv2.rectangle(image_draw, (x,y), (x+w,y+h), (200,200,200), 2)`



이미지 경계 및 사각형 그리기 :

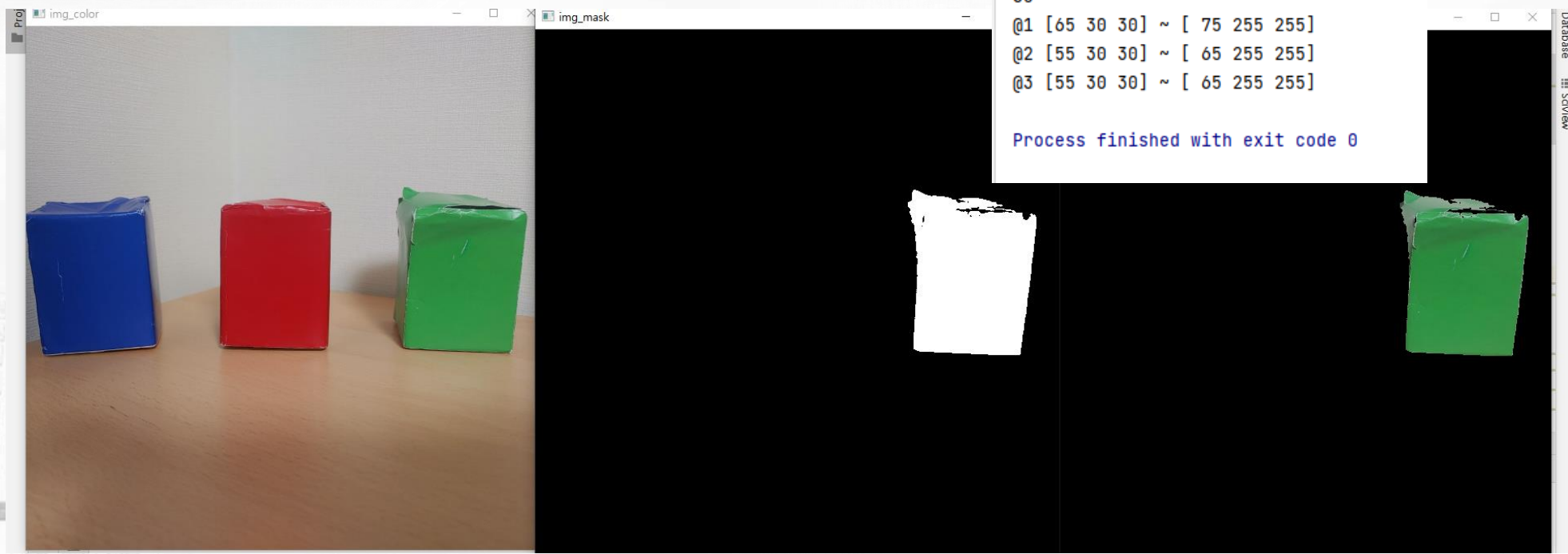
Contour로 둘러싸인 부분의 넓이 및 중심점

- `cv2.contourArea(contours[0])`
- 일반적으로 중심점은 `boundingrect`의 중심점을 찾는다
 - $(x+w/2, y+h/2)$



Noise가 있는 이미지의 색상을 찾을 때...

- img_color 창에 임의의 지점 클릭 시 콘솔 창에 그 지점의 HSV와 근처의 H값들을 선택 적용하여 @1,2,3의 합집합으로 masking작업 진행. Masking 결과와 범위 값에 해당하는 영상 부분 반환.



과제2 :

rgbobject.png에서 각 물체 색상 별 masking결과, contour, boundingrect를 그리고 해당 영역의 넓이와 중심점을 구하시오.

- 제목을 example2.py로 새 python file 생성
- Hint:
- HSV 변환하기 전 noise를 제거하기 위해 GaussianBlur 진행

```
import cv2
image = cv2.imread('rgbobject.png')
frame = cv2.GaussianBlur(image, (15,15), 0)
image_HSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

- hsvpicker.py를 반복 이용하여 각 물체 색상의 HSV range를 찾음
- drawcontours.py와 비슷하게 진행
- 출력물(masking이미지x3, contour이미지x3, boundingrect이미지x3, 각 영역 및 중심점)

