

빅데이터 분석이지만 "OpenCV" 강의

강사: 김정호

kimsmap@outlook.com

8.1 마커 탐지

<https://april.eecs.umich.edu/software/apriltag>

8.1.1 AprilTag

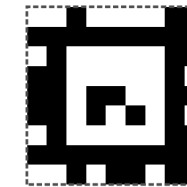
- 숫자를 담고 있는 마커(marker)의 여러가지 종류 중 하나
- 여러가지 타입(family로 부름)이 있으며 각각 특성이 있음
- Tag36h11이 가장 대표적이며 가장 많이 쓰임
- 미리 만들어 놓은 태그 이미지는

[https://github.com/AprilRobotics/apriltag-
imgs/tree/master/tag36h11](https://github.com/AprilRobotics/apriltag-
imgs/tree/master/tag36h11) 에서 다운로드 가능함

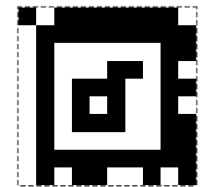
- 설치를 하기 위해서는 pip에서 작업
→ `pip install pupil-apriltags`
- 설치가 완료되면 파이썬 코드에서 반입(import) 가능



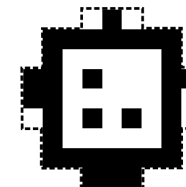
Tag36h11



TagStandard41h12



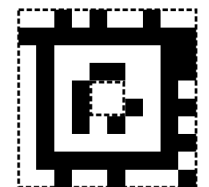
TagStandard52h13



TagCircle21h7



TagCircle49h12



TagCustom48h12



8.1 마커 탐지

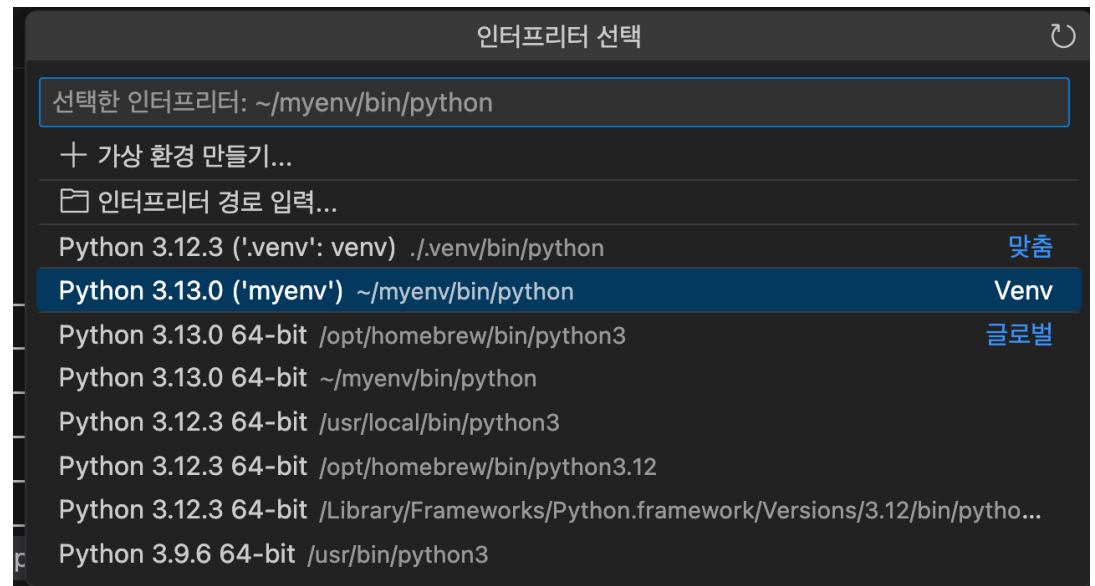
<https://april.eecs.umich.edu/software/apriltag>

8.1.1 AprilTag

- 만약 설치가 되지 않는다면? (ex. MacOS)

: 가상환경에서 설치한 뒤 사용하도록 함

- 1) 가상환경 생성
 - `python3 -m venv ~/myenv`
- 2) 가상 환경 활성화
 - `source ~/myenv/bin/activate`
- 3) 패키지 설치
 - `pip install pupil-apriltags`
- 4) 인터프리터 설정(`~/myenv/bin/python`)
- 5) (다 끝나면) 가상 환경 비활성화
 - `deactivate`



8.1 마커 탐지

8.1.2 예제

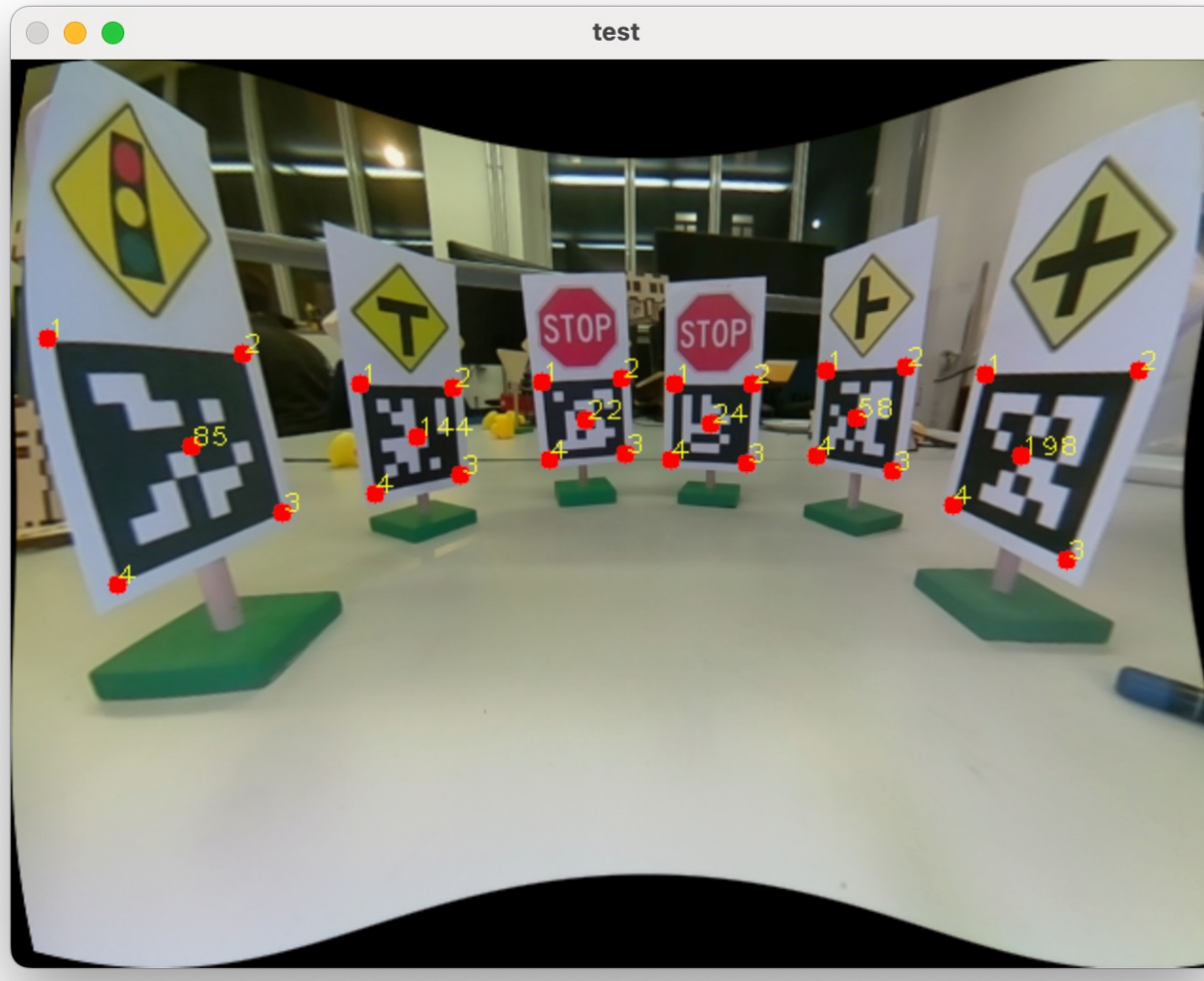
<https://april.eecs.umich.edu/software/apriltag>

```
1  import cv2
2  import pupil_apriltags as apriltag
3
4  # 사진 불러오기
5  img = cv2.imread('aprilImage.png')
6  img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
7
8  # 탐지를 위한 detector 생성
9  # detector = apriltag.Detector()
10 detector = apriltag.Detector(families='tag36h11')
11
12 # 영상에서 aprilTag 찾기
13 results = detector.detect(img_gray)
14
15 # 찾은 aprilTag를 하나하나 확인 및 표시하기
16 for result in results:
17     print(result.tag_id, ': ', result.center[0], ', ', result.center[1])
18     cv2.circle(img, (int(result.center[0]), int(result.center[1])), 5, (0, 0, 255), -1)
19     cv2.putText(img, str(result.tag_id), (int(result.center[0]), int(result.center[1])), 1, 1, (0, 255, 255))
20
21     # Tag의 모서리를 표시하기 위한 for loop
22     idx = 0
23     for corner in result.corners:
24         idx = idx + 1
25         cv2.circle(img, (int(corner[0]), int(corner[1])), 5, (0, 0, 255), -1)
26         cv2.putText(img, str(idx), (int(corner[0]), int(corner[1])), 1, 1, (0, 255, 255))
27
28 # 최종적으로 표시된 결과 출력하기
29 cv2.imshow('test', img)
30 cv2.waitKey(0)
```


8.1 마커 탐지

8.1.2 예제

- 탐지 결과는 다음 그림과 같이 표시됩니다.



8.1 마커 탐지

도전과제

- 다음의 코드를 참고합니다.



```
1 # 관련 라이브러리 선언
2 import numpy as np
3 import cv2
4 from matplotlib import pyplot as plt
5
6 # 영상 읽기
7 img1 = cv2.imread("../images/img12.jpg", cv2.IMREAD_GRAYSCALE)
8 img1 = cv2.resize(img1, (320, 240))
9
10 # 투시 변환 수행
11 h, w = img1.shape
12 point1_src = np.float32([[1, 1], [w-10, 10], [5, h-5], [w-4, h-4]])
13 point1_dst = np.float32([[15, 15], [w-60, 15], [10, h-25], [w-100, h-50]])
14 point2_src = np.float32([[148, 145], [168, 144], [136, 223], [188, 222]])
15 point2_dst = np.float32([[136, 145], [188, 144], [136, 223], [188, 222]])
16 per_mat1 = cv2.getPerspectiveTransform(point1_src, point1_dst)
17 per_mat2 = cv2.getPerspectiveTransform(point2_src, point2_dst)
18 res1 = cv2.warpPerspective(img1, per_mat1, (w, h))
19 res2 = cv2.warpPerspective(img1, per_mat2, (w, h))
20
21 # 결과 영상 출력
22 res = []
23 res.append(img1), res.append(res1), res.append(res2)
24
25 for i in range(3):
26     plt.subplot(2, 2, i+1)
27     plt.imshow(res[i], cmap='gray')
28     plt.xticks([], plt.yticks([]))
29
30 plt.show()
```