

창업연계공학설계입문 AD 보고서

조번호	4 분반 3 조				
조장/조원	성명	학과	학번	학년	비고
조 장	유상록	소프트웨어학부	20191630	1	
조 원	이도원	소프트웨어학부	20191636	1	
	이승화	소프트웨어학부	20191642	1	
	양진우	소프트웨어학부	20191622	1	
	김하연	소프트웨어학부	20180086	2	

2019 년 12 월 18 일

I. 창업연계공학설계입문 AD 보고서

차선 인식과 표지판에 따른 속도 조절 및 장애물 회피 기능의 자율주행자동차 연구

국민대학교 소프트웨어융합대학

소프트웨어학부

4 분반 3 조

2019 년 12 월 18 일

차선 인식과 표지판에 따른 속도 조절 및 장애물 회피 기능의

자율주행자동차 연구

연구의 목적, 내용, 방법

1 연구 목적

1.1 속도 조절 연구

교통사고 발생의 가장 큰 이유 중 하나는 운전자의 규정 속도 미준수이다. 실제로 뉴스에 따르면 속도 위반으로 인한 사고율은 4 년 새 43%나 급증했다고 한다. 이처럼 규정 속도 미준수로 인해 사고 발생 위험이 꾸준히 오르고 있는 요즘, 자율 주행 자동차의 속도 조절 기능이 사고율을 대폭 줄여줄 것이라고 예상되고 있다. 갖가지 이유로 사람들은 속도 위반을 한다. 조금만 더 빠르게 달린다고 사고가 일어나겠냐는 안이한 마음으로 인한 것이든, 아니면 정말로 바쁜 일이 있어 빨리 달려야 해서이든 여러 이유로 속도 위반으로 인한 사고율은 꾸준히 오르고 있다. 특히나 빠른 속도로 주행해야 하는 고속도로에서는 속도 미준수로 인해 인명 피해 문제가 심각한 실정이다. 사람이 아닌 정확한 데이터를 토대로 돌아가는 센서로 속도 조절을 한다면 특별한 오류가 있지 않은 이상 속도 위반 가능성은 거의 0 으로 수렴하게 될 것이다. 이러한 이유 때문에 속도 조절을 연구하기로 결정했다.

1.2 장애물 회피 연구

장애물을 피하는 일은 쉽지 않은 일이다. 사람의 순발력에는 한계가 있기 때문이다. 시야확보가 쉬운 낮에는 장애물을 회피하는 일이 쉽겠지만 밤과 같이 시야가 어두운 경우, 혹은 다른 여러 이유로 시야 확보에 어려움을 겪을 경우에는 장애물 발견이 늦어 회피에 실패하는 경우가 있다. 그렇기에 초음파 센서의 도움을 받는다면 사고 발생률을 낮출 수 있을 것이라고 생각하였다. 사람의 시각이 아닌, 초음파로 장애물을 탐색하는 것이기

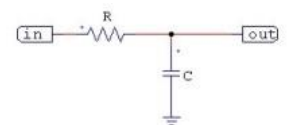
때문에 장애물 회피에 있어 훨씬 정교한 컨트롤이 가능하다. 또한 장애물과의 거리가 일정해지면 바로 차체에 브레이크를 걸기 때문에 정확한 값을 토대로 회피하는 자동차가 장애물을 인식하고 피하는 시간이 더 짧다. 이처럼 센서를 통해 장애물을 피하는 것이 더 안전하고 빠를 것 같다는 생각에 장애물 감지를 연구하기로 결정했다.

2 연구 내용

1. 초음파.

처음 초음파를 사용했을 때에는 자동차가 갑작스럽게 멈추는 현상이 빈번하게 일어났다. 뿐만 아니라, 똑같은 변수의 값을 집어 넣어도 다른 위치에서 멈추는 것을 직관적으로 알 수 있었다. 그래서 우리는 이러한 초음파의 흔히 말하는 '튀는 값'들을 잡기위해 필터를 이용하고자 했다.

우리가 처음으로 사용한 필터는 LPF(Low Pass Filter)였다. 이 필터는 저항과 캐피시터로 구성된 회로에서 저주파를 통과 시킬때 그 값을 조정하는 필터 중 하나다. 그러나 LPF 를 이용했음에도 불구하고, 가끔 원하지 않은 위치에서 멈추는 경우가 있었다.



그 다음에 사용한 필터는 리스트를 만들고 각각의 값들에 가중치를 주어 그 가중치의 평균을 내어 사용하는 필터를 이용하기로 하였다. 이 필터는 모든 값들과 가중치를 더하기 때문에 값이 높아져 거리를 조정하는데에 힘들어 있었지만, 위의 방법 보다 정교하게 움직였다.

$$\omega_c = 2\pi f_c = \frac{1}{RC}$$

$$f_c = \frac{1}{2\pi RC} [\text{Hz}]$$

$$T = 2\pi RC [\text{sec}]$$

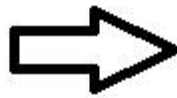
2. 차선인식

처음 차선인식을 했을 때에는 카메라로 우선 ROI 를 잘라낸 다음 BGR2HSV 를 이용하여 HSV 로 바꾼 다음 inRange 로 이진화후, canny 를 이용하여 차선을 검출하였다. 그러나 이 경우 차량이 계속적으로 조금씩 움직이는 현상을 우리는 발견했다. 그래서 우리는 canny 를 이용하여 차선을 검출을 하는 것 까지는 똑같이 하였지만, ROI 를 바로 아래차선이 아닌, 삼각형으로 하였고 ROI 에서 차선이 있느냐

없느냐가 아닌, 차선의 기울기가 있는가 없는가를 이용하여 검출하였다.



기존의 ROI



새로운 ROI

3. QRCode

QRCode 를 파이썬에서 인식을 하기 위해서는 pyzbar 이라는 라이브러리가 필요했다. 그러나 이 라이브러리는 우리에게 주어진 xycar 에는 기본으로 주어져 있지 않아, pip 을 이용하여 install 해줘야만 했다. 그러나 우리의 xycar 은 외부의 네트워크와는 연결이 되어 있지 않은 상황이 었다, 그래서 우리는 Lan 선을 구입하여 xycar 에 연결을 한 다음, 하나의 랜선을 우리의 노트북에 연결하여 install 을 할려고 했지만, pip 마저 막혀있는 상황이었다. 그래서 우리는 수동으로 pyzbar 을 install 하는 방법을 찾아내었고, 설치에 성공을 하여 qrcode 를 무사히 decode 할 수 있었다.

3 연구 방법

가장 먼저 장애물을 인식하기 위해서 초음파 센서를 이용하였는데 usonic_data 에 6 방향의 초음파 센서로 읽어들이는 값을 받아와 앞에 장애물이 몇 이하로 있는지부터 확인하게 하였다.

하지만 초음파 센서가 일정값 이하로 떨어졌을 때 멈추는 코드를 짜니 이 초음파 센서가 몇몇 튀는 값을 가지게 되어서 장애물이 없음에도 멈추는 현상이 발생하였다.

이 값을 지우기 위해서 리스트를 만들었고 그 리스트에 초음파 값을 집어넣어 가중치를 주고 평균을 구하여 튀는 값을 없애 멈출수 있게 해주었다.

다음으로 구현한건 라인디텍터이다.

차선을 읽고 그 차선 안에서 달리기 위해 가장 먼저 카메라로 가져온 이미지를 bgr 에서 hsv 로 바꾼 다음 이진화를 하여주었다.

이진화한 값을 이용하여 차선을 검출하기 쉽도록 해주었고 blur 를 이용하여 이미지를 선명하게 만든 후 canny 로 선을 따서 이 선의 기울기에 따라 좌회전과 우회전을 정해주었다.



창업연계공학설계입문 수업에서 AD 발표를 할 때 사용한 qr 코드 인식 후 속도 조절은 pyzbar 라는 파이썬 모듈을 사용하였다.

자이카 카메라를 이용하여 수집한 이미지를 검사하는데



다음과 같은 qr 코드는 인식할 시 110 이라는 값을 리턴 받는다.

pyzbar 에서는 가져온 이미지에서 qr 코드로 인식되는 부분을 잘라 그 인식된 qr 코드의 값을 가져오는데 리턴 받은 110 이라는 값을 스피드 변수에 집어넣어 autodrives 코드에서는 이 스피드 변수로 달리는 속도를 조절한다.

장애물 피하기는 초음파 센서를 이용하였고 일정치 이상 장애물이 가까워졌을 때 잠시 정차 후 후진하여 뒤로 빠진 다음에 오른쪽으로 빗겨가 장애물과의 충돌을 막는다.

#오른쪽으로 빠지는 함수

```
def turn_right(self):  
    #후진을 하기 전 멈추는 부분  
    for i in range(2):  
        self.driver.drive(90,90)  
        time.sleep(0.1)  
        self.driver.drive(90,60)  
        time.sleep(0.1)
```

#1 초 정도 후진하여 장애물과의 거리를 벌린다.

```
for back_cnt in range(10):
```

```
    self.driver.drive(90,60)
```

```
    time.sleep(0.1)
```

#오른쪽으로 빠지고

```
self.driver.drive(170,self.line_detector.speed-5)
```

```
time.sleep(1)
```

#왼쪽으로 돌아 원래 자리로 돌아온 후에

```
self.driver.drive(40,self.line_detector.speed-5)
```

```
time.sleep(4)
```

#중앙에 자리잡는다.

```
self.driver.drive(110,self.line_detector.speed-5)
```

```
time.sleep(1)
```

#초음파 센서가 50 보다 작을 때 리스트에 전송

```
if usonic_data[1] < 50:
```

```
    front_mm.add_sample(usonic_data[1])
```

#가중치를 곱하여 나온 평균 값이 40 보다 작을 때 장애물이 있다고 인식 후 함수호출

```
if front_mm.get_wmm() < 40:
```

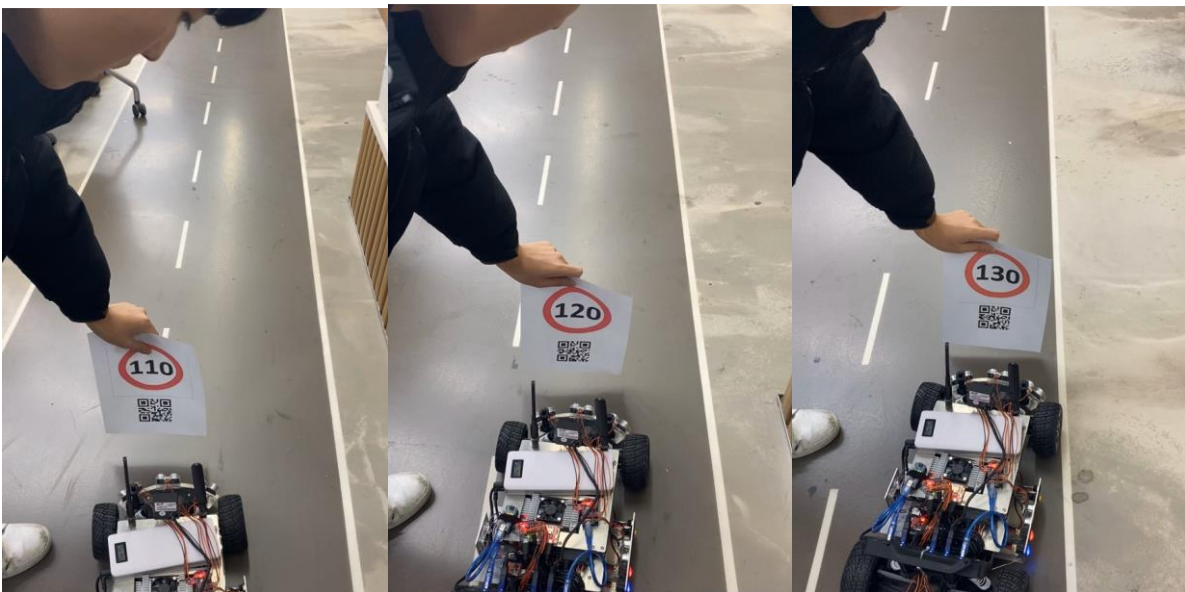
```
    car.turn_right()
```

```
    front_mm = MovingAverage(3)
```


2 장 연구의 결과

qr 코드 인식 후 속도 조절, 장애물 회피

창연공 AD 프로젝트의 첫번째 주제인 qr 코드 인식은 xycar 속도 기준 110, 120, 130 표지판을 만들고 그 밑에 110, 120, 130 값을 불러오는 qr 코드를 생성하여 xycar 가 인식 하여 표지판의 값에 따라 xycar 속도가 바뀌게 하였다. 그러나 xycar 가 qr 코드를 인식할때, 라인을 벗어난 곳에서는 qr 코드 인식을 잘하지못하고, 높은곳에 있는 qr 코드를 인식을 잘못한다는 문제점이 존재하여 카메라 바로 앞에서 qr 코드를 인식을 해야만 했다. 주행을 하다가 xycar 가 멈추었을때, xycar 가 qr 코드를 인식할 수 있게 카메라 앞에 qr 코드를 대면 xycar 의 속도가 바뀌게 하였다. 그 결과, xycar 가 qr 코드를 잘 인식하여 qr 코드의 값에 따라 속도가 변화하는 결과를 얻게 됐다.



장애물 회피는 말그대로 xycar 가 앞에있는 장애물을 인식하고 장애물을 회피하여 지나가는것이다. 맨 처음에는 xycar 가 장애물을 인식하였을때 후진을 하지 않고 장애물을 회피하고 지나가는 코드를 짰었는데, xycar 가 장애물을 인식하였지만 장애물을 잘 회피하지 못하는 문제가 발생하였다. 그래서 장애물을 인식하고 후진한 뒤에 장애물을 회피하는 코드를 짰더니 장애물을 잘 회피하였다.



3. 결 론

QR 코드를 이용해 속도 제한 표지판에 맞추어 속도를 조절하는 기능은 QR 코드가 Xycar 의 전방카메라 바로 앞에 있을 경우 인식하는 정도로 구현하는 데에 성공했다. 하지만 실제 도로에서는 차량과 속도 제한 표지판 사이의 거리가 어느 정도 있을 것이므로 보다 먼 거리에서도 인식이 가능하도록 코드를 보완할 필요가 있다. 또한, 도로의 상황에 따라 제한 속도보다 매우 낮은 속도로 주행해야 하는 경우가 있을 수 있다. 이 경우 어떻게 속도를 조절할지 상황에 따라 속도를 잘 조절할 수 있도록 다양한 케이스를 만들어야 할 것이다. 뿐만 아니라 속도 표지판의 글자 자체를 인식하는 것이 아니라 QR 코드를 인식 후 속도값을 받아와 인식하는 것이므로 모든 표지판에 QR 코드를 설치해야 하는 비용적 문제도 발생할 것이다.

장애물 인식 및 회피 기능의 경우 너비와 길이가 짧은 물체의 경우 잘 회피했지만, 그렇지 않은 경우 잘 회피하지 못하는 문제가 있었다. 실제 도로에서는 바위나 나무 같이 매우 큰 장애물이 있을 수 있으므로 이러한 경우에도 잘 피할 수 있게 보완해야 할 것이다. 이는 초음파 센서의 한계 때문이기도 한데, 필터를

사용해 보완하기는 했지만 초음파의 경우 튀는 값들이 많이 발생하기 때문에 정확성이 높지 않다. 초음파 센서가 아닌 다른 센서를 활용해 장애물을 회피하는 방법도 고려해보아야 할 것 같다. 또한, 장애물을 인식하면 잠시 후진했다 방향을 바꿔 장애물을 회피하도록 구현했는데 실제 도로에서 이러한 주행 방법은 어려울 수 있다. 따라서 속도를 줄이고 즉각적으로 방향을 회피할 수 있도록 잘 구현해야 할 것이다. 다만 이 경우에도 좌우 차선의 후방에서 차량이 주행하고 있을 경우 사고 발생의 위험이 있기 때문에 좌우와 전후방의 상황을 잘 고려해 장애물을 회피해야 할 것이다.

Xycar 로 차선 인식, 속도 조절, 장애물 회피 기능을 구현해보면서 자율주행자동차의 편리함과 개발 시의 어려움을 깨달을 수 있었다. 전후진, 방향 변경, 정지 등 자동차의 일반적인 기능을 구현하는 것은 어렵지 않았지만 차선 인식, 장애물 인식 등 환경을 인식하고 주행하는 것이 매우 어려웠다. 자율주행자동차 개발에는 정말 많은 노력과 끈기, 책임감이 필요하다는 것을 깨달았고 생각보다 고려해야 하는 상황이 많다는 것을 느꼈다.