

차선 찾기 알고리즘

류정필

차선 찾기(1/3)

참고할 소스코드: <https://github.com/jungpilyu/sdc-advanced-lane-finding>

1. 카메라 보정

- `cv2.findChessboardCorners(image, patternSize[, corners[, flags]])` → `retval, corners`
- `cv2.calibrateCamera(objectPoints, imagePoints, imageSize, cameraMatrix, distCoeffs[, rvecs[, tvecs[, flags[, criteria]]]])` → `retval, cameraMatrix, distCoeffs, rvecs, tvecs`

2. 테스트 이미지 보정

- `cv2.undistort(src, cameraMatrix, distCoeffs[, dst[, newCameraMatrix]])` → `dst`

차선 찾기(2/3)

3. 컬러 공간 변환

- `cv2.cvtColor(src, code[, dst[, dstCn]])` → dst
- HLS 컬러공간 : https://ko.wikipedia.org/wiki/%EC%83%89_%EA%B3%B5%EA%B0%84

4. 소벨 변환

- `cv2.Sobel(src, ddepth, dx, dy[, dst[, ksize[, scale[, delta[, borderType]]]])` → dst

5. 컬러변환 + 소벨 + 노이즈필터링

- `cv2.filter2D(src, ddepth, kernel[, dst[, anchor[, delta[, borderType]]]])` → dst

차선 찾기(3/3)

6. 새(bird) 공간 시점 변환

- `cv2.getPerspectiveTransform(src, dst) → retval`

7. 히스토그램으로 차선 시작점 확인

8. 차선을 이차 곡선으로 근사화

- `numpy.polyfit(x, y, deg, rcond=None, full=False, w=None, cov=False)`

9. 곡률반경, 중심이탈량 계산 후 운전자(driver) 공간시점으로 변환

11. 비디오의 모든 프레임에 지금까지의 계산 적용