

로봇공학개론 1분반 기말 프로젝트 보고서

모바일시스템공학과 2학년 손보경 (32212190)

| | |
|--------------------|---|
| I. 서론 | 1 |
| II. 본론 | 1 |
| 1. 알고리즘 및 특징 | 1 |
| 2. 한계점 | 2 |
| III. 결론 | 2 |

I. 서론

중간고사 이후 배웠던 차선 인식 및 ROS를 응용하여, Turtlebot3를 이용한 Gazebo 시뮬레이션을 제작한다.

II. 본론

1. 알고리즘 및 특징

전반적인 프로젝트의 알고리즘은 로봇이 카메라로 입력받은 사진을 이용하여 차선을 인식하고, 차선을 따라 이동하는 것이다. OpenCV 라이브러리의 cv2를 활용하여 차선 인식을 진행하였다.

먼저, 차선을 구하기 위해, 흰 선은 RGB를 사용하고, 노란 선은 HSV를 사용하여 구한 후 이미지를 합쳐주었다. Gray Scale을 사용하지 않은 점이 특징이다. Gray Scale을 사용하면 그 주변의 벽이나 그 밖의 다른 곳 또한 추출되기 때문에 이 점을 해결하기 위하여 흰 선과 노란 선을 따로 추출한 것이다. 그 후, Canny Edge를 이용하여 픽셀값이 급격하게 변하는 지점을 추출하여 차선을 구했다.

관심 영역은 (10, height), (20, height*0.85), (width-20, height*0.85), (width-10, height)로 매우 작은 영역으로 제작하였다. 그 이유는 큰 사이즈 다각형을 만들면,

많은 정보(주변 차선이 아닌 다른 물체, 다음 경로에 있는 차선 등)가 저장되어 오류가 발생하기 때문이다.

이후, 차선 인식을 위해 Haough transform과 Filtering을 하여 원하는 선분만을 추출하였다.

이 추출한 선분을 이용하여 로봇의 회전각을 지정했다. 필터링된 직선을 좌측 선분, 우측 선분으로 나누어 각각 선분들의 기울기 평균을 구해주었다. 만약, 좌측 선분 값이 없다면, 로봇은 우측으로 회전을 해야 하므로 z축 회전각을 우측 선분의 기울기 비율만큼 좌측 회전을 하도록 하였다. 또한, 회전할 때 범위에 벗어나지 않기 위해 x축 속도를 약 0.1만큼 줄여주었다. 마찬가지로, 우측 선분의 값이 없을 경우도 비슷하게 진행된다. 만약 양쪽 모두 기울기가 있다면 x축의 속도를 약 0.1만큼 올려주었다.

2. 한계점

로봇의 움직임을 제어하는 코드 부분에서 정확한 수식을 사용하지 못한 점이 한계점이다. 이러한 값으로 인해 직선 도로임에도 불구하고 지그재그로 움직이는 모습이 보인다. 다음 수정 때는 많은 시도로 얻은 임의의 수식 값이 아닌 PID 제어기 또는 카르만 필터를 활용하여 정확한 값을 얻어보고 싶다.

또한, 이동하는 중간중간 다른 차선이 인식된다는 점이 한계다. 특히 중간중간 유턴하여 차선의 간격이 작은 곳, 주차 공간에 있는 차선이 인식되어 많은 수정을 거쳐야만 했다. 이를 해결하기 위해, 차선을 그냥 그리는 것이 아닌, 대표 선을 추출해서 굵은 알고리즘을 추가하면 좋을 것 같다.

III. 결론

이러한 알고리즘을 사용하여 직접 Gazebo Simulation을 제작해 보았다. 차선을 추출하여 어떻게 사용할지에 대한 고민을 하면서, 많은 사고력을 키울 수 있었던 것 같다. 또한, 중간고사 이전에 배웠던 이론적인 부분을 적용했다면 좀 더 완성도 높은 프로그램을 제작할 수 있을 텐데 하는 아쉬움이 있었다. 하지만, 내 손으로 직접 제작한 로봇이 움직이는 것이 뿌듯함을 느낄 수 있는 부분이었다.