

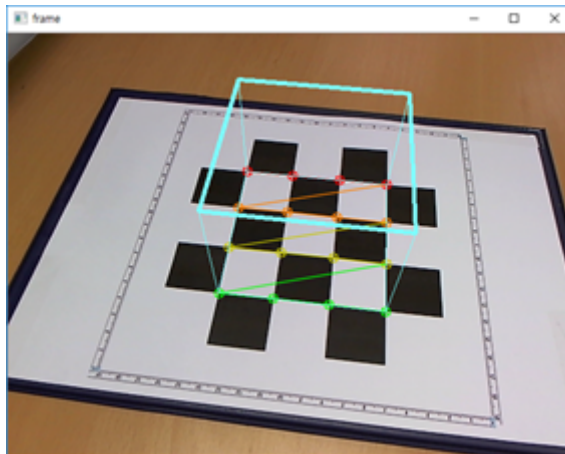
# Assign#1. AR Cube 구현하기

## 과제 개요

- 제출기한: 10월 15일 (일요일) 24:00
- 제출물:
  - 구현코드 github에 올리기 (각자 개인 github)
  - Report 제출 (문제, 구현내용(주요코드), 결과(캡처), 결론, github링크 포함)

## [Assignment #1] AR Cube

- 과제 개요: 라이브 카메라 영상에서 마커(체커보드)를 인식하고, 인식된 마커 위에 가상의 AR 큐브를 그려주는 프로그램 구현. 카메라의 움직임에 관계없이 가상의 AR 큐브가 자연스럽게 보이도록 해야 함



- 프로그래밍 환경: 컴퓨터비전 라이브러리인 OpenCV를 이용하되, 프로그래밍 언어는 c++, python 등 본인이 편한 언어를 사용
- 마커(체커보드): 체커보드를 출력하여 마커로 사용 (격자개수는 3x4, 4x5 등 무관)
- 추천 과정
  - related theoretical basics
    - camera parameters: focal length, principal point
    - coordinate system: world coordinate, camera coordinate, normalized image coordinate, image coordinate (pixel)
    - the process of projection from world coordinate to image coordinate
  - perform camera calibration and get the intrinsic parameters of your camera
    - camera calibration basics: <http://darkpgmr.tistory.com/32>
    - calibration toolbox: <https://darkpgmr.tistory.com/139>
  - install opencv library and build a simple program that displays live cam video
    - 참고용 template code: <https://github.com/oricljy/mathvision>
      - feature matching 코드 참조
      - windows용 opencv binary도 다운로드 받을 수 있음

- write a program that recognizes a checkerboard marker and detects and displays corner points.
  - `cv::findChessboardCorners()`
  - `cv::cornerSubPix()`
  - `cv::drawChessboardCorners()`
- estimate camera pose  $[R|t]$  (position and orientation) from the detected corner points
  - `cv::solvePnP()`
  - $R$ : 3 x 3 rotation matrix,  $t$ : 3 x 1 translation vector
  - $P_{\text{camera}} = R * P_{\text{world}} + t$  ( $P_{\text{camera}}$ : camera coord.,  $P_{\text{world}}$ : world coord.)
- write the final program
  - define world coordinate system relative to the marker
  - set world coordinate for 8 cubic corner points
  - compute the projected image coordinate of the cubic corner points by using the estimated  $R$  and  $t$ .
  - draw AR: `cv::line()`
- write the report
- 참고 자료 링크
  - opencv: <http://opencv.org/>
  - opencv online reference: <http://docs.opencv.org/3.2.0/>
  - imaging coordinate systems
    - <http://darkpgmr.tistory.com/77>
  - camera pose estimation
    - <http://darkpgmr.tistory.com/99>
    - <http://darkpgmr.tistory.com/122>