Folder : AdvancedLineDetection\_modified (referred by [MehdiSv/AdvancedLineDetection](https://github.com/MehdiSv/AdvancedLineDetection) )

1.function callstack(완전히 정확한지는 의문, logic을 확인하기 위한 용도)

function callstack : main -> generate\_imshow -> process\_image -> polyfitter -> polyfitter\_sliding -> polyfitter -> process\_image -> generate\_imshow -> main -> time\_queue -> pid\_control -> twiddle -> pid\_control -> time\_queue -> main

2.코드 설명

folder : AdvancedLineDetection\_modified

main.py

main 에 두개의 thread가 있음 : t1-generate imshow, t2-time queue

* generate imshow : 비전에 대한 처리(polyfitter), location을 queue에 push
* time queue : queue에 location을 pop 과pid처리, ipc(namedpipe)를 이용하여 ros에 location을 넘겨줌(racecar package)

warper.py

이미지 공간 왜곡(top view 또는 bird eye view 만들기 위함)

polyfitter.py

* 아래 사각형 부분에 라인의 시작점을 인식
* 윈도우 슬라이딩과 polyfit(empty space)을 이용하여 선 추적

왼쪽선을 인식하면 오른쪽선을 도로폭만큼 x축으로 이동하여 그려줌

* 왼쪽 시작점구간에 일정개수 이상의 점이 발견되지 않으면 중간구간을 인식함
* 중간구간에 polyfit을 이용하여 좌회전을 인식

오른쪽선을 인식하면 도로폭만큼 -x축으로 이동하여 왼쪽선을 그려줌

ros폴더 진입하기 : roscd racecar -> cd racecar/scripts

joy\_teleop.py

line 79 ~ 117

* 속도, 각도 초기화, 속도를 ros 시스템에 전송
* while loop : pid를 ipc(namedpipe)를 통해 송신하고 그 값을 ros시스템에 전송
* weight를 만들어 최대속도 조절(커브구간)

3.사용법

folder : AdvancedLineDetection\_modified

terminal1 -> command : python main.py

after 3~4 second

folder : xycar

terminal2 -> command : ./xycar\_start.sh

xycar\_start.sh 는 terminal1 : ctrl+c 로 종료

python은 terminal2 : ctrl+z로 종료 후 terminal2 -> command : ./kill.sh