Due date : 2021.06.03

|  |
| --- |
|  |
| **Visual Computing - Team Project 2** |

****

|  |  |  |
| --- | --- | --- |
| **과목명** | 비주얼컴퓨팅 | |
| **담당교수** | 김동준 | |
| **팀번호** | 4 | |
| **팀구성원** | 2017204034 | 김두연 (팀장) |
| 2018204017 | 남지수 |
| ~~2019204012~~ | ~~전우진~~ **(불참)** |
| ~~2019204040~~ | ~~정재완~~ **(불참)** |

**Camera Calibration** pattern\_size = (9,6) / square\_size = 22(mm) / 25 images

**화살이(가) 표시된 사진

자동 생성된 설명**

Image1 ) cv.drawChessboardCorners(img, pattern\_size, corners, found)

를 통해 해당 이미지에 패턴을 넣어준 25개의 이미지

**텍스트이(가) 표시된 사진

자동 생성된 설명**

Image2 ) 패턴을 보여주기 위해 image1의 10번째 (index=9) 이미지 일부 확대

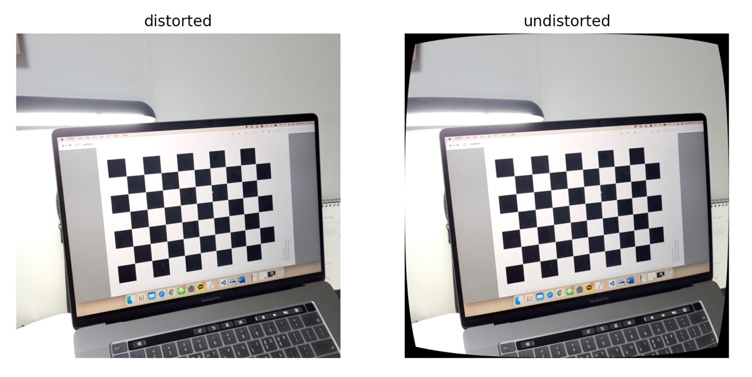
****

Image3 ) 10번째 (index=9) image의 distorted, undistorted 버전

newcameramtx, roi = cv.getOptimalNewCameraMatrix(intrisic\_mtx, dist\_coefs, (w,h), 1, (w,h)) 인자에 1을 넣어서 undistorted 영상에 까만 부분이 나옴

텍스트이(가) 표시된 사진

자동 생성된 설명

Image4) 위 : original camera 의 intrinsic matrix(= K matrix)

아래 : virtual camera 의 intrinsic matrix(= K matrix)

calibration결과가 이와 같이 나옴

**Camera pose estimation**

텍스트, 모니터, 실내이(가) 표시된 사진

자동 생성된 설명

Image1 ) 7번째 (index=6) undistorted image의 X(red), Y(Green), Z(Blue) 표시

axis = np.float32([[square\_size, 0, 0], [0, square\_size, 0], [0, 0, square\_size]]).reshape(-1, 3) 로

3차원의 축을 axis에 저장하고

cv.line(img\_undist\_test, axis\_center, tuple(imgpts[0].ravel().astype(int)), (0,0,255), 5)

         cv.line(img\_undist\_test, axis\_center, tuple(imgpts[1].ravel().astype(int)), (0,255,0), 5)

         cv.line(img\_undist\_test, axis\_center, tuple(imgpts[2].ravel().astype(int)), (255,0,0), 5) 을 통해

10번째 (index=9) undistorted image에 해당 축을 X(red), Y(Green), Z(Blue)로 표시

텍스트이(가) 표시된 사진

자동 생성된 설명

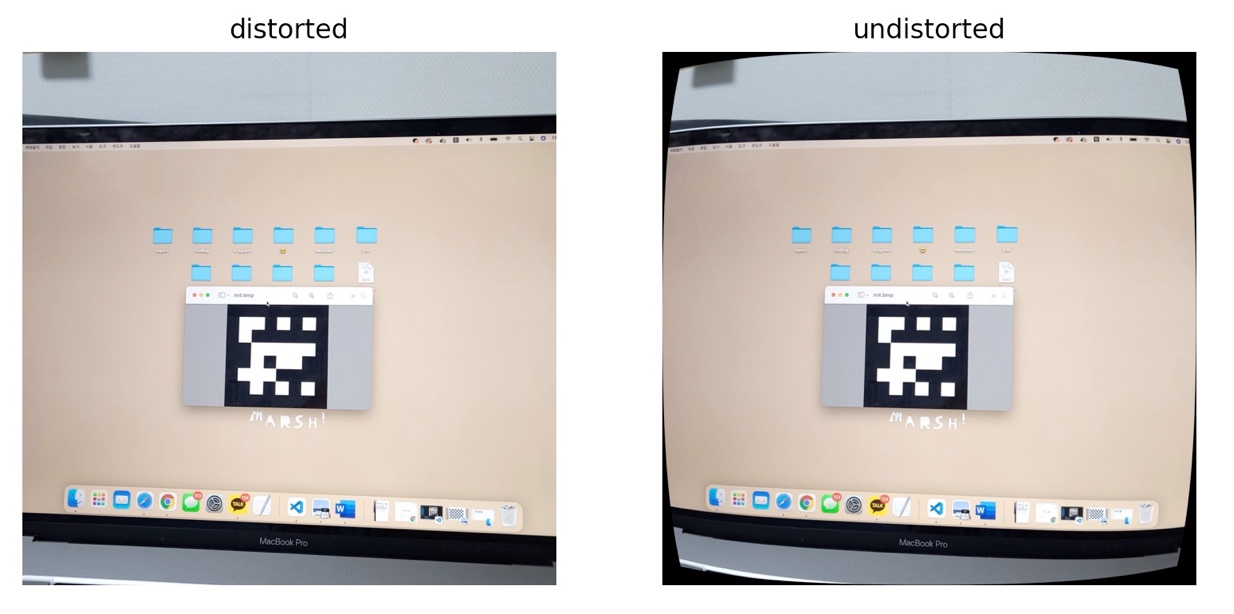
Image2) 앞서 저장한 calibration 결과를 불러와서 위에 같은 calibration 결과값이 나타나고

ret, rvecs, tvecs = cv.solvePnP(pattern\_points, corners2, newcameramtx, None) 을 undistorted 영상에 대해 처리했기 때문에 newcameramtx(virtual camera에 대한 intrinsic parameter)를 인자로 줘서 extrinsic parameter에 해당하는 rotation을 나타내는 3차원 vector인 rvecs, translation을 나타내는 3차원 vector인 tvecs를 출력해줌

**Camera calibration and Camera pose estimation by Aruco Marker**

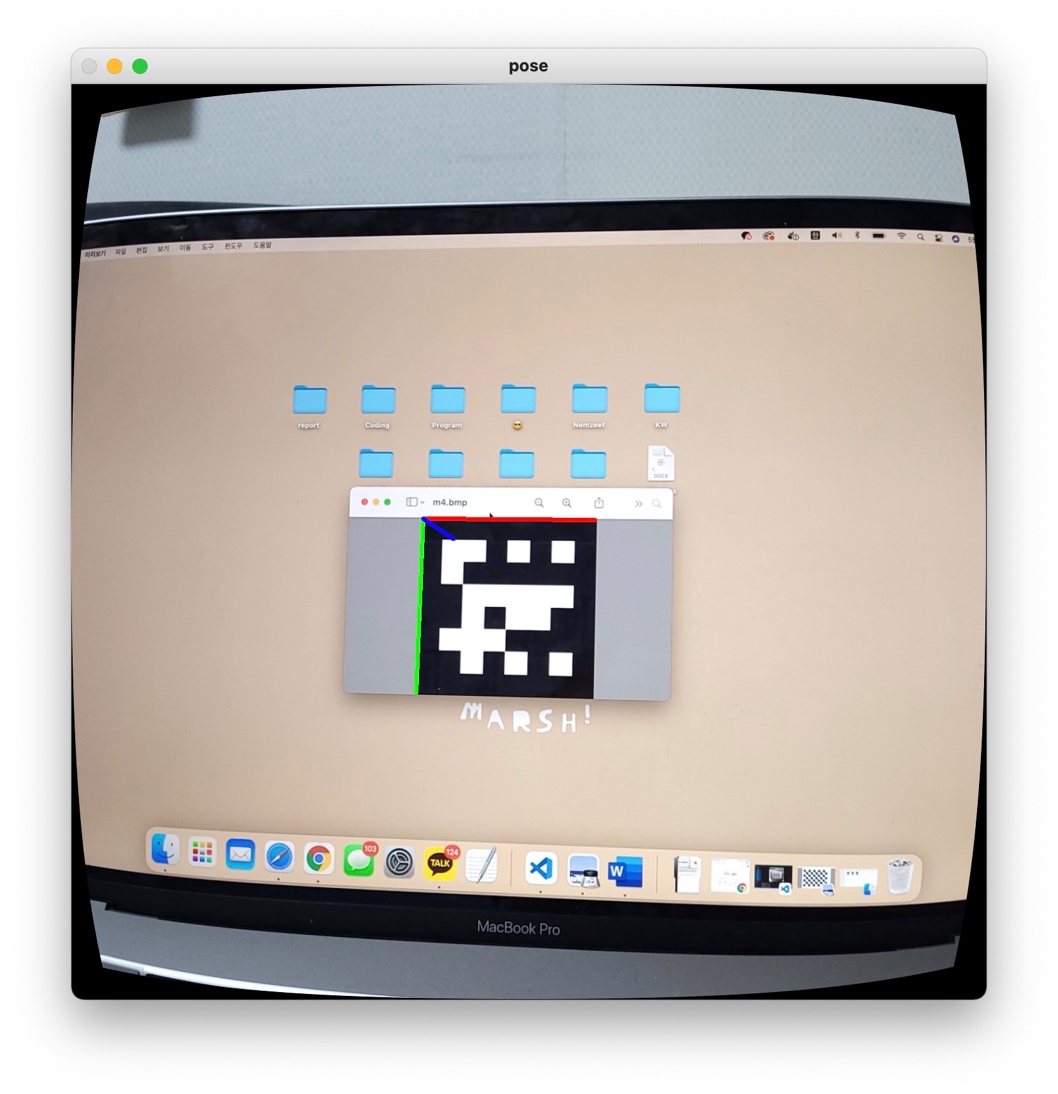
* 1번 ID aruco marker : cv.aruco.drawMarker(arucoDict, 4, 300, marker\_img, 1)

pattern\_size = (2,2) / square\_size = 57(mm)



Camera calibration )

좌측 : distorted한 1번 aruco / 우측 : camera calibration을 이용해 undistorted 해준 1번 aruco

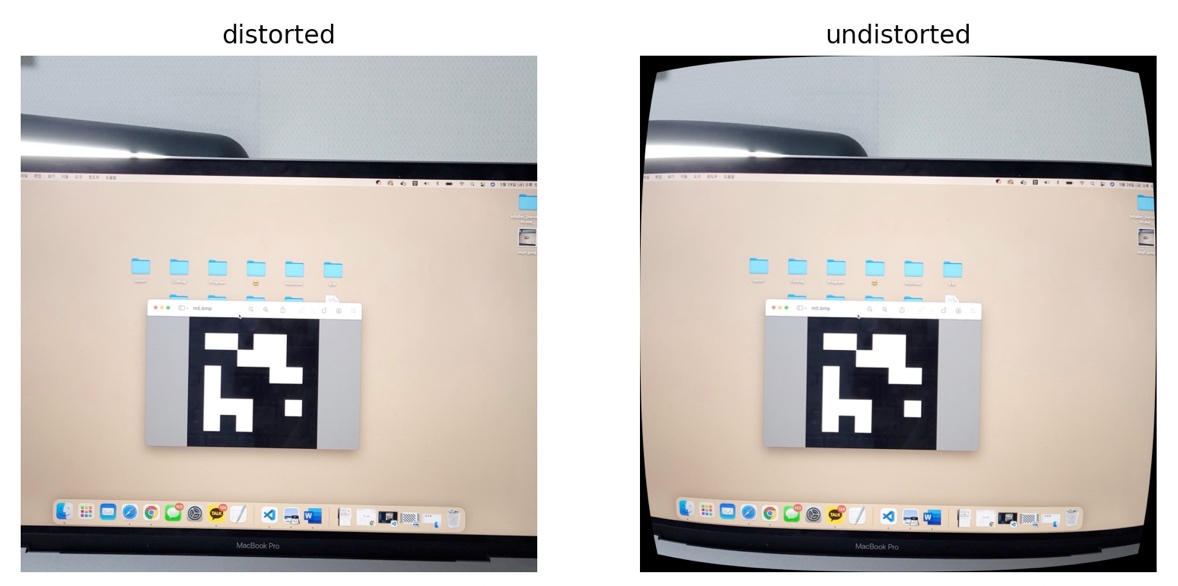


Camera pose estimation )

undistorted 1번 aruco위에 x(red), y(green), z(blue) 3개의 축을 표시

* 2번 ID aruco marker : cv.aruco.drawMarker(arucoDict, 5, 300, marker\_img, 1)

pattern\_size = (2,2) / square\_size = 79.5(mm)



Camera calibration )

좌측 : distorted한 2번 aruco / 우측 : camera calibration을 이용해 undistorted 해준 2번 aruco

텍스트, 모니터, 실내, 디스플레이이(가) 표시된 사진

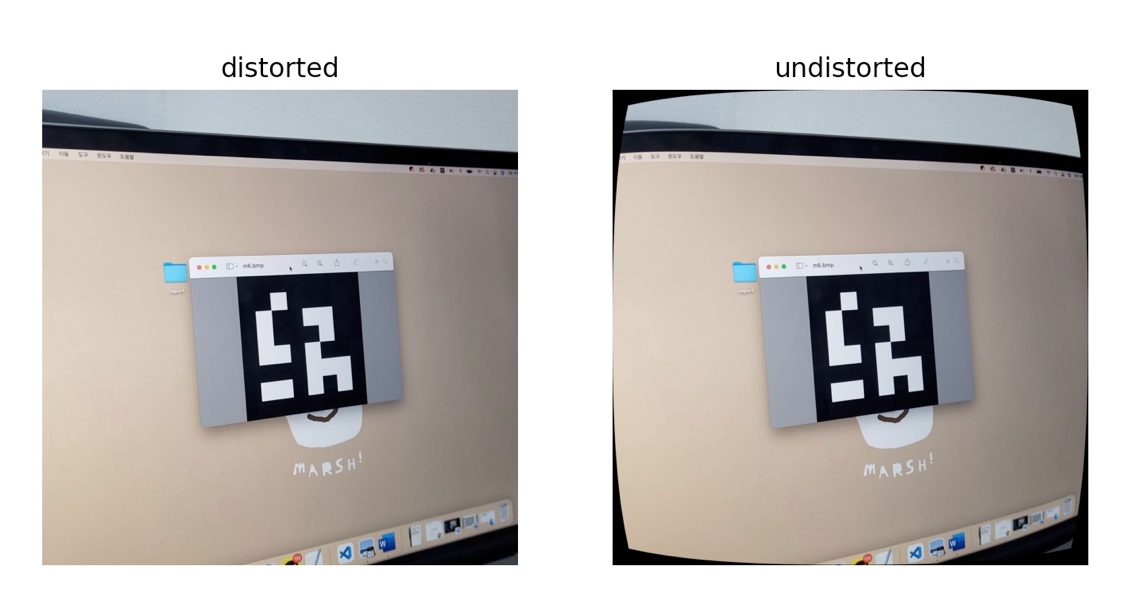
자동 생성된 설명

Camera pose estimation )

undistorted 2번 aruco위에 x(red), y(green), z(blue) 3개의 축을 표시

* 3번 ID aruco marker : cv.aruco.drawMarker(arucoDict, 6, 300, marker\_img, 1)

pattern\_size = (2,2) / square\_size = 71(mm)



Camera calibration )

좌측 : distorted한 3번 aruco / 우측 : camera calibration을 이용해 undistorted 해준 3번 aruco

텍스트, 모니터, 실내이(가) 표시된 사진

자동 생성된 설명

Camera pose estimation )

undistorted 3번 aruco위에 x(red), y(green), z(blue) 3개의 축을 표시