보고서

보고서 및 논문 윤리 서약

1. 나는 보고서 및 논문의 내용을 조작하지 않겠습니다.
2. 나는 다른 사람의 보고서 및 논문의 내용을 내 것처럼 무단으로 복사하지 않겠습니다.
3. 나는 다른 사람의 보고서 및 논문의 내용을 참고하거나 인용할 시 참고 및 인용 형식을 갖추고 출처를 반드시 밝히겠습니다.
4. 나는 보고서 및 논문을 대신하여 작성하도록 청탁하지도 청탁받지도 않겠습니다.

나는 보고서 및 논문 작성 시 위법 행위를 하지 않고, 명지인으로서 또한 공학인으로서 나의 양심과 명예를 지킬 것을 약속합니다.

보고서명 : 캡스톤디자인2 2차보고서

학 과 : 전기공학과
과 목 : 캡스톤디자인2
담당교수 : 이범주
마 감 일 : 2023.11.22
제 출 일 : 2023.11.22

팀 명 : 캡스톤디자인2 7조

팀원1	학번 :	60205095	이름 :	강승원	(서명)
팀원2	학번 :	60195101	이름 :	장원준	(서명)
팀원3	학번 :	60191782	이름 :	임홍균	(서명)
팀원4	학번 :	60205098	이름 :	김민석	(서명)

제안상품명: Shortest path following vehicle

발주자: 명지대학교 전기공학과 이범주 교수
보고기관: 명지대학교 전기공학과
설계팀: 강승원: 명지대학교 소속
경기도 성남시 분당구 장미로 55, 010-3642-7438
장원준: 명지대학교 소속
경기도 서울시 송파구 마천동 344-5, 010-3734-8415
임홍균: 명지대학교 소속
경기도 서울시 강북구 도봉로 18길 48, 010-6456-7011
김민석: 명지대학교 소속
경기도 화성시 동탄대로 469-12, 010-9054-1670

작성일자: 2023년 11월 20일
문서버전: Ver 1

목 차

요약문

캡스톤디자인2 7조는 이범주 교수님과의 미팅을 거쳐 트랙에서 출발 위치와 도착 위치가 결정되면 최단 경로로 출발 위치에서 도착 위치까지 이동하는 차량인 '최단 경로 추종 차량'을 제작하기로 하였다. 현실적 제약조건은 야외의 지형을 고려해서 제품을 제작하기가 어려워 제품이 정해진 트랙에서 이동하고 제품이 장애물을 스캔하여 지도를 생성하는 과정에서 라이다 대신 카메라를 사용하는 것이다. 제품의 요구사항은 차량이 출발 위치에서 도착 위치로 이동하는지, 이동할 때 최단 경로로 이동하는지, 트랙의 길이 변화하였을 때 다시 경로를 생성하는 데 걸리는 시간이 최소인지이다. 이 요구사항들을 제품의 제작 목적을 세우기 위해 구체적으로 해석하면 트랙상의 좌표, 출발 위치와 도착 위치 사이의 좌표들을 지정하는 알고리즘, 지정된 좌표들을 차량이 오차 없이 추종, 알고리즘 소스 코드의 최적화가 필요하다는 것으로 해석할 수 있다. 따라서 앞의 사항들을 만족하기 위해서 제품에 OPENCV 좌표계, A* 알고리즘, Pure Pursuit 알고리즘 이론 등을 적용한다.

Abstract

After a meeting with Professor Bumjoo Lee, Group 7 of Capstone Design 1 decided to produce a 'Shortest path following vehicle', a vehicle that moves from start to arrival in the shortest route when the start and arrival points are determined on the track. The realistic constraint is that it is difficult to manufacture a product considering the outdoor terrain, so the product moves on a set track and the product scans obstacles to create a map, using a camera instead of a lidar. The product's requirements are whether the vehicle moves from the starting point to the arriving point, moves the shortest route, and whether the time taken to generate the path again is the minimum when the track path changes. To set the purpose of production, these requirements can be interpreted to need the coordinates on the track, the algorithm that specifies the coordinates between the starting point and the arrival point, and the vehicle follows the specified coordinates without error, and the optimization of algorithm source code. Therefore, OPENCV coordinate system, A* algorithm, and Pure Pursuit algorithm theory are applied to the product to satisfy the above.

1. 서론

캡스톤디자인2 7조 강승원, 장원준, 임홍균, 김민석 4명은 사회적 약자 중에 가장 문제가 심각해 보이는 시각 장애인에 대한 제품인 '시각 장애인을 위한 웨어러블 기기(Wearable devices for the blind)' 를 제작하려고 했었다.

그림 1: 사용자를 따라다니는 캐리어

이후 이범주 교수님과의 1차 미팅에서 피드백의 결과로 주제를 변경하기로 하였다. 그래서 조원들끼리 프로젝트 주제에 대한 브레인스토밍을 통해서 시각 장애인을 위한 웨어러블 기기 대신 사용자가 원하는 위치를 설정하면 해당 위치로 장애물을 회피하고 사용자와 거리를 유지하며 길을 안내하는 로봇을 제작하기로 하였다. 그러나 2차 미팅에서 교수님께서 해당 주제가 너무 어려운 주제라고 피드백을 주셨다. 이를 고려해서 제작하기로 결정한 제품은 다음과 같았다. 로봇을 차량 형태로 제작하여 차량이 트랙 위에서 장애물을 회피하며 정해진 경로를 추종하는 라인 트레이서를 제작한다. 그러나 3차 미팅 이전 중간 미팅에서 교수님이 4명이라는 인원에 비해 라인 트레이서는 제작하기 쉽다는 것을 지적하셨다. 조원들과 새롭게 토의한 결과, 라인 트레이서 트랙이 아닌 도로형 트랙에서 출발 위치와 도착 위치가 결정되면 최단 경로로 출발 위치에서 도착 위치까지 이동하는 일종의 자율주행 차량으로 주제를 결정했다.

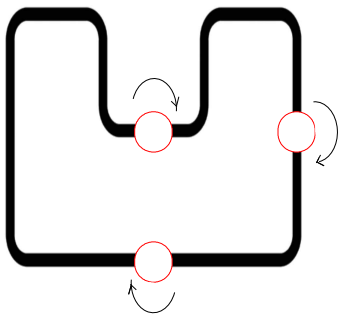


그림 2: 라인 트레이서 트랙

3차 미팅에서 캡스톤디자인2 7조는 앞에서 서술한 제품인 '최단 경로 추종 차량(Shortest path following vehicle)' 을 제작하는 것으로 결정하였다. 그 이후 각 조원 간의 역할 분담을 했는데 그 내용은 다음과 같다. 강승원 조원은 A* 알고리즘 모의실험 및 planning, 임홍균 조원은 라즈베리파이를 이용한 A* 알고리즘 동작과 아두이노 차량으로의 데이터 전송과 라이다 데이터 처리, 장원준 조원은 OPENCV 및 SLAM 담당, 김민석 조원은 차량의 속도 및 위치제어와 경로 추종 담당으로 결정되었다.

본 보고서의 구성은 다음과 같다. 2절에서는 현실적 제한조건인 제품이 야외에서 이동하지 않는 이유와 라이다를 사용하지 않는 이유를 서술한다. 3절에서는 제품에 대한 사용자의 요구사항을 서술하고 이를 제품의 제작 목적을 세우기 위해 구체적으로 해석한 내용을 서술한다. 4절에서는 조원별로 문제 해결에 적용된 이론과 SW를 이용한 설계/해석한 내용을 서술한다. 5절에서는 조원별로 설계에 대한 평가를 진행한 내용을 서술한다.

2. 설계/프로젝트의 현실적 제한조건

최단 경로 추종 차량은 정해진 트랙에서 이동하고 야외에서 이동하지 않으며 최단 경로 추종 차량이 장애물을 스캔하여 지도를 생성하는 과정에서 라이다를 사용하지 않는다. 이에 관해서 서술한다.

최단 경로를 생성하기 위해서는 야외 지도상의 좌표가 있어야 하며 이는 네이버 지도 API, 카카오 지도 API 등을 통해서 좌표 (위도와 경도)를 얻을 수 있다. 하지만 좌표로는

지도상의 지형을 알 수 없다. 지도상의 좌표로는 최단 경로에 산 같은 장애물이 존재할 수 있지만 지형을 고려한다면 산을 돌아가는 경로가 최단 경로일 가능성이 크다.

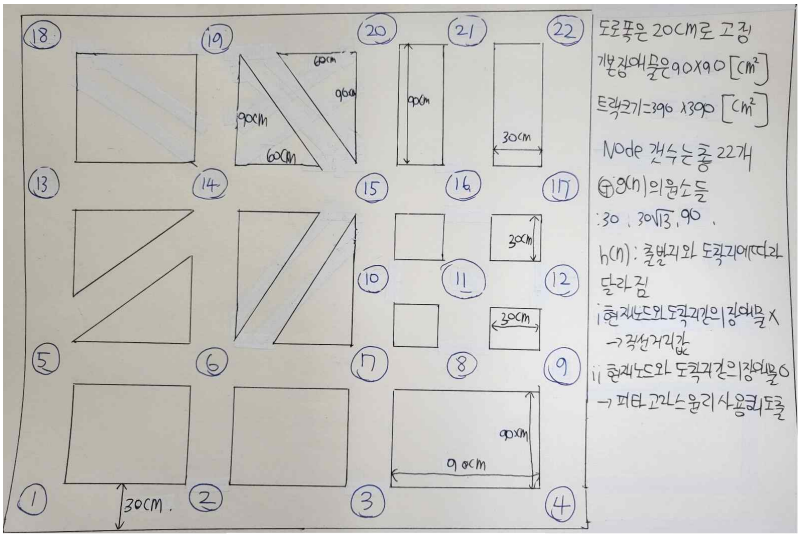


그림 3: 최단 경로 추종 트랙

따라서 캡스톤디자인2 7조는 정해진 트랙에서 이동하는 최단 경로 추종 차량을 제작하기로 하였다. 트랙의 크기는 가변이며, 트랙의 길은 평지에 놓여 있다. 트랙의 예시는 위 그림과 같으며 장애물의 위치는 변경될 수 있다.

또한, 원래는 라이다를 이용해서 SLAM (Simultaneous Localization And Mapping, 동시적 위치추정 및 지도 생성)을 하려 했었다. 라이다가 장애물을 스캔하면서 지도를 생성하는 과정에서 외부 센서의 도움 없이 라이다가 장애물을 스캔한 정보를 이용해 라이다의 현재 위치도 추정하는 것이다. SLAM에 대해서 학습하는 과정에서 SLAM이 ROS 같은 프로그램의 도움이 없이는 구현하기가 매우 힘들다는 것을 알게 되었다.

그래서 라이다를 이용해서 SLAM의 M (Mapping)만을 하려 했었다. SL (Simultaneous Localization)은 외부 센서인 카메라를 이용해서 라이다의 현재 위치를 측정하는 방식으로 해결할 수 있기 때문이다. 하지만 이 카메라를 이용한다면 사실 라이다가 없어도 지도를 생성할 수 있다. 카메라를 이용하면 OPENCV를 사용하여 장애물을 스캔할 수 있기 때문이다.

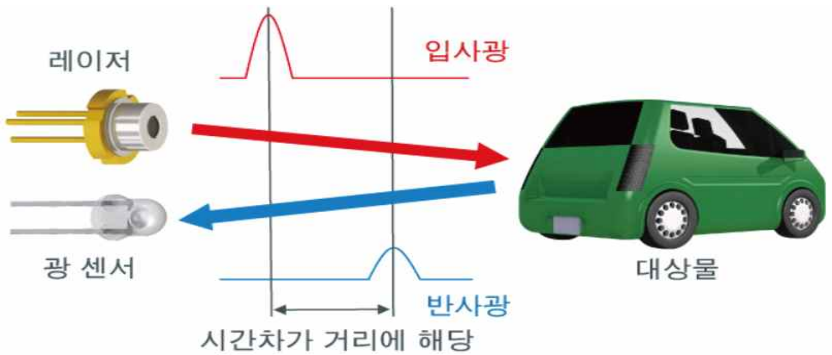


그림 4: 라이다의 원리[1)]

1) 로옴 주식회사 - ROHM Semiconductor, “LiDAR (라이다)”, https://www.rohm.co.kr/electronics-basics/laser-diodes/ld_what10.

장애물을 스캔하여 지도를 생성하는 것은 비교적 라이다보다 카메라가 쉽다. 카메라를 정해진 트랙 위에 설치하고 OPENCV를 사용해서 장애물을 스캔한다면 장애물 좌표를 계산할 수 있다. 라이다는 장애물에 레이저를 조사하고 반사되는 레이저를 이용해서 장애물 좌표를 계산한다. 하지만 라이다는 장애물에 레이저를 조사해야 하므로 공기 중의 먼지 등도 장애물 좌표로 계산될 수 있다. 그리고 라이다의 현재 좌표도 외부 센서를 통해 알려 줘야 한다.

따라서 캡스톤디자인2 7조는 최단 경로 추종 차량이 장애물을 스캔하여 지도를 생성하는 과정에서 라이다 대신 카메라를 사용한다.

3. 설계 목적

현실적 제한조건을 생각하였을 때, 캡스톤디자인2 7조 조원들이 판단한 최단 경로 추종 차량의 요구사항은 다음과 같다.

첫 번째 요구사항은 차량이 출발 위치에서 도착 위치로 이동하는지이다. 이를 만족하지 않는다면 길 안내 차량이 아니므로 이는 반드시 달성해야 할 중요한 요구사항이다. 두 번째 요구사항은 최단 경로 추종 차량이므로 차량이 이동할 때 최단 경로로 이동하는지이다. 세 번째 요구 사항은 트랙의 길이 변화하였을 때 다시 경로를 생성하는 데 걸리는 시간이 최소인지이다. 이 요구사항들을 만족하지 않는다면 차량이 길 안내를 하더라도 도착 위치까지 가는 데 낭비되는 시간이 생긴다.

앞에서 서술한 3가지 요구사항들을 제품의 제작 목적을 세우기 위해 좀 더 구체적으로 해석할 필요성이 있다.

첫 번째 요구사항을 만족하기 위해서는 트랙 상의 좌표가 필요하다. 출발 위치의 좌표, 도착 위치의 좌표, 출발 위치와 도착 위치 사이의 좌표들이 지정된다면 차량이 이 좌표들을 추종하게 하여 출발 위치에서 도착 위치로 이동할 수 있다. 또한, 주어진 좌표들을 정확하게 따라가도록 하는 경로 추종 알고리즘 또한 필요할 것이다.

두 번째 요구사항을 만족하기 위해서는 출발 위치와 도착 위치 사이의 좌표들을 지정하는 알고리즘이 필요하다. 출발 좌표에서 목표 좌표까지 가는 최단 경로를 나타내는 그래프 탐색 알고리즘인 A* 알고리즘을 사용한다. 또한, 이 요구사항들을 만족하기 위해서는 지정된 좌표들을 차량이 오차 없이 추종하게 하는 것도 필요하다.

세 번째 요구 사항을 만족하기 위해서는 A* 알고리즘 소스 코드의 최적화가 필요하다.

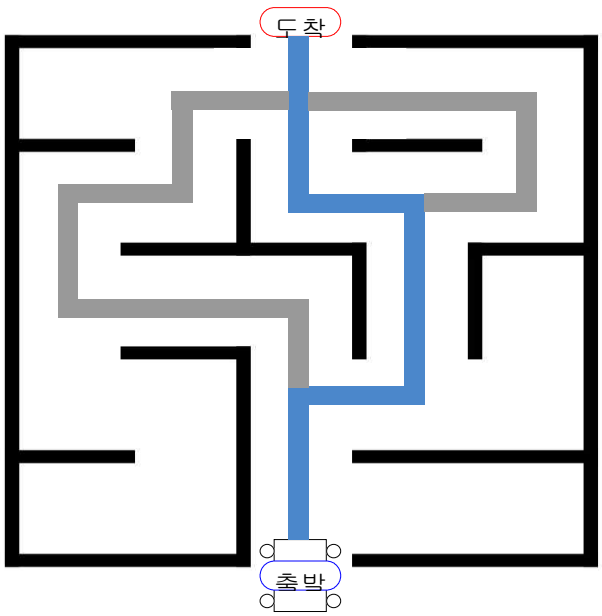


그림 5: 자율주행 차량 동작 그림

그림 5는 앞의 요구사항들과 제한조건들을 적용한 최단 경로 추종 차량의 간단한 동작 그림이다. A* 알고리즘을 기반으로 도착지부터 목적지까지 최적 경로를 탐색하여 이동하는 차량이다. A* 알고리즘 동작에 필요한 좌표와 차량의 현재 위치와 헤딩 각은 직접 카메라로 찍어 OPENCV를 활용하여 동작하며 라즈베리파이에서 해당 좌표 데이터들을 전송받은 뒤 A* 알고리즘을 사용하여 최단 경로를 계산 후, Waypoint의 형태로 데이터를 차량의 아두이노 우노에 전송한다. 전송된 경로 데이터와 차량의 현재 위치와 헤딩 각 데이터를 이용한 경로 추종은 Pure Pursuit 알고리즘을 사용한다.

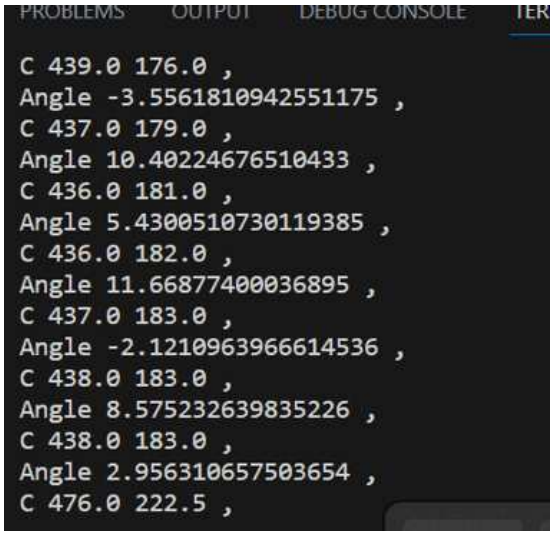
4. 설계 결정에 대한 진행 상황

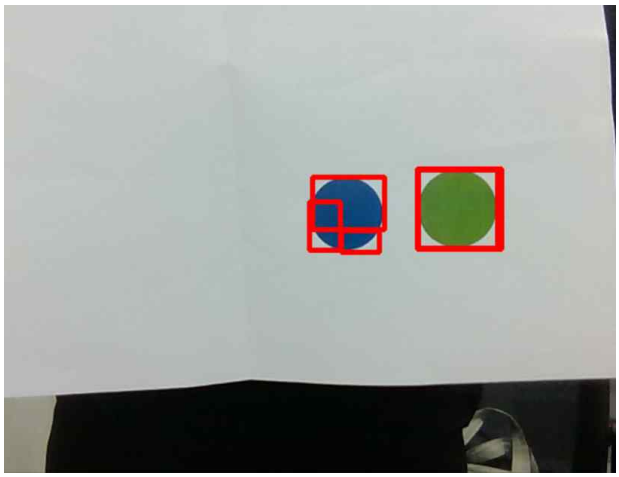
4-1 개인이 맡은 부분에 대한 이론

저는 컴퓨터비전(OPENCV)을 통하여 차량의 출발점과 도착점 지정, 차량의 현재 위치, 차량이 이동함에 따라 변하는 각도, 그리고 장애물을 각각 인식하여 지도에 표시하는 부분을 맡았습니다.

우선 컴퓨터비전을 사용하기 위해선 카메라가 필요하였는데 제가 원하는 위치에 자유롭게 설치할 수 있으며 예산을 줄이기 위해 휴대전화 카메라를 이용하였습니다. 라즈베리파이 카메라나 아두이노 카메라의 경우 선이 연결되며 공간에 대한 제약이 발생하며 카메라 세팅에 대한 자유도나 화질이 낮다는 문제점을 고려하여 휴대전화에서 카메라 IP webcam 앱을 통해 와이파이로 촬영하는 화면을 컴퓨터가 무선으로 카메라가 찍고 있는 화면을 컴퓨터에 공유하는 형태로 카메라를 이용하였습니다.

두 번째로 차량의 위치, 각도, 장애물들을 인식하기 위해 우선 좌표계가 필요한데 PYTHON의 경우 포인터가 없다 보니 numpy를 설치하여 임의의 좌표계를 만들어 사용하였습니다.
numpy는 행렬을 생성하는 라이브러리인데 원래는 통계에 많이 쓰이는 라이브러리이지만 이번 캡스톤 프로젝트에서는 눈에 보이지 않는 행렬을 카메라가 촬영하는 영상 화면 위에 올려 좌표계로 활용하였습니다.
차량의 현재 위치, 각도, 장애물 등의 위치 좌표를 생성하는 역할로 응용하여 numpy를 이용했습니다. 이를 통하여 스캔, 제어, 차량 관련 대부분의 역할을 수행하였습니다.





세 번째는 차량의 위치입니다. 차량의 위치는 numpy와 컴퓨터비전에 있는 HSV 기능을 이용했습니다.

HSV는 RGB 값을 HSV로 변환해 RGB는 원래 빨간색, 초록색, 파란색 3가지 색으로 구성하지만, HSV는 색조, 채도, 명도로 구분하여 기존의 빨간색, 초록색, 파란색으로 구분하는 컴퓨터에 가까운 방법에서 더욱 인간에게 가까운 방법으로 색을 구분하는 방법을 사용하였습니다.

카메라를 통해 확인된 색깔을 RGB로 받아서 HSV로 출력하게 하는 형태로써

차량 위에 파란색 종이를 부착하여 촬영되는 화면에서 어떤 위치에 현재 차량이 존재하는지 카메라가 인식할 때마다 그 위치를 터미널에 알려주는 코드를 짜 지속해서 현재 위치가 어디인지 알려주며 또한 차량의 경우 특정 좌표가 아닌 부피가 존재하기에 스티커 위치에서 값을 더하여 보정하였으며 이를 통해 A* 알고리즘과 Pure Persuit 알고리즘에 위치 데이터를 전송합니다.

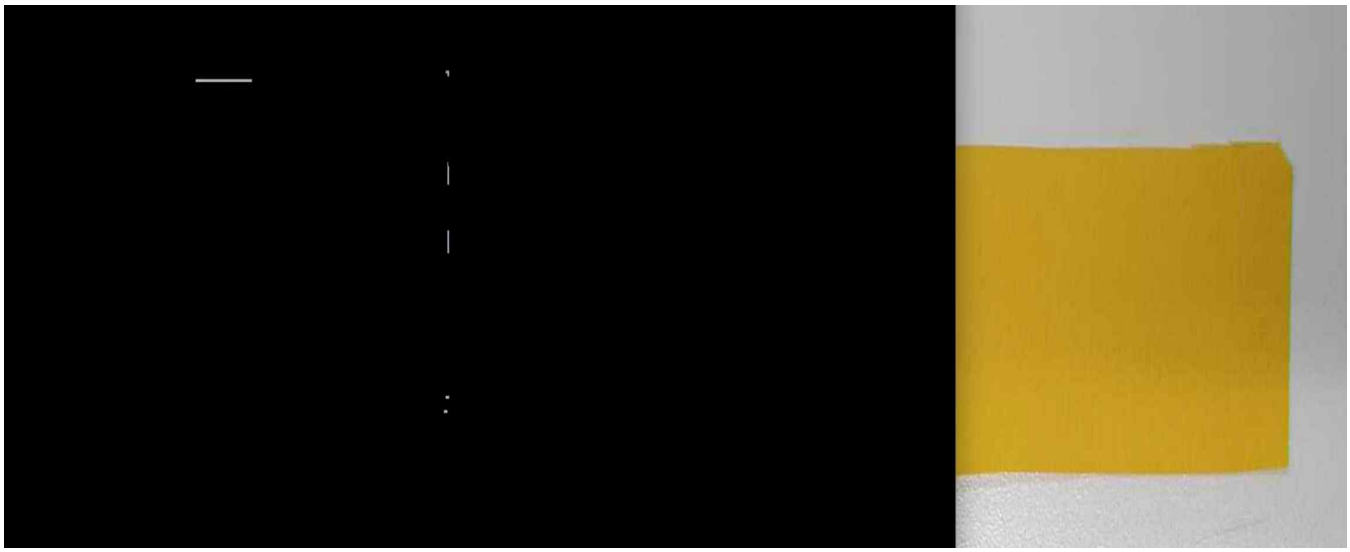
위치의 경우 위의 사진 터미널과 같이 C x 좌표 y좌표, 의 형태로 터미널에 너무 많은 정보가 가기 때문에 구분하기 어려운 점이 있어 앞에 C를 붙여 좌푯값에 대한 출력인지 구분하도록 코드를 짜게 되었습니다.

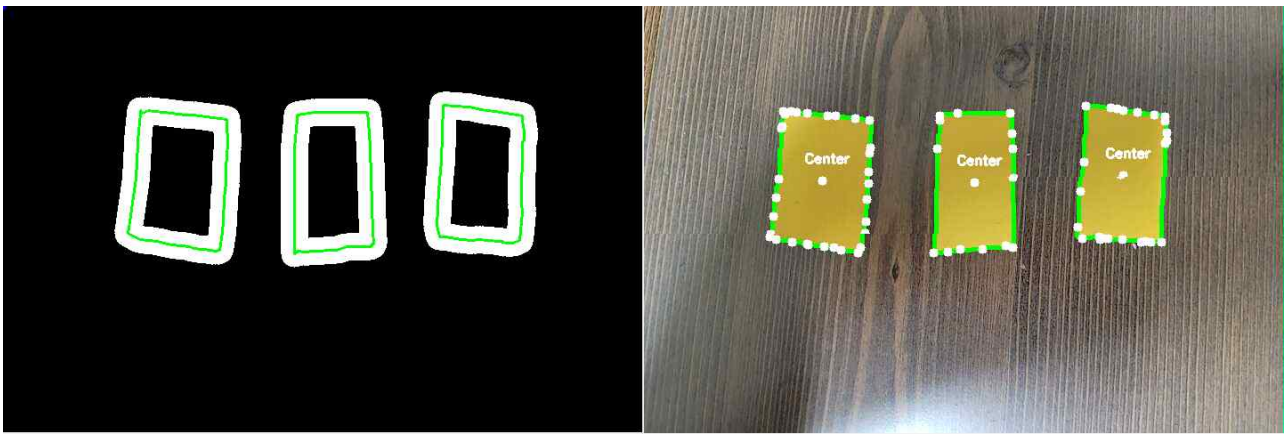
네 번째는 차량의 각도입니다. 차량의 각도 또한 컴퓨터비전에 HSV 기능과 math 라이브러리를 이용하였습니다.

각도의 경우 위치를 알려주는 파란색 종이와 뒷부분에 초록색 종이를 부착하여 앞서 말한 RGB 값을 HSV로 변환하여 색깔을 인식하였으며 두 종이의 angle 값을 math 라이브러리를 통해 계산하여 차량의 앞부분과 뒷부분이 얼마나 기울었는지 계산하고 차량이 바퀴를 통해 이동하며 자연스럽게 각도가 꺾이며 원래 가려던 경로를 벗어날 수 있기에 지속해서 각도 값을 전송하며 해당 각도 값을 Pure persuit 알고리즘에 전송하여 차량 방향에 대한 제어를 만들어 차량이 문제없이 원하는 지점까지 갈 수 있도록 각도 값을 터미널로 계속 보내줍니다.

위의 사진과 같이 앞에 Angle을 붙여 터미널에 출력되는 다양한 정보 중에 각도에 대한 데

이터가 어떤 것인지 구분하고 현재 기울어짐이 어느 정도인지 나타내며 차량이 가면서 바퀴에 의해 점차 기울어지는 것에 대해 제어하는 데에 있어 도움을 주는 코드를 구성하였습니다.

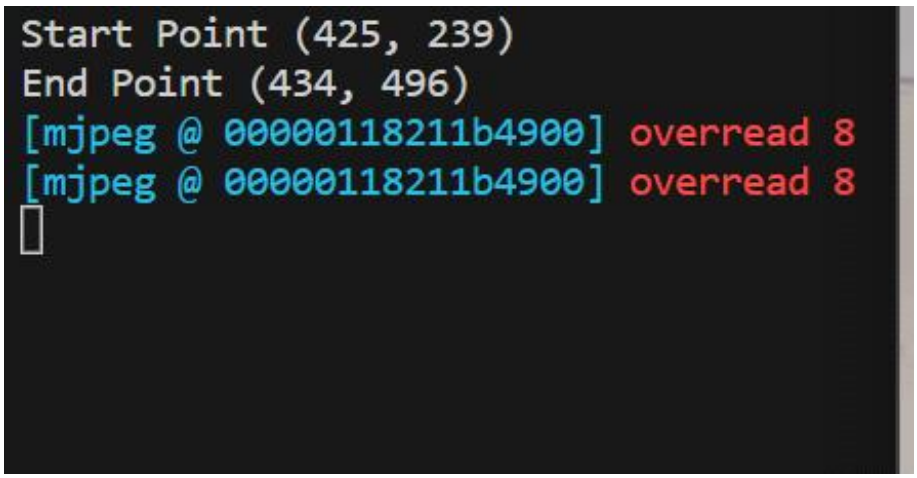




다섯 번째는 장애물입니다.
장애물의 경우 컴퓨터비전의 HSV 기능과 moments 기능을 이용하였습니다.
장애물의 경우 위에 노란색 종이를 붙여 지도에서 현재 장애물의 위치가 어디인지 표현하였으며 장애물의 경우 앞에 나온 차량의 각도 같은 어떤 지점 한 부분을 나타내는 것이 아닌 부피가 존재하기에 컴퓨터비전에 존재하는 moments 기능을 이용하였습니다.
moments는 인식한 물체의 중심이 어디인지 인지하는 기능이며 이 기능을 통해 장애물의 중심점을 찾은 다음 부피만큼의 좌표를 보정하여 코드에서 많이 쓰는 [] 즉 리스트에 담은 다

음 A* 알고리즘에 전송하게 됩니다. 또한 findContours을 통해 노란색 종이의 테두리를 인식하여 어디까지가 장애물 범위인지를 알려주어 보다 정확한 장애물 범위를 리스트에 담아 A* 알고리즘에 전체 전송을 하여 장애물을 정확히 인지하여 차량이 장애물에 부딪히지 않도록 합니다.

위의 사진은 노란색 필터 화면과 카메라 화면을 띄운 사진입니다.
노란색 종이의 테두리를 인식하고 있으며 테두리에 크기 보정을 넣어 벽이 있다가 화면에 띄워주는 형태의 코드입니다.



여섯 번째는 출발점과 도착점입니다.
컴퓨터비전에 있는 EVENT_LBUTTONDOWN 기능과 numpy로 만들어진 좌표계를 이용하여 기능을 만들었습니다.
출발점과 도착점의 경우 카메라가 촬영하고 있는 화면을 마우스 왼쪽 클릭하게 되면 출발점에 대한 좌표가 터미널을 통해 출력이 되며 한번 마우스 왼쪽 클릭하게 되면 도착점에 대한 좌표가 터미널로 출력이 되어 저장됩니다.
여기서 만들어진 출발점과 도착점에 대한 좌푯값은 A* 알고리즘과 Pure Persuit 알고리즘으로 전송이 되며 출발점과 도착점에 대한 값을 바탕으로 위의 두 알고리즘이 동작하게 됩니다.

5 설계평가

컴퓨터비전의 경우 카메라를 센서로 이용하여 제어, 데이터를 전송하는 역할이기에 주변 환경에 따른 변수가 상당히 많았다.
처음에 문제가 되었던 부분은 빛에 대한 부분이었다.
스티커의 경우 겉 부분이 코팅이 되어있기에 아무리 빛에 대한 값을 유지하려 해도 코팅된 스티커 겉면으로 인해 빛이 반사되며 빛에 의해 색이 번지게 되며 카메라 화면에서 해당 색깔을 인식하면 주변까지 다 인식하여 정확한 측정이 어려웠다.

이러한 부분을 해결하기 위해 기존 스티커를 이용해서 하는 방식에서 그림판을 통해 RGB 값을 입력하여 파일로 저장한 뒤 해당 파일을 출력하여 차량 위에 설치, 그리고 장애물 위에 설치하여 색 인식 코드인 HSV에서 H에 해당하는 색깔과 S 채도 부분을 어느 정도 고정을 한 다음 측정을 하고자 하는 강의실에 맞는 V 즉 명도 값을 찾아내어 작성하였다.

두 번째로 문제가 되었던 부분은 장애물을 어떻게 나타내야 하는가였습니다
장애물의 경우 위치 같은 점이 아닌 물체이기에 부피가 존재하여 해당 위치만 딱 찍는 것으로는 장애물을 표시하기 어려운 점이 많았습니다.
그래서 어떤 방식을 가져와야 하는지 찾다가 장애물을 직사각형으로 만든 다음 해당 장애물의 중심점을 코드로 찾아낸 뒤 중심점으로부터 가장 멀리 있는 테두리를 리스트 안에 집어넣은 다음 보정을 추가하여 장애물의 위치를 카메라상에 띄우는 방식을 택하였다.

세 번째는 설계 작품에 대한 설치였으며
처음에는 천장에 카메라를 달아서 위에서 아래로 촬영하는 것을 계획했으나 천장이 높은 대형강의실 같은 경우 도저히 설치가 안 된다는 점, 카메라가 지도의 정중앙으로 들어와야 한다는 점과 장소를 바꿀 때마다 빛에 의해 명도와 채도의 값이 바뀌기에 설치를 어떻게 해야 하는지에 대한 고민을 했으며
2층 자습실에서 테스트할 땐 앞에서 말한 휴대전화 거치대에 휴대전화를 꽂은 뒤 거치대 밑부분을 천장에 붙여 위에서 아래로 촬영하는 형태로 하였으며
계단강의실의 경우 교탁에 슬라이드로 연장하는 부분에 거치대를 설치하고 밑에 트랙을 설치해 카메라를 위에서 아래로 촬영, 카메라가 지도 중앙으로 들어오는 조건과 위에서 아래로 찍어야 하는 점 모두 만족하는 방식을 택하였다.

네 번째는 전원 공급의 문제였으며 컴퓨터비전과 와이파이를 통해 화면을 공유한다는 점 즉 그래픽과 통신을 많이 사용하여 그만큼 배터리 소모량이 커 실험을 오래 하기 어렵다는 문제가 자주 발생하였으며
 전원 공급에 대해 처음에는 리튬이온배터리 4개를 이용해 전원을 설치한 다음 레벨 다운 컨버터에 있는 가변저항기를 통해 5V로 설정한 후 레벨 다운 컨버터에서 입력 전압이 5v인 아두이노에 연결 그리고 라즈베리파이 4b의 입력 전압도 5v이기 때문에 전원을 공유하는 방식으로 가져가려 하였으나
아두이노에 연결된 소자가 많은 점과 라즈베리파이 또한 앞에서 말한 너무 무거운 프로그래밍 즉 컴퓨터비전과 통신을 이용하여 전류 소모 값이 컸으며 레벨 다운 컨버터에 있는 서큘레이터 또한 용량이 작아 원활하게 동작할 수 있는 수준의 전류를 불러오지 못해 동작이 멈추는 현상이 계속되어 아두이노 전용 전원 공급 기기와 라즈베리파이의 경우 멀티탭으로 끌어와 사용하는 것으로 바꿨다.

6. 설계참고자료

https://www.inflearn.com/course/%EB%82%98%EB%8F%84%EC%BD%94%EB%94%A9-%EC%9D%B4%EB%AF%B8%EC%A7%80%EC%B2%98%EB%A6%AC
(인프런 opencv 이미지처리 강의)

https://junworld.tistory.com/25
(객체 인식 및 중앙점과 기울기 추적)