자율주행 자동차와 온라인 사고 기록 장치에 관한 연구

장덕진* 김택수** 신지윤*** 김인준**** 김현지*****
*명지대학교 전자공학과
**가천대학교 전자공학과
***이화여자대학교 전자공학과
****숭실대학교 전자정보공학부 전자공학과
*****세종대학교 전자정보통신공학과
e-mail: jwer7329@naver.com

A Study on Autonomous Vehicle and Online Event Data Recorder

Duck-Jin Jang*, Taek-Soo Kim**, Ji-Yoon Sin***, In-Jun Kim****, Hyeon-Ji Kim*****
*Electronic Engineering , Myong Ji University
**Electronic Engineering, Ga Cheon University
***Electronic Engineering, Ewha Womans University
****Electronic Eng, Dept of Electronic&Information, Soong Sil University
*****Electronic&Information&Communication Engineering, Se Jeong University

요 약

최근 자율주행 자동차의 기술개발이 활발히 진행되고, 실제 출시까지 이어지는 가운데 자율주행자동차의 사고 발생도 증가하고 있다. 자율주행자동차를 이해하기 위하여 컴퓨터 비전 기반의 자율주행 자동차를 구현하고 사고 발생 시 차량 구동과 관련된 중요 정보를 소실하지 않기 위해 외부 서버에 저장하는 연구를 시행한다.

자율주행 자동차의 주행정보 저장에 관한 연구

1. 서론

현대시대가 4차 산업혁명으로 들어서면서 빅 데이터와 인공지능, 사물인터넷(IoT)에 관한 연구가 활발히 진행되고 있다. 이러한 인공지능과 IoT 기술을 접목시켜 새로운 주행형태의 자동차 개발이 한창인데, 그 기술을 접목시킨 자동차가 바로 자율주행 자동차이다. 자율주행 자동차란, 운전자의 직접적인 개입 없이 자동차 내부에 탑재된 마이크로컴퓨터가 주변상황을 읽어 자동으로 주행가능하게 하는 시스템이다. 많은 기업들이 자율주행 기술을 발전시키고 실제 자동차로 출시하였지만 동시에 자율주행자동차의 사고 또한 증가하고 있다. 최근 사고 발생 시 제조사와 운전자 간의 주장 차이로 책임여부를 가르기가 어려운 경우가 발생하고 있어 차량 내부에 중요 동작을 기록해줄 이벤트 데이터 레코더(EDR)의 필요성이 증대되고 있다. 현재 독일에서는 자율주행자동차의 EDR기능을 수행하는 블랙박스 설치를 의무화 하는 법안을 추진하고 있다. 다만 사고 시 이벤트 데이터 레코더 또한 파손될 가능성이 존재한다.

이를 토대로 본 논문에서는 라즈베리파이(Raspberry Pi)를 이용한 컴퓨터 비전 기반의 자율주행 기술을 설계하고 중요 이벤트 데이터를 외부 서버(Google Sheet)에 저장하는 방식을 제안한다.

그림1. 자율주행 자동차의 사고 사례

2.차량설계

속도제어를 위한 2개의 모터, 방향 제어를 위한 1개의 모터가 포함된 Sunfounder Kit로 차량 차체를 구성하였다. 계산과 제어에 이용할 라즈베리파이(Raspberry Pi)는 Raspberry Pi 3 Model B를 이용하였다. 차선인식을 위해 Raspberry Pi Camera V2를 사용하여 차량의 방향을 제어하였고, 물체 인식을 위해 초음파센서(Arduino Ultrasonic sensor C22)를 사용하여 전방 장애물 출현 시 속도를 제어하도록 설계하였다. 아래의 첫 번째 그림은 차량 전반부의 시스템 설계도이고, 두 번째 그림은 차량의 H/W 회로도이다.

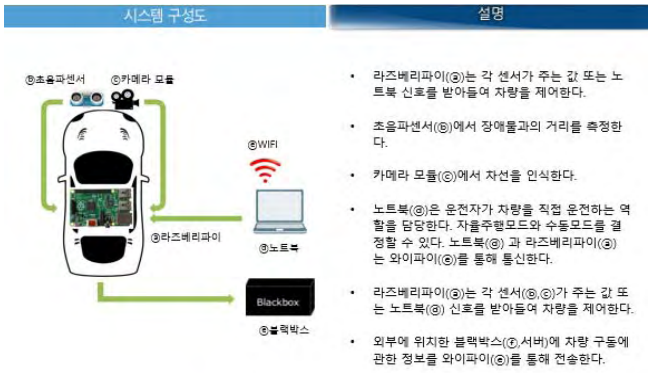

그림2-1. 차량 설계 설명도

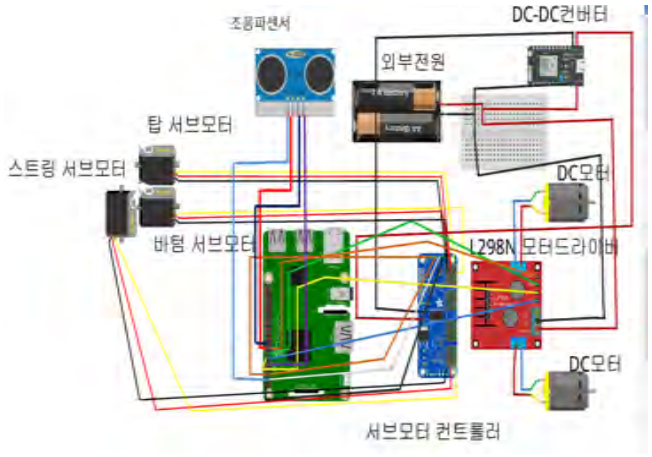

그림2-2. 차량 H/W 회로도

3. 자율주행 시스템 설계

3.1 차선인식

차선인식은 Raspberry Pi Camera와 Open CV를 이용하였다. Open CV의 코드는 Udacity의 advanced-lane-detection[1]을 수정하여 이용하였다. 본 코드에서는 thresholding을 이용하여 이진 이미지를 생성한다. 생성된 이진 이미지는 투영변환(Perspective Transform)을 통해 이미지를 공중에서 내려다보는 형태(aerial viewpoint, Bird's-eye view)로 변환 한다. 변환된 이미지에서 히스토그램(histogram)과 2차 다항식을 이용하여 레인을 검출한다. 그 후 레인의 곡률 반경을 계산하여 자동차의 회전에 이용한다.

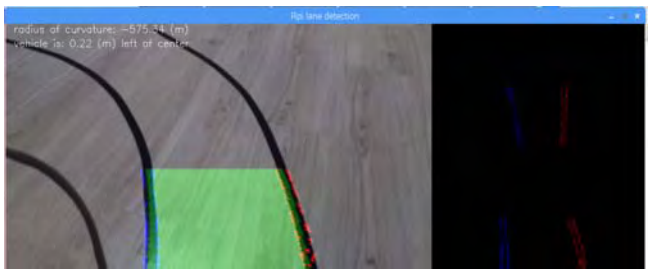

그림3-1. 차선인식

3.2 거리인식

거리 인식은 초음파 센서(HC-SR04)를 이용한다. 앞차 또는 장애물까지의 거리를 인식하고 충돌하지 않도록 모터의 속도를 조절한다. 거리측정의 정확성을 높이기 위해서 거리측정 값을 3개를 얻은 뒤, 3개의 평균값을 구하여 그 값을 통하여 거리에 대한 모터의 속도를 제어한다.

```
def measure_average():
  # This function takes 3 measurements and
  # returns the average.

  distance1=measure()
  time.sleep(0.1)
  distance2=measure()
  time.sleep(0.1)
  distance3=measure()
  distance = distance1 + distance2 + distance3
  distance = distance / 3
  return distance
```

그림3-2. 거리인식에 대한 수치 값 계산 Function Code

3.3 신호등인식

신호등인식은 Raspberry Pi Camera와 Open CV를 이용하였다. 허프 변환(Hough transformation)을 이용해 원을 인식하고 원 안의 빨간색과 초록색을 판별하기 위해 RGB에서 HSV 로 색상을 변환시켜 채널을 분리시킨 후 각 채널에 대한 범주를 넣어 신호등의 색을 감지하였다.

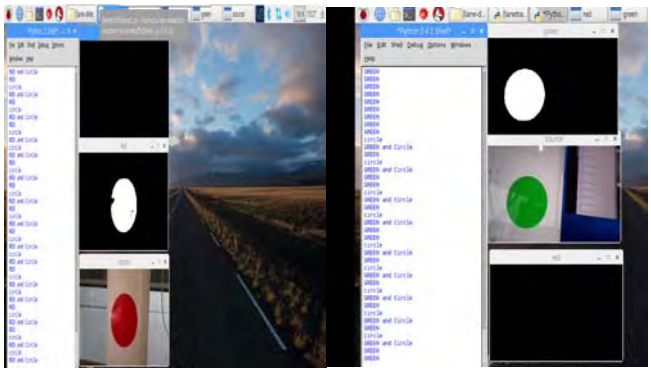

그림3-3. 신호등인식 (좌측 Red , 우측 Green)

4. 이벤트 데이터 레코더(EDR)

차량의 사고, EDR 파손, 소비자 주행습관 등의 정보를 얻기 위해 차량의 중요 동작을 차량 외부 서버에 저장한다. 저장할 서버는 구글시트 (Google Sheet)를 사용하였다. 시간, 주행모드, 레인의 곡률반경, 초음파 센서 값, 모터 동작이 실시간으로 구글시트 (Google Sheet)에 저장된다. 실시간으로 업로드를 하기 위해서는 자동차는 항상 네트워크에 연결되어야 한다. 이 기술을 통하여 차량 내부의 블랙박스가 물리적으로 손상되어도 데이터 복구를 빠르게 할 수 있다. 또한 자율주행자동차의 개발이 한창인 시점에

자율주행자동차의 시운전중 발생하는 사고의 문제점에 대해서 빠르고 정확한 원인 분석이 가능하다. 추가적으로 운전자의 주행습관에 대한 정보를 빅 데이터 형식으로 얻을 수 있다. 아래의 그림은 이벤트 데이터 디코더의 예시이다.

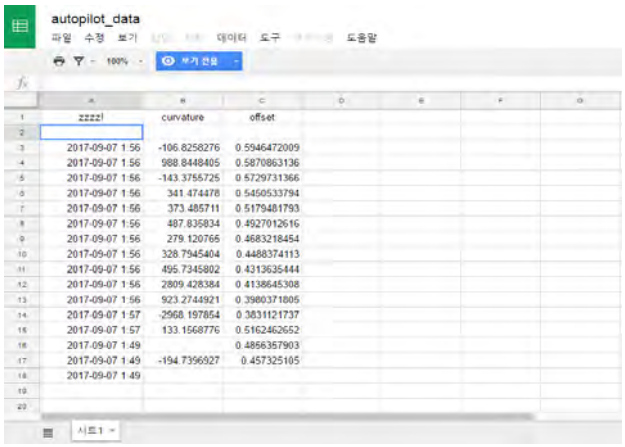

그림4. 이벤트 데이터 디코더(EDR) 의 예시

5. 주행모드

주행모드는 사용자가 선택할 수 있도록 총 3가지의 Mode를 설계하였다. Mode는 설계과정에서 노트북에 선택창이 나타나면 사용자가 원하는 Mode를 클릭하여 변경하도록 하였다. Mode는 총 3가지로 구성되어 있으며, Manual Mode, Semi-Auto Mode, Auto Mode로 구성되어 있다. Mode에 대한 설명은 아래의 내용이 뒷받침 한다. 모드변환을 위하여 사용된 방식은 받은 명령을 tcpCliSock으로 바로 보내는 방식과, tcpCliSock으로 받은 명령을 Socket.py로 이동하여 데이터를 txt파일에 Write모드로 입력한 후 그 txt파일을 Server에서 읽도록 하는 방식을 사용하였다. 전자의 경우에는 사용자가 한번 입력할 때마다 한번 씩 명령이 들어가기에 일반적인 주행제어를 하는데 사용되었고, 후자의 경우에는 Auto Mode 나 Semi Auto Mode로 주행모드를 변경하였을 때, 제어 부분에 무한루프가 생기는데, 이 무한루프를 나와서 주행모드를 변경하도록 외부와의 통신을 연결해주는 부분에서 사용되었다.

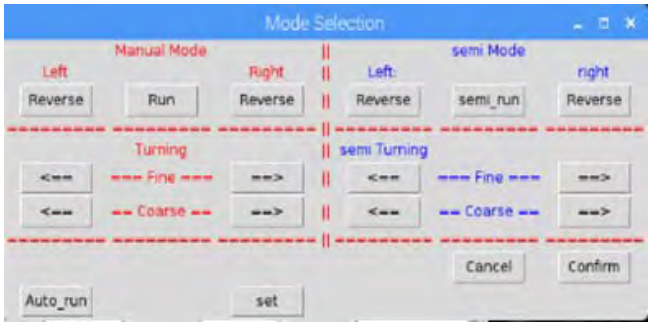

그림5-1. Mode 선택 화면

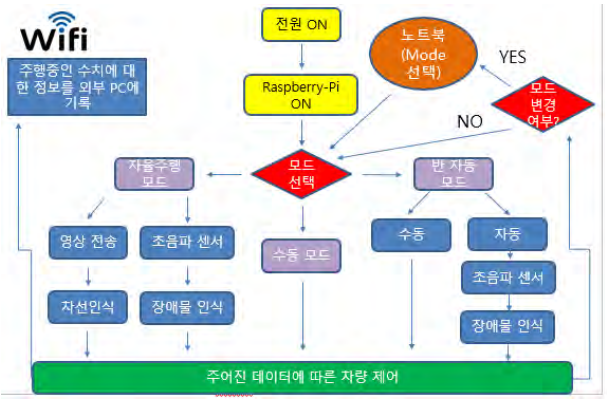

그림5-2. Mode선택에 따른 제어 흐름도

5.1 수동주행모드

노트북과 라즈베리파이(Raspberry Pi)를 VNC(Virtual Network Computing)로 연결하여 노트북에서 원격으로 수동주행을 가능하게 하였다. 위의 **그림 5-1**을 보면 왼쪽에 수동주행모드를 설계하여 사용자가 주행 제어에 필요한 offset변경 버튼(방향 전환)과 가속과 정지버튼(속도 제어)에 해당하는 Run 버튼을 만들었다. 수동주행모드가 진행되는 동안에는 다른 센서나 영상처리를 통한 자동제어를 받지 않고 오직 사용자의 조작에 의해서만 차량이 제어되도록 설계하였다. 모드를 변경하고 싶을 경우는 다른 모드의 버튼을 누른 후 set버튼을 두 번 누르게 되면 모드가 교체 되면서 다른 모드로 제어가 변경된다. 이 모드를 제어하기 위해 쓰인 방식은 tcpCliSock으로 바로 명령을 받는 방식이다.

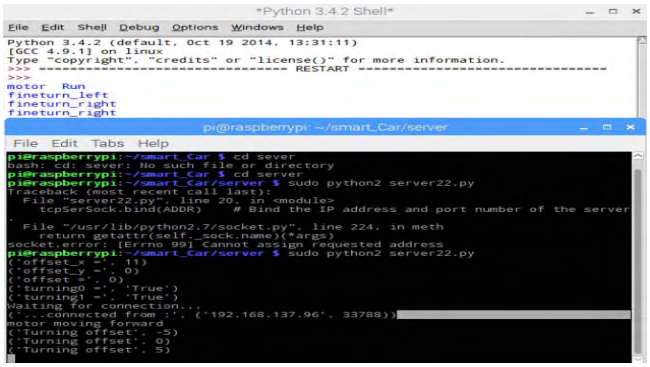

그림5-3. 수동주행모드

5.2 반자동 주행모드

사용자가 반자동 주행모드를 선택하게 되면 사용자의 직접적인 제어와 센서를 통한 자동제어가 융합된 주행이 이루어지도록 설계하였다. 기본적인 주행은 사용자가 제어하지만, 만약 주행도중 전방에 추돌위험이 있는 장애물이 초음파 센서를 통해서 감지될 경우, 자동차는 사용자의 제어를 무시하고 우선적으로 자동감속을 하게 된다. 주행모드 변경방법은 위에서 나왔던 방식과 동일하다. 이 모드를 제어하기 tcpCliSock을 바로 명령어로 받는 방식과 그

명령어를 Socket.py에 거쳐서 받는 방식 두 개가 전부 사용되었다.

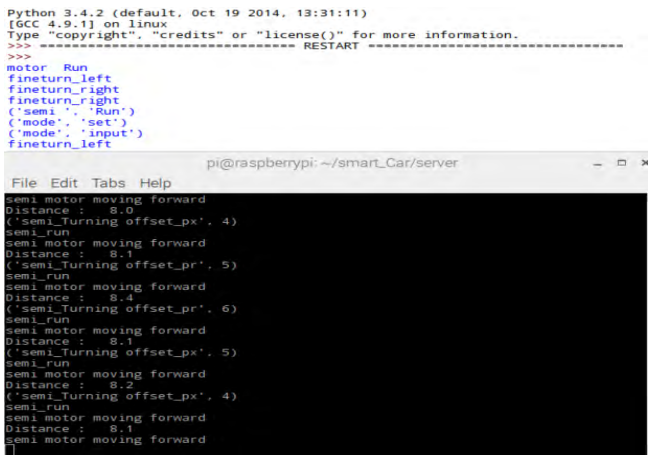

그림5-4. 반자동주행모드

5.3 자율주행모드

자동차는 전원 인가 후 기본적으로는 수동주행모드이다. 위의 **그림5-1**의 창에서 'Auto' 버튼을 누르면 자율주행 모드로 변화되게 된다. 자율주행모드가 선택이 되면 사용자는 자동차에 직접적인 제어를 하지 않고 오직 Open CV 영상처리를 통한 차선과 표지판인식, 그리고 초음파 센서를 통한 전방 장애물 인식을 통해 속도와 방향을 자동제어하게 된다. 주행모드 변경방법은 위에서 나왔던 방식과 동일하다. 이 모드를 제어하기 tcpCliSock을 바로 명령어로 받는 방식과 그 명령어를 Socket.py에 거쳐서 받는 방식 두 개가 전부 사용되었다.

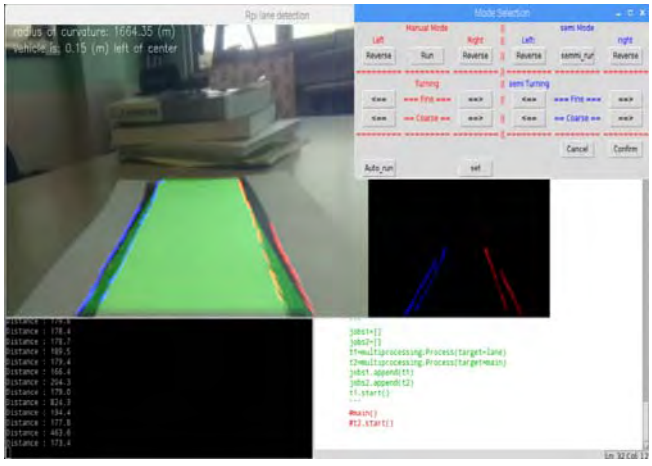

그림5-4 자율주행모드

6. 결론

본 연구에서는 자율주행기술을 구현하고 차량의 중요 동작 정보를 외부(Google Sheet)에 저장하였다.

그러나 본 연구에서는 다음과 같은 한계를 지닌다.

첫 번째, 라즈베리파이의 성능의 제한으로 영상처리, 구글 시트(Google Sheet)로의 정보 전송의 속도에 제약이 있었다.

두 번째, Raspberry Pi에 초음파 센서 HC-SR04를 사용하여 연구를 한 결과, 값이 잘 나오다가 가끔씩 값이 튀는 결과를 확인 할 수 있었다. 이것은, 실제 차량의 센서가 아닌 시중의 저렴한 MCU관련 초음파센서를 이용하여 연구를 하다 보니 그러한 측정값의 오차가 발생하는 제약이 나타났다.

세 번째, Wi-fi를 통한 차량 제어와 주행정보 전송이므로 Raspberry Pi의 Wi-fi의 연결이 끊기게 되면 송신이 중단되어 제어가 불가능해진다.

네 번째, 이 모든 데이터를 Raspberry Pi로 관리 제어를 하기에는 시스템이 무거워서 Raspberry Pi의 사양으로는 버거워 처리속도가 느려지는 한계를 지닌다.

하지만 위와 같은 문제들은 대부분 기술적인 한계와 맞닿은 문제이며, 해결하기 어려운 문제는 아니다. 그렇기 때문에 이 연구에 대한 이론을 가지고 다른 실생활 분야에 참고하여 적용할 경우, 자율주행 자동차의 개발과정에서 문제점과 그에 대한 해결책을 빠르고 정확하게 제공해 주는 기능을 해줄 수 있다. 또한, 일반적인 차량주행 중 차량이 심각하게 파손되어 블랙박스의 데이터 복구가 어려운 경우에도 도움을 줄 수 있을 것으로 보인다. 추가적으로 운전자의 주행습관이나 운전취향을 분석 하여 자동차 회사들이 빅 데이터 정보를 사용한다면 소비자의 기호에 맞게 차량 시스템 설계를 하는데 활용할 수 있을 것이라 예상된다. 마지막으로 Mode selection을 통하여 사용자의 기호를 고려함으로써, 직접적으로 운전하는 것을 즐겨서 자율주행 자동차에 대해 부정적인 시각을 가지고 있는 자동차 시장의 소비자 의견 또한 해결해 줄 것으로 예상된다.

참고문헌

[1] 자율주행자동차의 사고 사례
https://www.youtube.com/watch?v=mihdko-VD3s
[2] Open CV를 통한 차선 인식
https://github.com/uvbakutan/lane-detection-raspberry-pi
[3] Open CV를 통한 신호등 인식
https://github.com/kanishkaganguly/TrafficLightDetection
[4] 거리 감지를 통한 초음파 센서
https://www.raspberrypi-spy.co.uk/2013/01/ultrasonic-distance-measurement-using-python-part-2/