라즈베리파이를 이용한 자율주행 자동차

조우리*, 안주현**, 이다영**, 유연주**
*동덕여자대학교 컴퓨터학과
** 동덕여자대학교 컴퓨터학과
e-mail : woori.helena.cho@gmail.com*

Automatic Moving Vehicle using by Raspberry Pi

Woo-Ri Cho*, Joo-Hyeon Ahn**, Da-Young Lee**, Yeon-Ju Yu**
*Dept. of Computer Science, DongDuk Women's University
** Dept. of Computer Science, DongDuk Women's University

요 약

본 연구의 목적은 차선 인식 및 장애물 인식을 통한 자율 주행 자동차를 개발하기 위한 것이다. 일반적인 자동차는 사용자가 직접 운전을 통해 자동차를 제어해야 한다. 이는 오로지 사용자의 판단 하에 주행하는 것으로 사용자가 몸이 불편한 경우 자동차 주행에 어려움을 겪을 수 있다. 따라서 본 논문에서는 OpenCV 오픈 소스 라이브러리를 중심으로 차선 인식 및 장애물 인식 기능을 탑재한 자동차로 PWM 을 이용한 자동차의 앞 바퀴 회전 및 뒷바퀴 속도 조절이 가능하도록 하였다. 그 결과, 차선 인식과 장애물 인식을 통해 사람이 직접 운전하지 않는 자율 주행 자동차를 개발하였다.

1. 서론

현재 IT 시장은 영상 처리를 이용한 여러 제품들을 개발하기 위해 많은 노력을 하고 있다. 2001 년부터 시장에서는 이미 실시간 영상 처리 시스템이 각광받고 있으며 더 많은 분야에 있어 연구가 진행 될 예정이다.[1]

기존의 자동차 시스템은 사용자가 직접 운전을 통해 자동차의 주행이 진행되고, 사용자의 판단 하에 장애물 및 신호등에 대한 주행, 대기가 이루어진다. 이는 곧 사용자의 몸이 불편한 경우, 자동차의 주행 및 제어가 힘들다는 것을 의미한다. 이에, 더 편리한 주행을 위해 실시간 영상 처리를 이용한 자율 주행 자동차가 연구되기 시작하였다. 최근 전세계의 정보 기술 업체에서 자율주행 시스템을 적용하는 자동차가 연구개발 되고 있다. 점점 더 많은 기업에서 자율주행 자동차 개발을 시도하고 그 수준은 향상되고 있다. 또한 자율주행 자동차의 실용화를 위하여 자율주행 자동차의 도로 규정이 논의되고 있다. 따라서 자율주행 자동차 개발의 중요성은 더욱 커졌으며, 도로 위에서의 자율주행 시스템의 실현 가능성은 전보다 더욱 높아졌다고 볼 수 있다. 자율 주행 자동차가 실생활에서 적용되는 때가 점점 가까워지는 것이다.

따라서, 본 논문에서 연구된 차선 인식 및 장애물 인식을 통한 자율 주행 자동차는 실제 자율주행 자동차의 기능과 유사하게 OpenCV 를 이용한 차선 인식 및 장애물 인식을 통해 사용자가 직접 제어를 하지 않고 자동적으로 제어되게 된다.

따라서 본 연구는 자동차에 탑재된 카메라가 차선과 장애물을 인식하면, 차선을 통해 자율적으로 주행이 되며, 장애물을 인식하면 일정 시간 동안 주행 대기하는 자동차를 만들고자 한다.

논문의 구성은 다음과 같다. 2 장에서는 연구에 사용된 기술인 OpenCV 와 PWM 에 대해 설명한다. 3 장에서는 구현한 기술들에 대한 구성도가 설명되어 있다. 4 장에서는 실제 개발에 대한 서술이 자세히 되어 있다. 마지막 장에서는 결론과 향후 방향이 나와 있다.

2. 사용된 기술

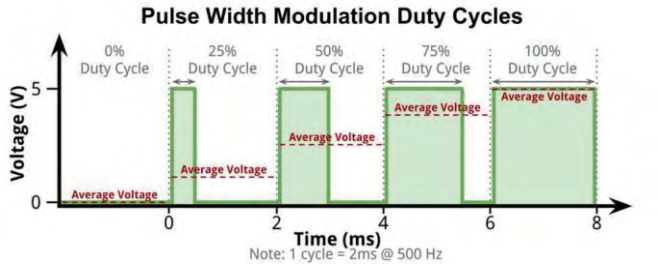
2.1 OpenCV 를 이용한 차선 인식 및 장애물 인식

영상처리(image processing)는 입력된 영상을 어떤 목적을 위해 처리하여 새로운 영상을 얻는 기술이다. 기본적으로 영상은 위치 값과 밝기 값을 가진 일정한 수의 화소들의 모임으로 정의된다. 영상은 f(x, y)로 여기서 (x, y)는 위치를 가리키는 좌표 값이다. 그리고 f( )는 해당 위치의 밝기 값이 된다. 즉, 2 차원 평면 x=100, y=50 위치에 밝기 값이 128 인 화소가 있다면 f(100, 50) = 128 이 되는 것이다.[2] OpenCV (Open Source Computer Vision Library)는 영상처리, 기계학습 및 컴퓨터 비전 기능을 갖는 소스가 공개된 라이브러리로[3] 컴퓨터 영상처리 프로그래밍을 위한 기능 패키지이다. 행동· 물체· 얼굴 인식 등의 응용프로그램에서 사용된다.

2.2 PWM

PWM 은 Pulse Width Modulation(펄스 폭 변조)의 약자로, 아날로그 신호를 디지털 형태로 나타낸 것이다.

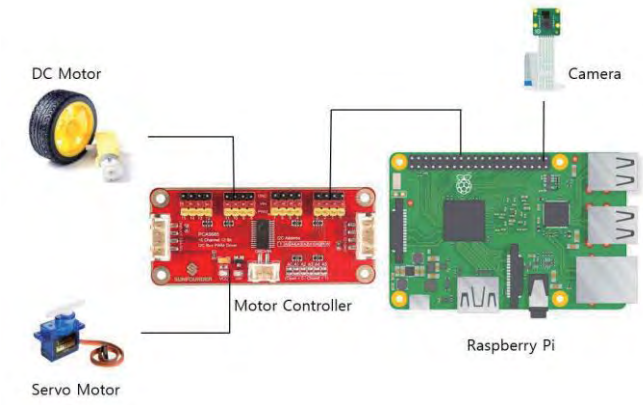
디지털 신호에 대해서 주파수를 설정하고, pulse width(duty cycle)는 아날로그 신호의 진폭에 따라 변한다. 아날로그 신호의 진폭과 디지털 신호의 pulse width(duty cycle)의 관계는 어플리케이션에 따라 달라진다. PWM 신호는 다방면에서 사용이 되며 특히 DC 모터 제어에 많이 쓰인다.



(그림 1) PWM 의 개념

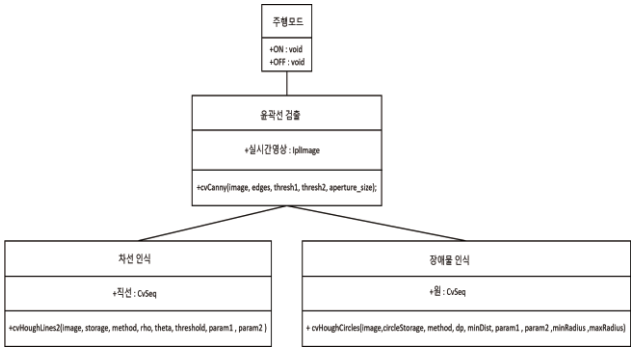
그림 1 을 보면 Duty Cycle 이 증가함에 따라 전압도 커지는 것을 알 수 있다.

3. 시스템 구성도



(그림 2) 하드웨어 구성도

그림 2 는 자율주행 자동차의 하드웨어 구성도이다. 라즈베리파이에서 모터 파일을 실행하면, Motor Controller 를 통하여 뒷바퀴를 움직이게 하는 DC Motor 가 구동되고, 앞바퀴의 방향을 담당하는 Servo Motor 가 방향을 제어하게 된다. 라즈베리파이에 연결되어 있는 Camera 는 시야에 있는 선을 도로로 인식하여 주행하고, 원 물체를 장애물로 인식하여 주행을 중지한다.



(그림 3) 소프트웨어 클래스 다이어그램

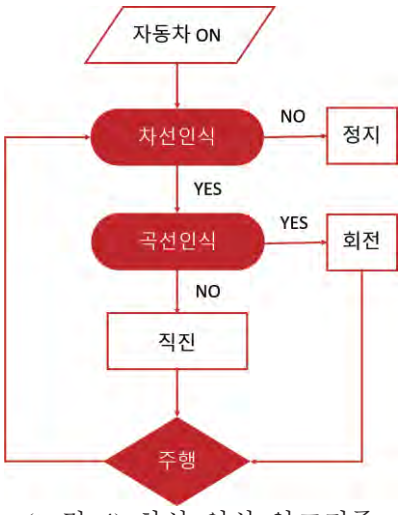
그림 3 은 라즈베리파이를 이용한 자율주행자동차의 소프트웨어적 작동과정을 클래스 다이어그램으로 작성한 것이다.

우선 윤곽선 검출 클래스를 거친 후 차선 인식, 장애물 인식 클래스로 나뉘어진다.

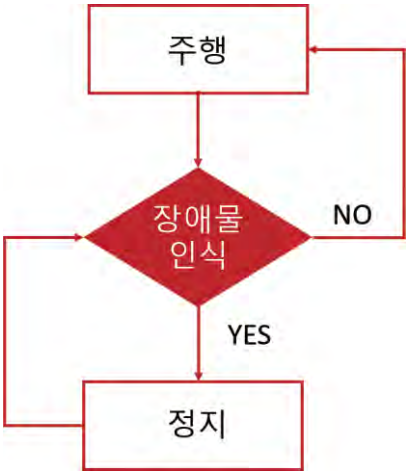
첫 번째로 윤곽선 검출은 파이카메라와 UV4L 을 통해 받은 실시간 영상 속에서 윤곽선을 검출하는 것이다. 컬러로 전송되는 실시간 영상을 흑백으로 바꾼 후 Canny Edge 를 구현하여 윤곽선이 검출되도록 한다. 이 영상을 토대로 차선 인식과 장애물 인식이 가능하게 된다.

두 번째로 차선 인식은 윤곽선이 검출된 실시간 영상 속에서 직선을 검출하여 라즈베리파이 자동차가 스스로 주행하도록 하는 것이다. HoughLine2 를 구현하여 직선이 검출되도록 하며 검출된 직선의 시작지점과 종료지점의 좌표 값을 파일로 실시간 저장하게 한다. 이 값에 따라 라즈베리파이 자동차가 주행하고 바퀴의 방향을 제어하게 된다.

마지막으로 장애물 인식은 윤곽선이 검출된 실시간 영상에서 원을 인식하는 것이다. 원 인식은 HoughCircle 을 통해 구현된다. 라즈베리파이 자동차가 주행을 하다가 원을 인식하였을 경우 지정된 값을 파일로 실시간 저장하여 모터를 정지할 수 있도록 한다



(그림 4) 차선 인식 알고리즘



(그림 5) 장애물 인식 알고리즘

그림 4 와 그림 5 는 라즈베리파이를 이용한 자율주행자동차의 가장 큰 기능인 차선인식과 장애물 인식 알고리즘을 순서도로 구현한 것이다.

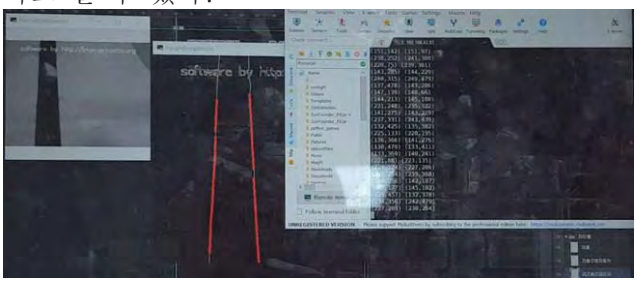
4. 개발

4.1 OpenCV 의 활용

OpenCV(Open Source Computer Vision)는 실시간 영상처리를 목적으로 한 프로그래밍 라이브러리이다.[4] 본 연구에서는 OpenCV 라이브러리를 이용하여 차선인식 및 장애물 인식 기능을 개발하였다.

먼저, 카메라가 차선을 인식하게 하기 위해서는, 카메라 안에 인식되는 모든 사물의 윤곽선을 추출해야 한다. 윤곽선을 추출하기 위해서는, 카메라가 실시간으로 받아오는 영상을 흑백으로 처리하는 작업이 우선되어야 한다. 그 후, 차선으로 인식되는 직선의 값만 추출하여 저장한다. 장애물은 원으로 설정하여, 카메라의 원 인식 여부에 따라 서로 다른 값을 저장하도록 하였다.

차선이 카메라 안에 인식되면, 차선의 양 끝점인 (x1,y1), (x2,y2) 좌표를 받아오게 된다. 이 좌표의 값들을 텍스트 파일로 저장한다. 차선이 인식되는 동안 계속 변화하는 좌표 값들을 실시간으로 저장한다. 이 저장된 값들을 따라 자동차가 움직이게 차선을 따라 간다고 할 수 있다.



(그림 6) 차선 인식 개발 실험

그림 6 은 카메라가 차선을 인식한 사진이다. 차선을 인식하면 그 직선의 양 끝점의 좌표를 표시한다. 실시간으로 저장되는 좌표가 자동차의 움직임을 결정하게 된다.

(그림 7) 장애물 인식 개발 실험

그림 7 은 카메라가 장애물로 설정한 원을 인식한 사진이다. 원을 인식하면 원의 좌표를 표시해주고 2 를 넘겨주게 된다. 2 가 넘어갈 경우, 자동차의 움직임은 멈추게 된다.

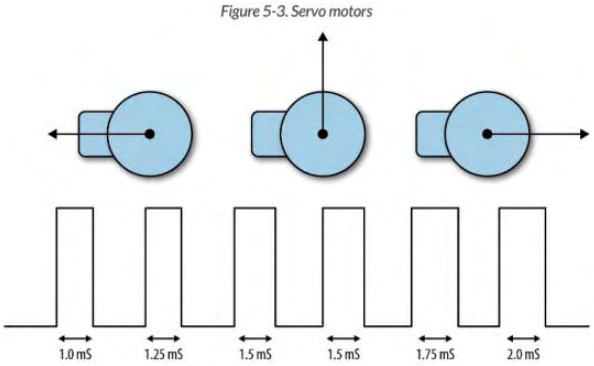
4.2 PWM 의 활용

이 연구에서 PWM 을 서보 모터의 방향 제어 및 DC 모터 제어하는데 사용했다. Python 을 이용해 프로그래밍 했을 때 핵심 코드는 다음과 같다.

```
pwm = Adafruit_PCA9685.PCA9685()    (1)
pwm.set_pwm_freq()                  (2)
pwm.set_pwm()                       (3)
```

(1)번 코드는 PWM 을 Adafruit_PCA9685 모터 드라이버를 통해 사용하겠다는 것을 의미한다.
모터 드라이버에 설정된 PWM 번호를 이용해 전력을 주고, 이를 통해 서보 모터 및 DC 모터를 제어하도록 한다.
(2)번 코드는 PWM 의 진폭을 설정해주며, 설정한 값에 따라 서보 모터의 각도 값 및 DC 모터의 속도에 차이가 생긴다. 이를 통해 자동차의 앞 바퀴 회전 양 및

(3)번 코드는 개발자가 조정하고자 하는 PWM 값 및 DC 모터의 속도 값을 지정할 수 있다.



(그림 8) Duty Cycle

그림 8 은 PWM 을 이용해 서보 모터의 각도 값을 제어하는 원리를 이용하였다. 기본적으로 서보 모터는 0 도에서 180 도까지의 양방향으로 회전시킬 수 있다.

여기서 본 연구는 Adafruit_PCA9685 모터 드라이버를 이용하였기 때문에 PWM 의 값은 모터 드라이버를 통해 전송이 된다. 이때 중심 값, 즉 서보 모터의 90 도 값을 알아낸다면, 이를 중심으로 90 도 보다 작은 각도, 90 도 보다 큰 각도를 각각 제어할 수 있다. 또한 DC 모터와 연결된 PWM 값을 알아낸다면, DC 모터 제어 또한 가능해진다.

5. 결론

라즈베리파이를 이용한 자동차에 카메라를 탑재하여 차선을 인식하도록 하면, 자동차의 자율주행이 가능하게 된다. 또한, 장애물 인식도 가능하여 주행 중 일어날 수 있는 사고를 예방할 수 있다.

그 밖에도 자율주행자동차의 장점은 많다. 사람이 직접 운전을 하지 않아도 자동차가 차선을 따라 가게 되므로 도로 위에서 졸음으로 인한 사고 또한 방지할 수 있다. 그리고 자율주행 기술은 신체적인 장애와 같은 이유로 운전을 할 수 없었던 사람들도 운전을 할 수 있게 해준다는 장점을 가진다.

현재 자율주행자동차는 가장 주목받는 기술 중 하나이다. 실제로 구글에서는 자율주행자동차의 상용화를 목전에 두고 있고, 현재 테스트 단계에 있다. 기본적인 차선인식과 장애물인식 기술에 추가하여 신호등인식, 갑자기 튀어나온 장애물 인식, 앞서가던 자동차의 급정거 여부 등 도로 위에서의 여러 상황들을 고려한 수많은 기술들이 적용된다면, 머지않아 자율주행자동차의 상용화를 기대할 수 있을 것이다. 자율주행자동차의 상용화가 가져올 많은 변화를 기대해보자.

참고문헌

[1]Xianghua Fan, Fuyou Zhang, Haixia Wang, Xiao Lu “The System of Face Detection Based on OpenCV” IEEE Comm 2012

[2] OpenCV로배우는영상처리및응용.
정성환,배종욱지음. 생능출판. 2017년3월3일초판발행

[3] OpenCV programming. 김동근 지음. 가메출판사. 2011년 7월 22일행

[4] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek “A brief introduction to OpenCV” MIPRO 2012

출처

그림 1
https://www.instructables.com/id/LinkIt-One-and-PWM-Pulse-Width-Modulation/

그림 8
http://razzpisampler.oreilly.com/ch05.html