

# 소프트웨어프로젝트 II

## AD 프로젝트

과제명	OpenCV 를 통한 T자 주차 및 지진 감지
소속	국민대학교 소프트웨어학부
학번	20191946, 20191651, 20191652 , 20191657, 20191659
성명	이재익, 이혁규, 이현, 장재만, 전기정
문서 제목	소프트웨어프로젝트 I AD 프로젝트
교수	이시윤 교수님

Version	1.5
Date	2019.12.18

### 문서 정보 / 수정 내역

Filename	OpenCV 를 통한 T자 주차 및 지진 감지
원안작성자	20191651 이혁규
수정작업자	20191651 이혁규

수정날짜	작성자	Revision	추가/수정 항목	내 용
2019.11.19	이혁규	1.1	아이디어 선정	OpenCV 를 활용한 신호등 인식
2019.11.26	이혁규	1.2	아이디어 구체화	IMU 센서를 통해 지진 감지하기
2019.12.03	이혁규	1.3	기능 구현	T자 주차 및 위의 2 가지 기능
2019.12.14	이혁규	1.4	기능 통합	3 가지 기능을 자율주행 과정에 적용
2019.12.18	이혁규	1.5	테스트, 영상 확보	주행 테스트 및 동영상 확보

## 목 차

1	개요.....	4
1.1	프로젝트 개요.....	4
1.2	추진 배경 및 필요성.....	4
2	개발 내용 및 결과물.....	5
2.1	.....	목표
5		
2.2	.....	연구/개발 내용 및 결과물
6		
2.2.1	연구/개발 내용.....	6
2.2.2	.....	활용/개발된 기술
6		
2.2.3	.....	문제점 및 해결 방안
6		



	2.2.4.....	결과물 목록
6		
3	자기평가.....	7
4	참고 문헌.....	7

# 1 개요

## 1.1 프로젝트 개요

4차 산업혁명이 진행되면서 자율주행 자동차는 가장 주목 받는 분야 중 하나로 떠오르고 있다. 자율주행 자동차는 운전자가 직접 조작하지 않아도 자동차가 주행환경을 인식하여 위험을 판단하고 주행경로를 계획해 스스로 운전하는 자동차로 정의된다. 과거 자동차는 기계공학적인 기술이 주를 이루는 분야였다. 하지만 현재는 컴퓨터공학과 첨단센서를 필요로 하는 소프트웨어적 산업이 되어, 보다 정밀한 측정과 분석을 필요로 하게 되었다. 창업연계 공학설계입문 수업에서는 이러한 시대의 흐름과 수업의 내용에 따라 자율주행의 소프트웨어적인 부분을 보다 잘 이해하고 분석하기 위해, 자율주행 자동차의 기초가 되는 운전자 보조 시스템의 일부분과 OpenCV의 실시간 영상처리를 이용하여 미국 교통부 도로교통안전국(NHTSA)에서 제시한 Level 2 단계(복수의 주행 기능 융합 보조)를 구현하였다. 우리는 이에 그치지 않고 AD 프로젝트에서 OpenCV를 이용한 T자 주차와 IMU 센서 데이터를 바탕으로 지진 감지 소프트웨어를 구현하였다. 이를 통해 보다 더 융합적인 주행 보조 기능을 완성할 수 있었다.

## 1.2 추진 배경 및 필요성

앞서 말하였듯이, 4차 산업혁명이 진행되면서 앞으로 자율주행 자동차 시대가 도래할 것이다. 그러한 미래의 상황에서 현재 자동차를 운전하기 필요한 면허가 필요할 것인가? 이러한 의문점이 본 AD 프로젝트의 시발점이 되었다. 이에 대한 답은 애초에 모순이었다. 자율주행이 된다는 것 자체가 운전자가 주행에 신경을 쓸 필요가 없다는 것이었다. 하지만 우리는 이에 동기를 얻어, 현재 운전면허 시험장에서 시행되고 있는 여러 주행 구간들을 구현하는 것을 AD 프로젝트의 목표로 삼았다. 첫째로, 운전면허 수험생들이 가장 어려워하는 직각 주차 구간을 OpenCV를 활용해 구현하기로 하였다. 둘째로, 지진이라는 돌발 상황을 연출해 이에 대응하는 구간을 구현하기로 하였다. 위의 내용들은 미래 자율주행 자동차가 필수적으로 갖춰야 하는 기능이다. 그렇기에 대단한 알고리즘은 아니더라도, Level 2 수준에서는 충분히 의미가 있는 활동이라 생각한다. 따라서 우리 조는 운전면허시험을 모토로 여러 구간에 대한 알고리즘을 구현할 것이다.

## 2 개발 내용 및 결과물

### 2.1 목표

우리 조의 목표는 다음 3 가지로 정의할 수 있다 .

#### 1. 신호 구간

- 신호(특정 색상)을 인식하여 특정 구간 임을 인식한다.

#### 2. 직각 주차 구간(T자 주차 구간)

- 신호 인식을 통해 직각 주차 구간임을 인식하면 , T자 주차를 진행하고 초음파 센서를 통해 주차를 조정한다. 그 후, 후진하여 다시 코스를 주행한다.

#### 3. 돌발 구간

- 지진이라는 돌발 상황을 연출하여 그 상황에 정지한다.

### 2.2 연구/개발 내용 및 결과물

## 2.1.1 연구/개발 내용

### 1. 신호구간 : 빨간색 인식

- 카메라 프레임을 받아와 이를 HSV 채널 형식으로 변경한 후, 지정 된 범위에 맞게 ROI를 설정해주었다. RED라는 색상을 인식하기 위해 빨간색 Hue, Saturation, value 범위를 나누어 주었고 cv2.inRange() 함수를 통해 ROI를 이진화 해주었다. 이 ROI에서 흰색 점의 비율이 0.2 보다 높으면 1을 아니면 0을 리턴하게 해주었다.

```

def trafficLight(self):
    screen = self.cam_img
    hsv = cv2.cvtColor(screen, cv2.COLOR_BGR2HSV)
    roi = hsv[120:370,470:640]

    scan_height, scan_width = 250, 170
    pixel_value = 0.2*scan_height*scan_width
    lower_red = np.array([-10, 100, 100], dtype=np.uint8)
    upper_red = np.array([10, 255, 255], dtype=np.uint8)

    red_range = cv2.inRange(roi, lower_red, upper_red)
    cv2.imshow("1", red_range)
    cv2.waitKey(1)

    red_value = cv2.countNonZero(red_range)

    return_value = 0

    if (red_value> pixel_value):
        return_value = 1

    return return_value

```

## 2. 주행 알고리즘

- 빨간색이 인식되었다면 forward 함수를 실행한다. 또 IMU 센서로부터 데이터들 받아오고 pitch 값이 -5 보다 작으면 3초간 정지한다.

```
def trace(self):
    redSignal = self.line_detector.trafficLight()
    if redSignal:
        print(redSignal)
        self.forward()
    r,p,y = self.imu.get_data()
    obs_l, obs_m, obs_r, obs_sl, obs_sr, B = self.obstacle_detector.get_distance()
    line_l, line_r = self.line_detector.detect_lines()
    self.angle = self.steer(line_l,line_r)
    speed = self.accelerate(self.angle, obs_l, obs_m, obs_r, p, obs_sl, obs_sr, B)
    self.driver.drive(self.angle + 90, speed + 90)
    print(p)
    if (p < -5):
        time.sleep(3)
```

### 2-1. forward 함수

- angle 은 35, speed 는 120 으로 3 초 간 주행한다.



```
def forward(self):  
    for i in range(30):  
        self.driver.drive(35, 120)  
        time.sleep(0.1)
```

### 3. accelerate 함수

- 만약 양 옆 및 정면 방향에 장애물이 40cm 보다 가까이 있다면 backward() 함수를 실행한다. 또 p의 값이 -5 보다 작다면 스피드는 0으로 리턴한다.





```
def accelerate(self, angle, left, mid, right, p, side_left, side_right):  
    if mid < 40 and side_left < 40 and side_right < 40:  
        self.backward()  
  
    if p < -5:  
        return 0  
  
    if (angle <= -25 or angle >= 35):  
        self.speed = 37  
    else:  
        self.speed = 46  
    return self.speed
```



### 3-1. backward 함수

- 전진에서 후진으로 변환하기 위한 코드를 작성해주었고 angle 은 35, speed 는 70 으로 5 초 동안 주행시켰다.

```
def backward(self):  
    for i in range(2):  
        self.driver.drive(90,90)  
        time.sleep(0.1)  
        self.driver.drive(90,60)  
        time.sleep(0.1)  
    for i in range(50):  
        self.driver.drive(35, 70)  
        time.sleep(0.1)
```

### 2.1.2 활용/개발된 기술의 알고리즘

- 1) HSV 색상 모델링을 통한 이진화 알고리즘
- 2)
- 3)

### 2.1.3 문제점 및 해결 방안

- 현실적으로 시간의 제약이 컸던 것 같다. AD 제안 발표 이후, AD 프로젝트에 많은 노력을 기울이려 하였지만 차선 추종 자율주행 알고리즘을 구현하는 것만으로 벅찼었다. 또 주마다 연쇄적으로 유레카 프로젝트 대회, 차선 인식 자율주행 등과 같이 큼지막한 과제들이 있어서 AD 프로젝트에 많은 노력을 쏟아 붓지 못했다. 뿐만 아니라, 이런 과정들이 끝나니 바로 다음 주가 시험이고 각자의 사정들도 있어 모일 시간이 마땅치 않았다. 하지만 매 주 모이는 정규 팀 활동 시간을 밀도있게 활용하여 제시한 목표들을 다 이룰 수 있었던 것 같다.

### 2.1.4 결과물 화면

<https://github.com/ehyeok9/Xycar>

- 위 링크에 영상이 첨부되어 있다.

### 3 자기평가

연번	팀장(원)	역할 수행 내용	수행 후기
1	이현	프로그래밍	처음부터 어떤 알았으면 좋았을 정보들이 수업 후반에 배치되어있어서 1 주면 끝났을 작품을 2 주 이상으로 끌고간것이 조금 아쉽다. 다음 학년이 이 수업을 들을때 뺄이치지 않게 미리 정보를 제시해주면 그들에게 도움이 될거 같다.
2	전기정	프로그래밍	교안에 소개되어있는 차선 검출방식은 기본이다. 나쁘다고 말하지는 못하지만 주행환경에 변수가 많이 생긴다. 차선검출하는 라이브러리 또는 오픈소스를 이용하면 잘될 수도 있지만 1 학년 모두가 하기에는 어렵지 않나 싶다. 각자의 생각이 달라서 차선을 검출한다거나 모든 상황에 대해서 의견이 갈렸다. 내가 생각하지 못한 부분을 조원이 생각하거나 그 반대일 경우도 많았으며 계속하여 앞으로 나아갈 수 있었다. 2~3 주동안 팀 프로젝트를 하면서 팀워크의 중요성을 깨닫게 되었으며 생각의 폭도 넓혔다.  개인이 아니라 5 명에서 프로젝트를 하다보니 순탄치 진행되지 못한 적도 있었지만, 많은 경험을 했다고 생각한다.
3	장재만	프로그래밍	수업 방식 자체는 알차고 좋았다. 직접 자동차를 움직이면서 값을 찾아 보는 활동을 주로 했는데 그 과정에서 느낀 점이 좀 있다. 지속적



			<p>인 알고리즘의 개선을 통해서 차근차근 개발 환경을 만들어 가야 했는데 그게 잘 이루어지지 않았던 것 같아서 아쉬웠다. 아무래도 5명이 함께 수행하는 프로젝트이다보니 새로운 알고리즘을 투입해도 잘 되지 않았을 때의 리스크와 그에 소요되는 시간을 고려하다보니 개발 속도가 느려졌던 점도 아쉬웠던 점 중 하나다. 나 자신의 입장에서 생각해봤을 때는 그게 가장 먼저 해야하는 것처럼 느껴졌지만 팀원들이 다른 생각을 가지고 있는 경우도 있어서 충돌이 일어나기도 했다. 혼자서 했다면 그냥 내 생각대로 하면 되지만 다른 사람과 함께 할 때는 충분히 납득이 갈 만한 설명을 지속적으로 해야 한다는 사실이 번거로웠지만 나중에는 받아들이게 되었다. 이번 조별과제는 어떤 방식으로 공동의 프로젝트를 효과적으로 수행할 수 있을 지를 고민하게 만들었다는 점에서 의미가 있었다.</p>
4	이재익	프로그래밍	<p>5명이라는 많은 인원이 참여하는 팀 플레이에 처음 참여하면서 많은 점을 느꼈다. 다들 팀 플레이에 대한 경험이 적어서 시간 날 때마다 프로젝트를 진행한 점이 아쉽다. 더 프로젝트의 시간이 길고, 역할이나 일정을 잡았으면 더 좋고 빠르게 결과가 도출되었을 것이다. 일정이 어긋나서 코딩 구현 현장에 참여하지 못하면, 다음 일정 때 코딩에 관련된 구체적인 참</p>



			<p>여가 힘들었다. 그래서 조원들이 이론을 바탕으로 구현한 알고리즘을 실제 주행을 하면서 경험을 통해 데이터를 수집했고, 이에 맞게 알고리즘을 고쳐나갈 수 있게 도와줬다. 구체적인 알고리즘에 대해 나도 큰 목소리를 내고 주장했으면, 다른 조원들이 더 편하고 빠르게 프로젝트를 진행할 수 있었을 텐데 그러지 못했다. 뒤에서 서포터를 해주는 경험이 더 많았다. 앞으로 이와 같은 프로젝트를 더 할 기회가 생긴다면, 더 열심히 해보고 싶다.</p>
5	이혁규	프로그래밍	<p>차선 추종 자율주행 소프트웨어다 구현하기 위해 팀원들과 같이 프로그래밍 해보며 많은 것을 느낄수 있었다. 일단 이정도 인원으로 협업을 해보는 것은 팀원 모두가 처음일 것이라 생각한다. 그렇기에 협업 과정에서 많은 갈등과 시행착오가 있었다. 조장으로서 이러한 문제점에 있어서 해결해 나가야 할 답을 제시해주었어야 했는데, 그러지 못한 점이 팀원들에게 미안하다. 비록 많이 부족한 조장이고 팀원이었지만, 그래도 친구들이 많이 배려해줘서 고맙다. 프로그래밍 과정에서 분업을 하기 위해 홀로 hough 변환 알고리즘을 구현하고 자율주행하는데 성공하였지만 팀원들이 경우의 수를 분류하고 여러 과정을 거친 결과물이 보다 안정적이고 의미가 있는 것 같아 그것을 통해 여러 주행테스트를 통과하였다. 그렇기</p>

			<p>에 약간 나의 활동이 주력이 되지는 못했지만 스스로 자율주행을 구현했다는 것에 대해 만족하고 의미가 있는 것 같다.</p>
--	--	--	---

#### 4   참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	Web	창업연계공학설계입문 교안	국민대학교			
2	Web	XyCar B2 Manual	Xytron			
3	Web	Category: OpenCV	wikipedia			
4	Web	ROS	wikipedia			