

3D 자동차 시뮬레이터 기반 상호작용형 ADAS 개발 및 검증 프레임워크

Interactive ADAS development and verification framework based on 3D car simulator

조 든 솔*, 정 세 열*, 김 형 수*, 이 승 기*, 김 원 태*

Deun-Sol Cho*, Sei-Youl Jung*, Hyeong-Su Kim*, Seung-gi Lee*, Won-Tae Kim*

Abstract

The autonomous vehicle is based on an advanced driver assistance system (ADAS) consisting of a sensor that collects information about the surrounding environment and a control module that determines the measured data. As interest in autonomous navigation technology grows recently, an easy development framework for ADAS beginners and learners is needed. However, existing development and verification methods are based on high performance vehicle simulator, which has drawbacks such as complexity of verification method and high cost. Also, most of the schemes do not provide the sensing data required by the ADAS directly from the simulator, which limits verification reliability. In this paper, we present an interactive ADAS development and verification framework using a 3D vehicle simulator that overcomes the problems of existing methods. ADAS with image recognition based artificial intelligence was implemented as a virtual sensor in a 3D car simulator, and autonomous driving verification was performed in real scenarios.

요 약

자율 주행 차량은 주변 환경의 정보를 수집하는 센서, 측정된 데이터를 판단하는 제어 모듈로 구성된 첨단 운전자 지원 시스템(ADAS)을 기반으로 있다. 최근에 자율주행 기술에 대한 관심이 증가함에 따라 ADAS 입문 개발자들 및 학습자들을 위한 손쉬운 개발프레임워크가 필요하다. 그러나, 기존 개발 및 검증 방식은 고성능 자동차 시뮬레이터를 기반하기 때문에 검증 방법의 복잡성 및 고비용 등의 단점이 있다. 또한, 대부분의 방식은 시뮬레이터로부터 ADAS에서 필요로 하는 센싱 데이터를 직접 제공하지 않으므로 검증 신뢰성의 한계가 있다. 본 논문에서는 기존 방식들의 문제점들을 극복하는 3D 자동차 시뮬레이터를 활용한 상호작용형 ADAS 개발 및 검증 프레임워크를 제시한다. 영상인지 기반의 인공지능을 적용한 ADAS를 3D 자동차 시뮬레이터에서의 가상센서로 구현하고, 실제 시나리오에 자율주행 검증을 진행하였다.

Key words : Autonomous vehicle, Vehicle Simulator, ADAS Development, ADAS Verification, Artificial Intelligence

* Dept. of Computer Science and Engineering, Koreatech University

★ Corresponding author

E-mail : wtkim@koreatech.ac.kr, Tel : +82-41-560-1485

※ Acknowledgment

This work was partly supported by the Institute for Information & communications Technology Promotion (NO. 2015-0-00816-004), and by the National Research Foundation of Korea (No. 2017010875) from the Korea government (MSIP). Manuscript received Dec. 5, 2018; revised Dec. 13, 2018; accepted Dec. 13, 2018

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

차량이 인간의 행동이나 제어 신호에 대해 의존적으로 움직이는 유인자동차와 달리 인간과 독립적으로 차량 스스로 주행을 판단하는 자율주행 자동차가 연구되고 있다. 자율주행자동차에 대한 연구는 2014년에 이르러 유럽, 미국, 일본 등 전 세계의 기업과 정부들이 주체 하에 참가하여 완전한 자율주행 기술을 개발 및 연구하고 있다[1]. 자율주행 자동차의 최종 목표는 차량 운전 시 인간의 제어가 일체 없이 완전한 자율적인 주행을 지향하는 것이다[2]. 자율주행 자동차는 인간의 감각과 동일한 역할을 하는 운전 상황을 판단하고, 제어하는 기술들의 집합체인 ADAS(Advanced driver-assistance systems)가 요구된다[3]. ADAS의 기술은 전방의 차량 또는 보행자와의 거리를 조절하는 ASCC(Advanced Smart Cruise Control), AEB(Autonomous Emergency Brake)와 차선이탈에 대한 알람을 올려주는 LDW(Lane Departure Warning)과 차선을 유지하도록 주행을 도와주는 LKAS(Lane Keeping Assist System) 등 차량의 전방, 측면, 후방을 인지하고, 운전자의 안전한 운전을 안내하는 목표를 가지는 시스템이 포함된다[4].

실제 차량 주행 환경에서 자율 주행의 정확성을 높이기 위해서는 다양하고 복합적인 센서들을 포함하는 ADAS를 필요로 한다. 다양한 내부 모듈들이 서로 상호작용 한다는 것은 복잡도가 증가한다는 의미를 갖고 있다. 실제 환경에서 복합적인 센서들의 상호작용을 검증하는 것은 상당한 노력과 비용이 발생한다. 따라서 차량 시뮬레이터를 사용하는 것은 시스템을 개발함에 있어서 매우 효율적인 방법이다[5].

그러나, 차량 시뮬레이터는 대부분 시나리오를 구성하기 위한 유연성을 제공하지만 모듈, 시스템을 수정하지 못하기 때문에 항상 같은 검증 시나리오만을 거치게 된다. 이를 보장하기 위해, 본 논문에서는 여러 환경을 구현가능하며 자동차 모델 그리고 시뮬레이션의 무결성 데이터를 제공해주는 마이크로소프트의 오픈소스 기반 AirSim 시뮬레이터를 사용한다. AirSim 시뮬레이터로 ADAS를 개발함으로써 검증의 신뢰성, 고비용 그리고 고정되어 있는 검증 시나리오의 단점을 극복하고자 한다[6].

본 논문에서는 쉽고 효율적인 ADAS의 개발을

위해 3D 자동차 시뮬레이터를 활용한 상호작용형 개발 및 검증 프레임워크를 제안한다. 3D 자동차 시뮬레이터에서 ADAS로서 필요한 센싱 데이터를 수집하며, 해당 데이터를 통해 ADAS를 구현하였다. ADAS의 LKAS를 구현하고자 라인을 인지하기 위해 OpenCV를 사용하여 영상 처리 모듈을 만들었다. 또한, AEB를 구현하기 위해 CNN(Convolution Neural Network)기반의 YOLO(You Only Look Once) 프레임워크를 활용하여 객체 탐지 모듈을 구현하였다[7]. 마지막으로 설계한 ADAS를 장착한 차량을 시뮬레이터 내에서 시나리오에 맞게 검증을 진행하였다.

본 논문은 2장에서 자율주행 자동차의 모듈의 유효성 테스트를 위한 오픈 소스 시뮬레이터, 고성능 시뮬레이터, 재구성 가능한 시뮬레이터 기반의 ADAS 개발 및 검증 시스템에 대한 연구를 소개한다. 3장에서는 제안하는 방법론으로 구현한 ADAS 개발 및 검증 프레임워크의 전체 구성도와 각 구성 요소에 대한 상세기술, 동작방식, 그리고 데이터 흐름을 설명하고, 4장에서는 개발한 ADAS의 유효성 검증에 대한 비교의 실험 분석 결과를 보여준다. 5장에서는 마지막으로 본 논문의 결론을 맺음으로 구성된다.

II. 관련 연구

2.1 오픈 소스 시뮬레이터 기반 ADAS 개발 및 검증 시스템

Choi는 ADAS의 LKAS를 오픈소스 기반의 자동차 시뮬레이터 TORCS(The Open Racing Car Simulator)를 이용하여 개발한 자율주행 시스템을 제안하였다[8]. TORCS에서 주행 차량, 주행환경 그리고 차량의 전방 영상에 대한 정보를 실시간으로 얻어 CNN 모델에 학습시켰다. 완성된 CNN 모델의 판단의 출력인 스티어링 휠 명령은 TORCS로 전달되어, TORCS의 차량 모델의 스티어링 휠의 방향을 조절하였다. 이 시스템은 차선만 인지하고 스티어링 휠의 출력 값만 결정하기 때문에 다른 복합적인 ADAS 검증이 불가능한 단점이 있다.

Park은 오픈소스 기반 교육용 주행 알고리즘 경쟁 플랫폼을 제안하였다. TORCS와 유저가 구현한 주행 알고리즘을 구현한 내장 ECU의 통신을 통해, 상호 간 경쟁을 통한 성능 검증을 할 수 있도록 하

었다. 본 논문과 유사하게 상호작용으로 기어, 차량 속도, 전방이미지 등의 주변 환경의 센싱 정보를 수집하고, 브레이크, 엑셀 등과 같은 차량 액션을 수행하도록 설계하였다. 다른 주행 알고리즘과 경쟁하는 플랫폼이므로, 차량의 다수인 환경에서의 구현할 수 있다는 장점이 있으나, 표준으로 되어있는 ADAS의 검증 방법에 대한 신뢰성이 미비하다[9].

2.2 고비용 고성능 시뮬레이터 기반 ADAS 개발 및 검증 시스템

ADAS의 기능 중 하나인 ASCC는 속도와 거리를 추종하는 성능에 초점이 맞추어져 있다. Park은 해당 기능의 단점을 해결하기 위해 기존의 시스템에서 연비를 향상시키는 ASCC 주행 제어 알고리즘을 제안하였다. 해당 알고리즘은 Matlab의 Simulink 프로그램과 고비용인 고성능 시뮬레이터 CarSim을 이용하여 검증이 진행되었다. 그러나, 해당 시뮬레이터를 ADAS 입문 개발자들과 교육자들이 사용하기에는 매우 고비용이기 때문에 비용 측면에서 효율이 낮다[10].

2.3 재구성 가능한 시뮬레이터

일반적으로 시뮬레이터는 모듈, 시스템을 추가, 삭제 그리고 변경하지 못하도록 고정되어 있다. Hassan은 이를 해결하기 위해 재구성 가능한 시뮬레이터를 정의하고 시스템의 주요 개념, 구성 요소 간의 인터페이스와 관계와 전체적인 시스템 구조를 설명하였다[11].

III. ADAS 개발 및 검증 프레임워크

3.1 전체 시스템 구성도

본 논문에서 제안하는 ADAS의 개발 및 검증 프레임워크(ADAS-DVS : ADAS-Development & Verification System)의 구성도는 그림 1과 같다. ADAS-DVS와 P-ADAS(Prototype ADAS)는 차량의 센싱 데이터와 제어 메시지를 서로 교환한다.

3.2 ADAS-DVS

ADAS-DVS는 가상 차량의 센싱된 데이터 송신과 ADAS의 차량 제어 데이터 수신 역할을 하는 ADAS 플러그인 인터페이스(ADAS plug-in interface) 모듈과 가상환경에서의 ADAS 검증을 위한 환경

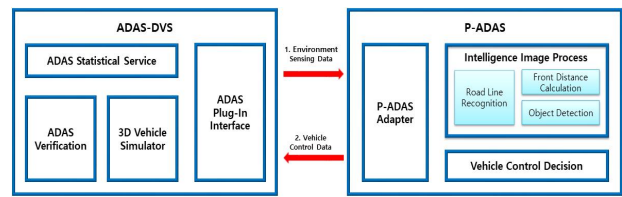


Fig. 1. The proposed ADAS development and verification framework.

그림 1. 제안하는 ADAS 개발 및 검증 프레임워크

을 제작하는 3D 차량 시뮬레이터(3D vehicle simulator) 모듈, ADAS의 유효성을 테스트하는 ADAS 검증(ADAS Verification) 모듈, 그리고 ADAS의 수집한 데이터를 통계로 나타내는 ADAS 통계 서비스(ADAS Statistical Service)모듈로 구성되어 있다.

가. ADAS Plug-in Interface

ADAS 플러그인 인터페이스는 ADAS와 연결하기 위해 인터페이스 역할을 맡은 모듈이다. ADAS 플러그인 인터페이스는 차량 시뮬레이터에서 나오는 카메라의 영상 이미지, 속도, 기어, 충돌 발생 여부의 데이터를 ADAS로 송신하며, ADAS의 차량 제어 값을 수신한다. 수신한 제어 값은 3D 차량 환경 시뮬레이터로 전달한다.

나. 3D Vehicle Simulator

3D 차량 시뮬레이터는 ADAS에서 계산된 결과를 반영한 최적의 차량 제어의 결과를 시뮬레이션 하며, ADAS의 성능 평가를 담당한다. 제안하는 ADAS 기능을 가진 가상의 차량 모델과 주행 중인 차량, 보행자, 신호등, 차선 등의 객체를 시각화하는 3D 시각화 모델을 바탕으로 가상의 운전 환경 테스트를 수행한다. 테스트 결과로 나오는 카메라 영상, 속도 등이 ADAS의 입력으로 받아드리도록 메시지를 만든다.

다. ADAS Verification

LKAS와 AEB에 대한 검증은 AirSim의 도시 환경에서 차량모델에 있는 직접 구현한 ADAS가 정상적으로 작동하고 있는지, 얼마나 높은 정확도를 가지는지에 대해 평가한다.

라. ADAS Statistical Service

ADAS 통계 서비스는 해당 모델이 전체적으로 운행한 시간과 중간에 기능이 끊긴 시간을 타임스

템플을 생성하여 체크한다. 차량의 ADAS가 얼마나 어떤 기능을 작동하였는지, 정확도가 얼마나 되는지에 대해 시각적으로 보기 쉽게 구현하였다.

3.3 Prototype ADAS (P-ADAS)

P-ADAS는 ADAS-DVS에서 ADAS의 검증을 위해 ADAS-DVS를 기반으로 구현된 ADAS의 Prototype이다. 해당 P-ADAS는 자율주행 차량을 위한 차량 영상 처리를 담당하는 지능적 이미지 처리(Intelligence image process) 모듈, 자율적인 차량 제어를 결정하는 차량 제어 결정(Vehicle control decision) 모듈, 그리고 ADAS-DVS와 연결하여 데이터를 교환하기 위해 인터페이스 역할을 하는 P-ADAS 어댑터(P-ADAS Adapter) 모듈이 있다. 다른 ADAS를 ADAS-DVS 프레임워크를 사용하여 검증하고자 하면, 제공하는 인터페이스를 준수하여 데이터의 in/output을 제공하면 해당 ADAS의 기능 및 성능을 검증할 수 있도록 설계하였다.

가. Intelligence Image Process

지능적 이미지 처리는 ADAS-DVS에서 차량모델의 데이터 중 카메라의 영상 데이터를 가공하는 모듈이다. 라인을 유지하는 LKAS의 기술과 보행자, 장애물, 신호등 등을 검출하고, 장애물로 인해 사고의 위험이 있을 때 주행을 중지하는 AEB 기술로 구성되어 있다. LKAS의 유효성을 위해 영상 이미지 처리로 라인 추출하는 기술이 요구되며, AEB는 보행자나 신호등 등의 추출을 위한 인공지능 기술과 차량 간 거리를 자동으로 조절해주는 계산 공식이 필수적이다[12].

나. Vehicle Control Decision

차량 제어 결정은 ADAS에서 처리된 결과를 활용하여 Rule Base 기반으로 차량의 속도, 방향, 브레이크, 기어 등을 결정한다. 이에 대해 결정된 액션은 ADAS-DVS에서 3D 차량 시뮬레이터의 차량 모델에게 제어하는 메시지를 만든다.

다. P-ADAS Adapter

P-ADAS 어댑터는 P-ADAS에서 ADAS-DVS로의 연결을 지원하는 어댑터 역할을 맡은 모듈이다. P-ADAS 어댑터는 3D 차량 시뮬레이터에서 나오는 카메라의 영상, 속도, 기어, 충돌 발생의 데

이터를 수신하며, 차량 제어 결정 모듈에서 속도의 값에 따라 만들어진 스톱, 브레이크, 스티어링, 차량 모드를 포함하는 메시지를 ADAS-DVS로 송신한다.

3.4 동작 순서 및 데이터 흐름

그림 2는 ADAS-DVS와 P-ADAS의 데이터 흐름을 나타낸다. ADAS-DVS는 차량의 영상 데이터와 차량의 센서 및 상태 데이터를 P-ADAS로 전송한다. P-ADAS에서 해당 데이터로 차선 검출 및 객체를 판단하여, 차량 액션에 대한 적절한 판단을 내린다. 해당 판단 결과를 ADAS-DVS에서 3D 시뮬레이터 내부에 있는 차량 모델에 적용시키기 위해 ADAS-DVS로 차량 제어 데이터를 전송한다.

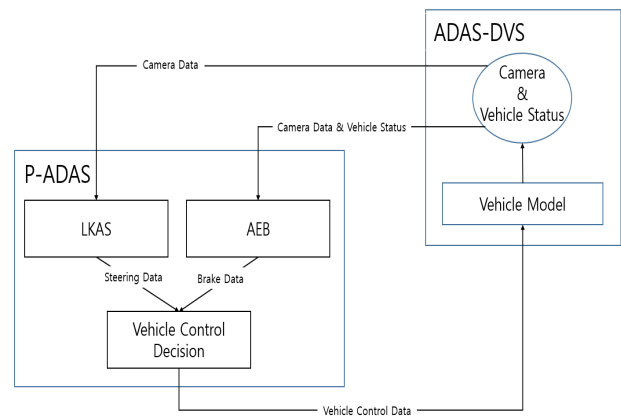


Fig. 2. The data flow diagram of ADAS development and validation framework.

그림 2. ADAS 개발 및 검증 프레임워크의 데이터 흐름도

IV. 실험 및 분석

ADAS의 개발 및 검증 프레임워크의 실제 구현을 위해 그림 3과 같이 설계하였다. 제안한 프레임워크에서 ADAS 검증 시스템은 마이크로소프트에서 오픈소스로 공개한 AirSim 시뮬레이터로 전후 좌우를 관찰 할 수 있는 4채널 카메라를 가진 차량 모델과 실제 도로, 차량, 보행자를 모델 구현하여, 검증 환경을 만들었다. ADAS는 LKAS와 AEB의 기술을 적용하였다. LKAS는 OpenCV의 라이브러리를 이용하여 차선 검출 알고리즘을 구현했으며, AEB의 경우는 YOLO 프레임워크로 차량, 보행자, 신호등을 판단하도록 구현하고, ADAS 기술들의 판단한 종합적인 결과를 바탕으로 차량 제어를 결

정하였다. 마지막으로 ADAS 검증 시스템과 ADAS 간 통신은 센싱 데이터와 차량 제어 데이터를 교환할 수 있도록 메시지를 RPC 프로토콜 기반으로 디자인 하였다.

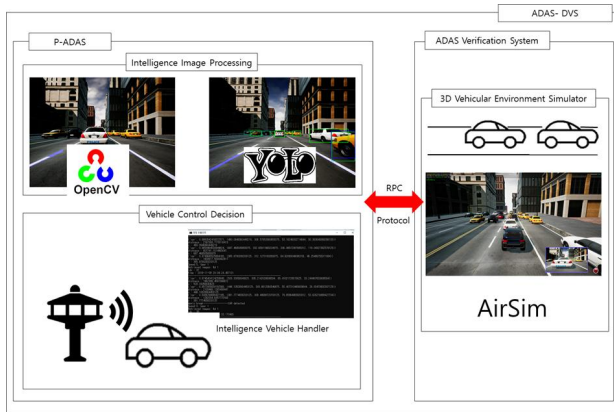


Fig. 3. ADAS development and verification configuration based on proposed framework.

그림 3. 제안 프레임워크 기반 ADAS 개발 및 검증 구성

가. 객체 인식률 검증 결과

ADAS의 객체 인식률을 검증하기 위해 YOLO framework로 학습한 CNN 모델을 사용하여 객체 검출을 Airsim에서 시행하였다. 학습은 자동차 2980개, 사람 1682개, 신호등 1239개의 데이터를 학습시켰고 Epoch은 64이며 Loss는 0.4483으로 측정이 됨으로써 CNN모델이 잘 학습되었다는 것을 확인할 수 있다.

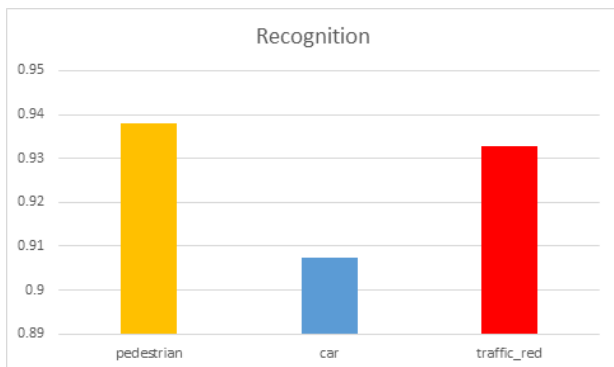


Fig. 4. Object recognition rate verification result.

그림 4. 객체 인식률 검증 결과

그림 4와 같이 해당 CNN 모델을 사용하여 시뮬레이터 내에서 객체를 인식하였을 때 보행자는 93.81%, 자동차는 90.73%, 신호등(빨간색)은 93.26%로 관찰되었다. 자동차가 다른 값에 비해 상대적으로

로 낮은 이유는 화면에서 자동차가 잘리는 상황이 있을 때 인식률이 떨어지기 때문이다.

나. AEB 검증 결과



Fig. 5. Object recognition in AEB.

그림 5. AEB에서의 객체 인식 사진

AEB의 기능으로서 그림 5와 같이 ADAS는 항상 객체를 인지하고 객체와의 거리를 체크하여 사고를 예방하고 만약 사고가 날 상황이라면 급정거를 판단하여 사고를 방지한다. 해당 기능을 검증하기 위해서 사고가 날 상황을 시나리오로 하여 사고를 얼마나 방어할 수 있었는지 확인하였다.

$$p_{defend} = \frac{t - t_{accident}}{t} * 100(\%) \quad (1)$$

t 는 사고가 일어날 수 있는 상황의 수 $t_{accident}$ 는 실제사고가 일어난 횟수, 그리고 p_{defend} 사고에 대한 방어율이다. AEB에 대한 검증으로써 해당 시나리오를 적용하여 20, 40, 60 km/h의 각각의 속도로 10회씩 실험을 하였을 때 AEB가 작동함으로써 사고 방어율이 100%가 나왔다. 따라서, 견고하고 안정적인 AEB를 구현하였다는 것을 검증할 수 있었다.

다. LKAS 검증 결과

$$p_{error} = \frac{t - t_{LKAS}}{t} * 100(\%) \quad (2)$$

t_{LKAS} 는 LKAS가 작동한 시간이고 p_{error} 는 에러율 그리고 t 는 자동차 전체 동작시간이며 단위는 초(s)이다. LKAS에 대한 검증으로써 AirSim의 도시 환경에서 직선구간을 약 40초, 13m/s이상의 속도로 4번 운전을 시행하였다.



Fig. 6. ROI of LKAS and lane recognition during driving.
그림 6. LKAS의 ROI 및 주행 중 차선 인식 사진

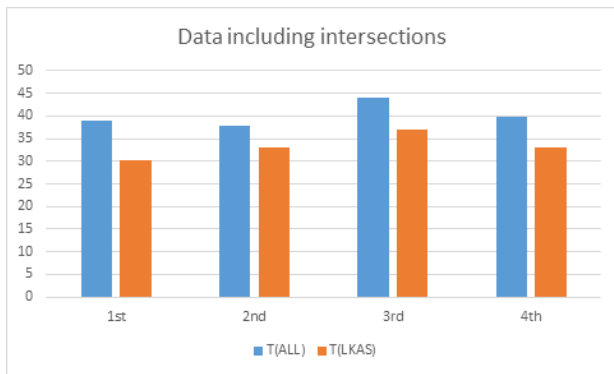


Fig. 7. Data including intersections(1st~4th).
그림 7. 교차로를 포함하는 데이터(1회~4회)

그림 7과 같이 교차로를 포함한 환경에서의 전체 t 값과 t_{LKAS} 이 1회(39/30.3), 2회(37.85/33) 3회(44/37.12), 4회(39.77/32.97)로 관찰되었다. 전체 t_{LKAS} 가 130.39초이며 전체 t 가 160.62초로 에러율이 16.95%가 나왔고 정확도는 83.04%로 측정되었다.

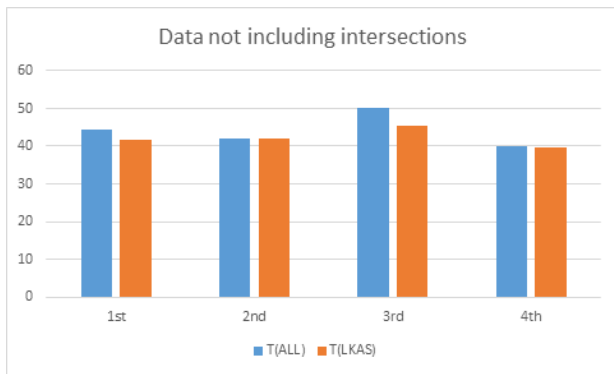


Fig. 8. Data not including intersections(1st~4th).
그림 8. 교차로를 포함하지 않는 데이터(1회~4회)

그림 8과 같이 교차로를 포함하지 않은 환경에서의 전체 t 값과 t_{LKAS} 이 1회(44.36/41.68), 2회(42/42) 3회(50.05/45.58), 4회(40.16/39.77)로 관찰되었다. 전체 t_{LKAS} 가 169.03초이며 전체 t 가 176.57초로 에러율이 4.27%가 나왔고 정확도는 95.73%로 측정되었다.

위에서 에러율이 크게 차이가 난 이유로는 교차로와 같은 차선을 인식하지 못하는 부분에서는 자동으로 LKAS가 꺼지도록 구현하였기 때문이다. 교차로에서 인식 되지 않은 결과를 기반으로 운행한다면 낮은 정확도로 인해 사고가 일어날 수 있다. 따라서, 교차로와 같이 인식되지 않는 구간에서는 LKAS가 자동으로 꺼져야만 자동차 사고를 방지할 수 있기 때문에 해당사항을 구현하게 되었다. 이와 같은 결과로 LKAS의 검증에 있어서 교차로를 제외한 상황에서 검증을 해야 한다는 것을 유추해 낼 수 있다.

V. 결론

ADAS를 개발하고 검증하는데 있어서 많은 장, 단점들이 존재한다. 고성능 시뮬레이터를 사용하면 좋은 결과를 이끌어낼 수 있지만 너무 비효율적인 측면에서 비효율적이다. 또한, 대부분의 시뮬레이터의 특성상 시뮬레이터를 자체적으로 변경하지 못하기 때문에 시나리오에 대해 제한적이고 단적인 결과만을 추출할 수 있다. 본 논문에서는 해당 단점들을 극복하고 ADAS의 손쉽고 효율적인 개발 및 검증을 위해 3D 자동차 시뮬레이터를 활용한 상호작용형 ADAS 개발 및 검증 프레임워크를 제안하였다.

P-ADAS는 AirSim 시뮬레이터에서 데이터를 수집하여 ADAS 기능 중 LKAS, AEB를 구현하였고 검증은 ADAS-DVS의 3D 차량 시뮬레이터인 AirSim 시뮬레이터를 사용하여 각 시나리오에 맞게 테스트를 진행했다. 해당 실험을 통해 P-ADAS의 LKAS, AEB 기능에 대해 각각 약 96%, 100%의 정확도를 보여줌으로써 본 논문에서 설계한 프레임워크가 신뢰성이 높다는 것을 보여준다.

References

- [1] M. König, and L. Neumayr, "Users' resistance towards radical innovations: The case of the self-driving car," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol.44, pp. 42–52, 2017. DOI:10.1016/j.trf.2016.10.013
- [2] U. Lee, S. Yoon, H. Shim, P. Vasseur, and C. Demonceaux, "Local path planning in a complex environment for self-driving car," *The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent*, pp.445–450, 2014. DOI:10.1109/CYBER.2014.6917505
- [3] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," *2013 IEEE Intelligent Vehicles Symposium*, pp.763–770, 2013. DOI:10.1109/IVS.2013.6629559
- [4] J. K. Choi, K. Kim, D. Kim, H. Choi, and B. Jang, "Driver-adaptive vehicle interaction system for the advanced digital cockpit," *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp.307–310, 2018. DOI:10.23919/ICACT.2018.8323736
- [5] Hassan, B, Berssenbrügge, J, Al Qaisi, I, & Stöcklein, J, "Reconfigurable driving simulator for testing and training of advanced driver assistance systems," *IEEE International Symposium*, pp. 337–339 2013. DOI:10.1109/ISAM.2013.6643472
- [6] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim : High-fidelity visual and physical simulation for autonomous vehicles," *Field and service robotics, Springer, Cham*, pp.621–635.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detectio," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.779–788, 2016.
- [8] S. U. Park and J. C. Kim. "Open Source Based Vehicle Control Algorithm Competition Platform for Education," *Korea Information Science Society*, pp.1411–1413, 2014.
- [9] D. S. Choi, T. H. An, K. H. An, and J. D.

- Choi, "End-to-end learning using TORCS for autonomous driving," *The Institute of Electronics and Information Engineers*, pp.740–743, 2017.
- [10] C. W. Park, S. O. Kim, H. C. Lee, "A Study on Approaching Mode of Adaptive Cruise Control System," *The Korean Society Of Automotive Engineers*, pp.666–669, 2014. DOI:10.1109/ISAM.2013.6643472
- [11] Hassan, B, Berssenbrügge, J, Al Qaisi, I, & Stöcklein, J, "Reconfigurable driving simulator for testing and training of advanced driver assistance systems," *IEEE International Symposium*, pp. 337–339, 2013. DOI:10.1109/ISAM.2013.6643472
- [12] J. S. Kim, "Effective Road Distance Estimation Using a Vehicle-attached Black Box Camera," *Korea Institute of Information and Communication Engineering*, pp.651–658, 2015. DOI:10.6109/jkiice.2015.19.3.651

BIOGRAPHY

Deun-Sol Cho (Student Member)



2018 : BS degree in Computer Science, Koreatech University

Sei-Youl Jung (Student Member)



2018 : BS degree in Computer Science, Koreatech University

Hyeong-Su Kim (Student Member)



2017 : BS degree in Computer Science, Koreatech University
2018 : MS degree in Computer Science, Koreatech University

Seung-Gi Lee (Student Member)

2018 : BS degree in Computer Science,
Koreatech University

Won-Tae Kim (Member)

1994 : BS degree in Electronics
Engineering, Hanyang University
1996 : MS degree in Electronics
Engineering, Hanyang University
2000 : PhD degree in Electronics
Engineering, Hanyang University
2001~2005 : CTO, Rostic Technologies
Inc.

2005~2015 : Team Manager, ETRI

2015~ : Assistant professor, Koreatech university