2021.09.14 조수익

|  |
| --- |
| 1번 : 코드 |
| # 캘리브레이션 영상을 가지고 정보 구하여 dump로 저장하기  import cv2  import numpy as np  import glob  import pickle  x = 9  y = 6  objp = np.zeros((y \* x, 3), np.float32)  objp[:, :2] = np.mgrid[0:x, 0:y].T.reshape(-1, 2)  objpoints = []  imgpoints = []  # 사용자가 제시한 조건에 맞는 파일명을 리스트 형식으로 반환  images = glob.glob('cal\_img/img\*.jpg')  total\_images = len(images)  for idx, fname in enumerate(images):  img = cv2.imread(fname)  gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  ret, corners = cv2.findChessboardCorners(gray, (x, y), None)  if ret == True:  objpoints.append(objp)  imgpoints.append(corners)  cv2.drawChessboardCorners(img, (x, y), corners, ret)  write\_name = 'output/corners\_found' + str(idx) + '.jpg'  cv2.imwrite(write\_name, img)  out\_str = f'{idx}/{total\_images}'  cv2.putText(img, out\_str, (10, 25), cv2.FONT\_HERSHEY\_SIMPLEX, 0.4, (0, 255, 255), 1)  cv2.imshow('img', img)  cv2.waitKey(500)  cv2.destroyAllWindows()  img = cv2.imread('cal\_img/img8.jpg')  height, width = img.shape[:2]  img\_size = (width, height)  ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, img\_size, None, None)  dst = cv2.undistort(img, mtx, dist, None, mtx)  cv2.imwrite('output/test\_undist.jpg', dst)  dist\_pickle = {}  dist\_pickle["mtx"] = mtx  dist\_pickle["dist"] = dist  pickle.dump(dist\_pickle, open('output/wide\_dist\_pickle.p', 'wb'))  print('mtx', mtx)  print('dist', dist)  img\_result = cv2.hconcat([img, dst])  img\_result = cv2.pyrDown(img\_result)  cv2.imshow('dst', img\_result)  cv2.waitKey(0)  cv2.destroyAllWindows() |

|  |
| --- |
| 2번 캘리브레이션 된 영상 10장(체크보드 에지 찾은영상) |
|  |

|  |
| --- |
| 3번 : 캘리브레이션 파라미터 |
| mtx  [[506.52503006 0. 639.90188285]  [ 0. 504.88241117 500.40168987]  [ 0. 0. 1. ]]  dist  [[-0.05909949 0.17707455 0.00035229 -0.00356956 -0.1514924 ]] |

|  |
| --- |
| 4번 캘리브레이션 원본영상 – 결과영상 |
|  |