

< 응용 논문 >

## 자율 주행 시스템 개발을 위한 가상도시 구축 및 검증

김혜원<sup>1)</sup> · 조규세<sup>1)</sup> · 손준우<sup>\*2)</sup>

소넷 자율주행개발팀<sup>1)</sup> · 대구경북과학기술원 인간중심지능형시스템연구실<sup>2)</sup>

### Implementation and Verification of Virtual Environment for Autonomous Driving System Development

Heyone Kim<sup>1)</sup> · Gyuse Jo<sup>1)</sup> · Joonwoo Son<sup>\*2)</sup>

<sup>1)</sup>Autonomous Driving Development Team, Sonnet.AI, 333 Technojungang-daero, Hyeonpung-eup, Dalseong-gun, Daegu 42988, Korea

<sup>2)</sup>HumanLAB, Daegu Gyeongbuk Institute of Science and Technology, 333 Technojungang-daero, Hyeonpung-eup, Dalseong-gun, Daegu 42988, Korea

(Received 19 November 2020 / Revised 27 November 2020 / Accepted 8 December 2020)

**Abstract** : Since the invention of the car, technology has been continuously enhanced in order to reduce the physical and mental workload of the driver and to improve driving safety. One of these technologies is autonomous driving, which requires a lot of cost and time to develop. In order to save resources, a virtual driving environment can be utilized. In this study, we developed a simulator-based virtual environment for K-City, which is an autonomous driving test city in Korea Automobile Testing and Research Institute(KATRI), to design and verify the autonomous driving system. In order to validate the virtual K-City, we drove on an identical route in the real world and virtual world with the same autonomous driving software, and then we compared the driving routes. The comparison results showed a 92.5 % agreement in the 0.288 m error range.

**Key words** : Autonomous driving(자율 주행), Driving environment(주행 환경), Virtual environment(가상환경), CARLA simulator(칼라 시뮬레이터), K-City(케이시티)

### 1. 서론

자동차의 발명 이후, 운전자의 신체적, 정신적 부담을 줄이고, 운전자의 안전을 지키기 위해 기술을 계속 발전시켜 왔다.<sup>1,2)</sup> 이러한 기술이 발달하면서, 운전자 보조 시스템(Advanced Driver Assistant System, ADAS)과 자율주행 시스템(Autonomous driving system)이 등장하였고, 이러한 주제로 다양한 연구가 진행되고 있다.<sup>2,3)</sup> 또한, 각 차량의 정보를 통합하여 교통 신호를 통제할 수 있도록, 협력-지능형 교통 시스템(Cooperative-Intelligent Transport System, C-ITS)에 대한 연구도 진행하고 있다.<sup>4)</sup> 이 중, 자율주행 시스템은 운전자의 조작 없이 자동차 스스로 운행이 가능한 시스템으로, 주행을 위하여 Sensing, Perception, Planning, Operation 과정을 거친다.<sup>1,2)</sup> 이러한 과정을 거치기 위하여 차량 하드웨어를 구성하고, 실측 데이터와

사전 정보를 이용하여 자율주행 시스템을 구동한다.<sup>5,6)</sup>

이러한 주제의 연구를 수행하기 위하여 하드웨어 및 소프트웨어를 구현해야 하며, 구현 결과를 평가하는데 많은 시간과 비용이 필요하다. 이러한 비용과 시간을 줄이기 위하여 가상 환경을 이용할 수 있으며, 가상 환경을 구현하기 위한 시뮬레이터를 개발에 관한 다양한 연구가 진행되고 있다.<sup>7-10)</sup>

본 논문에서는 자율주행 시스템 개발 및 검증을 위하여 자동차안전연구원의 자율주행 실험도시(K-City)<sup>11)</sup>에 대한 가상 환경을 제작하고, 가상 자율주행 궤적을 실제 차량의 주행 결과와 비교하여 타당성을 검증하고자 한다. 이를 위해 자율주행 환경의 구성 요소와 자율주행 시스템의 구성 요소를 바탕으로 가상 환경을 설계하였다. 가상 환경은 K-City 고정밀 지도(High-definition map)를

\*Corresponding author, E-mail: [json@dgist.ac.kr](mailto:json@dgist.ac.kr)

<sup>1)</sup>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.

기반으로 구현하였으며, 교통 신호, 차량, 보행자, 날씨를 선택할 수 있다. 가상환경 내에서 사용하는 차량 모델은 CARLA 시뮬레이터에서 제공하는 기본 모델을 사용하였다.<sup>9)</sup> CARLA 시뮬레이터에서 구현한 가상 환경에서 당사에서 개발한 자율주행 시스템을 구동하여 가상 자율 주행실험을 실시하여 가상 환경의 유효성을 살펴보았다. 또한 K-City에서의 실제 자율주행하여 획득한 데이터와 가상 환경에서 자율주행 데이터를 비교함으로써 가상 주행 환경의 타당성을 검증하였다.

## 2. 자율 주행의 구성 요소

자율 주행의 구성 요소는 주행 환경과 자율주행 차량, 그리고 자율주행 제어 시스템으로 이루어진다.

### 2.1 자율 주행 환경의 구성

Fig. 1과 같이 자율주행 환경은 차량 주행로, 교통 통제 설비, 주행을 방해하는 물체, 그리고 날씨로 구성된다. 차량 주행로는 차량의 현재 위치, 목적지, 그리고 도로 교통법에 따라 결정된다. 교통 통제 설비는 교통을 규제, 지시하기 위해 사용한다. 이러한 설비로는 신호등, 표지, 노면 표지가 있으며, 관제소(Control center)에서 교통 신호를 통제한다. 주행을 방해하는 물체로는 고정형 물체와 기동형 물체로 나뉜다. 대표적인 고정형 물체는 바리케이드(Barricade)이며, 대표적인 기동형 물체는 보행자(Walker)이다. 날씨는 기온, 기압, 풍향, 풍속, 습도, 강수량으로 이루어진다.

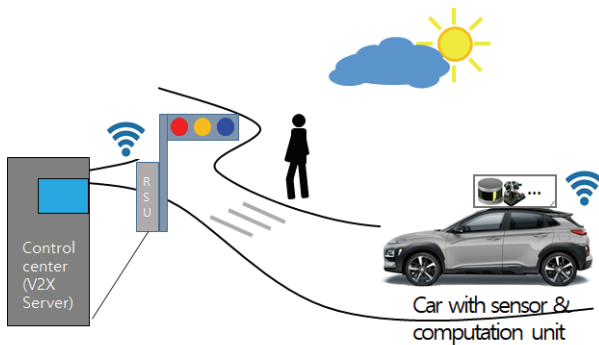


Fig. 1 Autonomous driving environment

### 2.2 자율 주행 차량의 구성

Fig. 2와 같이 자율 주행 차량은 주행 환경을 감지하는 센서부(Sensor unit), 주행 방향 및 속도를 제어하는 제어부(Control unit), 그리고 차량 주행을 수행하는 구동부(Driving unit)로 나뉜다. 센서부는 이미지 데이터 수집을 수행하는 Mobileye와 Camera, 물체까지의 거리를 측정

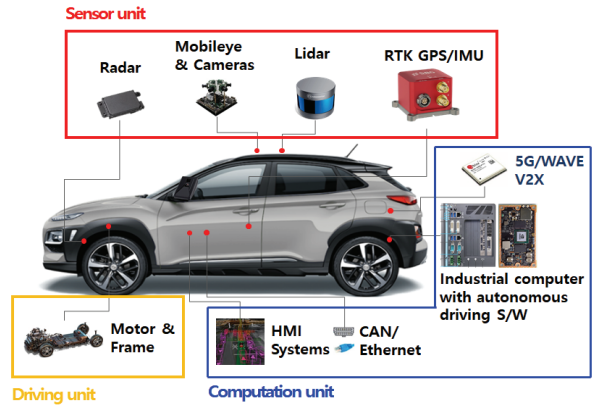


Fig. 2 Configuration of autonomous car

하는 Radar(Radio detection and ranging) 및 Lidar(Light detection and ranging)가 있다. 그리고 차량의 위치와 방향을 알려주는 RTK GPS/IMU가 있다. 이러한 센서부는 강수량 및 습도에 따라 오차가 발생할 수 있다. 제어부는 차량 제어에 필요한 데이터를 전달하는 CAN/Ethernet Interface, 자율 주행 소프트웨어를 탑재한 산업용 컴퓨터, 그리고 수집한 데이터와 제어 결과를 알려주는 HMI(Human-Machine Interface) 시스템이 있다. 구동부는 제어부의 명령에 따라 속도와 진행방향을 변경하며, 모터와 차체로 구성된다.

### 2.3 자율 주행 제어 시스템의 구성

자율 주행 시스템은 운전자 명령과 센서부의 출력, 사전에 수집한 지도 데이터(Map data)를 이용하여, 차량을 제어한다. 이러한 시스템 개발을 위해 다양한 라이브러리가 공개되어 있고, 여러 센서 출력을 취합하기 용이한 ROS(Robot Operating System)에서 자율 주행 제어 시스템을 구성할 수 있다.<sup>12)</sup>

Fig. 3과 같이 Perception에서는 정밀한 차량 제어를 위해 센서부 출력을 취합 및 보정하는 작업을 수행한다. 그 결과를 이용하여 물체 탐지(Object detection), 차로 탐지(Lane detection), 교통 신호 탐지(Traffic light detection), 그리고 시스템 모니터링 및 진단(System monitoring & diagnosis) 수행한다.

Planning은 Position estimator, Mission planning, Path/Motion planning, 그리고 Speed/Steering controller로 구성한다. 그리고, Perception 결과와 운전자의 명령, 그리고 지도 데이터를 입력 받는다. Position estimator에서 차량 위치를 추정하고, Mission planning에서 차량 주행 경로를 결정한다. Path/Motion planning에서 차량 주행을 방해하는 물체를 회피하며, Speed/Steering controller에서 차량 속도 및 진행 방향을 제어한다.

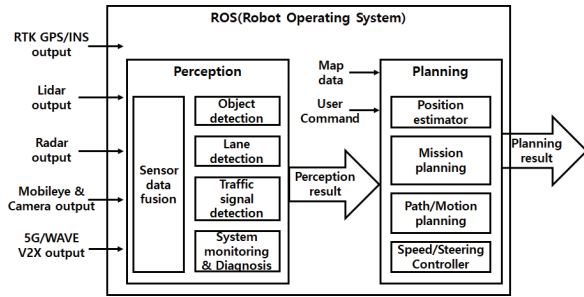


Fig. 3 Configuration of autonomous driving system

### 3. K-City 가상 환경

앞에서 언급한 주행 환경과 차량을 가상환경에서 설계 및 구현하고, 구현한 가상환경에서 자율 주행 제어 시스템과 연동하고자 한다. 실제 자율 주행 차량 제어 환경과 호환성을 유지하기 위하여 Fig. 4와 같이 가상 환경과 차량의 자율 주행 시스템을 분리한다. 그러므로 가상 환경은 자율 주행 시스템 구동에 필요한 센서 출력을 주어야 하며, 자율 주행 시스템은 차량 제어 명령을 가상 환경에 전달해야 한다.

가상으로 구현할 주행 환경은 경기도 화성에 있는 자율 주행차 실험 도시인 K-City로 선정하였다. K-City는 자율 주행에 특화된 가상 도시로 주행 환경에 대한 데이터 수집이 용이하고, 자율 주행 성능 평가에 유리한 환경이기 때문이다. Fig. 5에 K-City 개략도를 나타내었다.<sup>11)</sup>

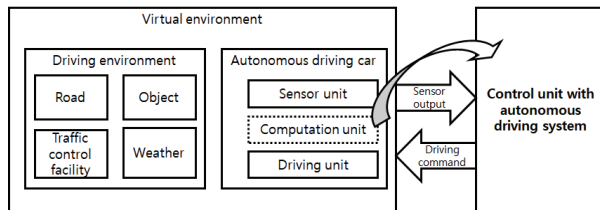


Fig. 4 Configuration of virtual autonomous driving environment



Fig. 5 K-City schematic<sup>11)</sup>

## 4. 가상의 K-City 구현 결과

### 4.1 CARLA 시뮬레이터용 가상 환경 구현

Ubuntu 18.04 버전에서 CARLA 시뮬레이터를 이용하여 가상 환경을 구현하였다. 가상 환경은 국토교통부의 국토정보플랫폼에서 제공하는 고정밀 지도(High-definition map)<sup>13)</sup>를 이용하여 자율 주행을 위한 맵 데이터 표준인 OpenDRIVE규격<sup>14)</sup>에 맞추어 구현하였다. 우선, 아래 Fig. 6과 같이 Mathworks사의 RoadRunner<sup>15)</sup>를 이용하여 가상 K-City 상의 도로를 구현하였다. 이를 위해 위도와 경도, 주행로, 교차로, 그리고 교통 신호에 대한 정보를 각각 설정하였다. 가상의 K-City를 Fig. 7과 같이 언리얼 엔진(Unreal engine) 4.24 기반인 CARLA 시뮬레이터에 임포트하고, 구동한 CARLA 시뮬레이터에서 가상의 차량을 생성하였다. 또한, 로스 브릿지(ROS bridge)를 이용하여 기존에 자율 주행 차량용으로 개발된 ROS 소프트웨어와 연계하여 가상 환경 및 차량으로부터 데이터를 수집하고, 차량에 명령을 전달하였다. 자율 주행 시스템을 구동하여 가상 환경에서 수집한 데이터와 주행로 데이터(Map data)를 이용하여 가상 K-City 내의 차량을 제어하였다.

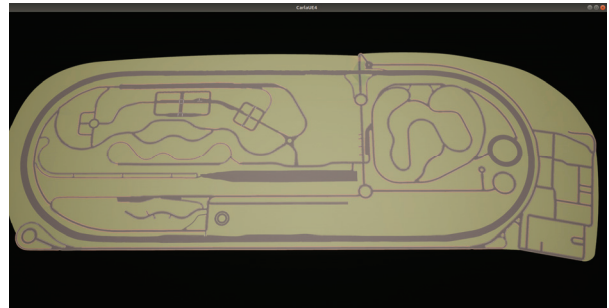


Fig. 6 Virtual K-City in Carla simulator



Fig. 7 Driving a car in virtual K-City

### 4.2 가상의 K-City 검증 방법

가상의 K-City 검증을 위해 실제 K-City와 가상의 K-City의 유사도를 판단하여 검증하였다. 고정밀 지도를 기반으로 구현한 가상 K-City와 실제 K-City를 직접 비교하기 위해서는 고성능의 GNSS 혹은 라이다 센서가 필요하다. 또한 주행로의 크기가 클수록, 객체의 수가 많을수록 검증에 필요한 시간과 비용이 커진다. 검증에 필요한 시간과 비용을 줄이기 위하여, 가상 K-City와 실제 K-City에서 차량의 이동 경로를 비교하여 유사도를 판단하였다. 가상 환경 검증 과정을 Fig. 8에 나타냈다. 검증을 위해 동일한 자율 주행 제어 시스템을 사용하였으며, 가상 환경과 실제 환경에서 자율 주행하는 차량의 위치 데이터를 수집하였다. 실제 자율 주행에서 수집한 차량 위치 데이터와 가상 차량 위치 데이터가 일정 범위(예, 0.288 [m]) 이내에 존재하면, 일치한 것으로 판정하였다. 일치한 위치 데이터의 개수와 수집한 위치 데이터의 개수를 이용하여 유사도를 판단하였다.

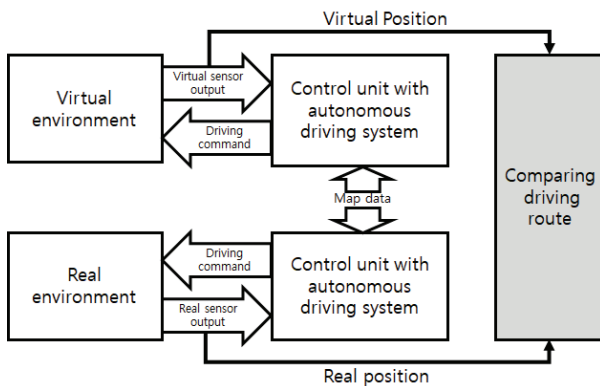


Fig. 8 Virtual environment verification

#### 4.2.1 주행 환경

검증을 위해 가상 K-City의 주행 환경이 실제 K-City와 동일하도록 구성하였다. 앞서 설명한 바와 같이 주행 환경은 차량 주행로, 교통 통제 설비, 주행을 방해하는 물체, 그리고 날씨로 나뉜다. 차량 주행로는 Fig. 6과 같이 실제 K-City 차량 주행로를 기반으로 가상 K-City 차량 주행로를 구성하였다. 또한 교통 통제 설비는 차량이 주행 가능한 교통 신호만을 송출하며, 주행을 방해하는 물체가 없도록 구성하였다. 또한 날씨는 구름이 없는 맑은 날을 기준으로 주행 환경을 구성하였다.

#### 4.2.2 가상환경 검증을 위해 설정한 차량 이동 경로

자율 주행 데이터 수집을 위하여 차량의 이동 경로를

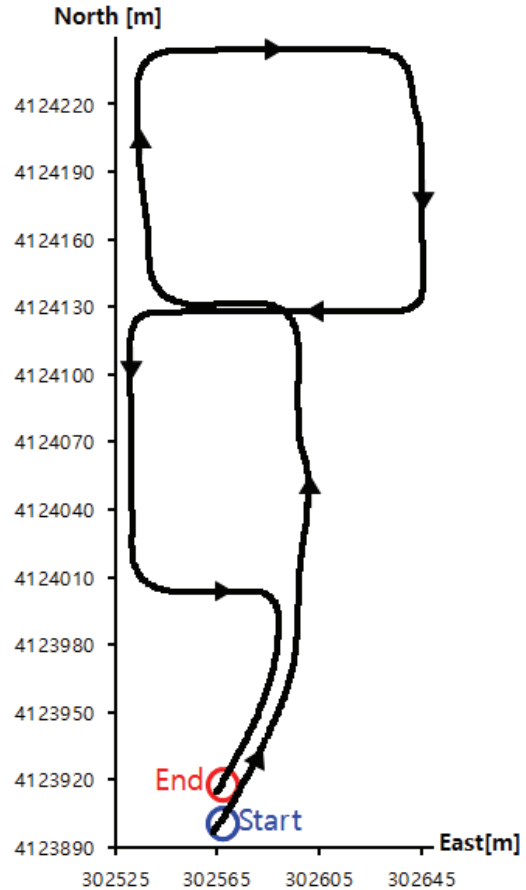


Fig. 9 Map data for comparing real K-City and virtual K-City

Fig. 9와 같이 구성하였다. 차량의 이동 경로는 평면직각 좌표계인 UTM(Universal Transverse Mercator) 좌표계로 나타냈다. UTM 좌표계는 타원체인 지구를 경도 및 위도를 기준으로 1,200개의 구역으로 나누며, 이 중 52 S 구역에 K-City가 포함된다. 주행 경로는 총 5개의 교차로와 7개의 횡단보도를 통과하도록 구성하였으며, 2회의 차로 변경을 수행하였다. 이렇게 구성한 차량의 이동 경로는 실제 차량 혹은 가상 차량을 제어할 자율 주행 제어 시스템에서 사용하였다.

#### 4.2.3 실제 K-City 자율 주행 데이터 수집

Fig. 10과 같이 실제 K-City에서 자율주행을 위해 개조한 코나 차량으로 주행 데이터를 수집하였다. 본 차량에는 RTK GPS/INS 센서를 장착하였으며 최고 속도 50 km/h 이하의 속도로 주행하였다.

#### 4.2.4 가상 K-City 자율 주행 데이터 수집

앞서 구현한 가상 K-City를 CARLA 시뮬레이터로 구동하였으며, 이를 위하여 4 GB 이상의 GPU로 PC를 구성

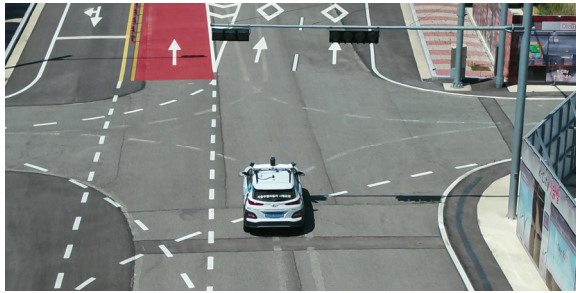


Fig. 10 Autonomous driving in real-world K-City



Fig. 11 Autonomous driving in virtual K-City

하였다. 앞에서 가상 환경에 대하여 설명한 바와 같이 CARLA 시뮬레이터와 자율 주행 시스템을 별도로 구성하였다. 이를 통해 가상 환경과 자율 주행 제어 시스템을 각기 다른 PC로 구동하였다. 분리한 CARLA 시뮬레이터와 자율 주행 시스템은 ROS bridge를 통해 데이터를 전달한다. 이렇게 구성한 결과를 Fig. 11에 나타냈다.

### 4.3 실험 결과

위와 같이 실제 K-City와 가상 K-City에서 동일한 자율 주행 시스템을 구동하여 데이터를 수집하였다. 차량의 위치 정보는 UTM 좌표계로 나타냈으며, 이를 바탕으로 Table 1에 실제 K-City와 가상 K-City의 유사도를 분석한 결과를 나타내었다. 실제 K-City를 동일한 가상 환경으로 구현하기 어렵고, 센서의 지연 시간 오차와 차량 구동에 소모하는 시간을 고려하여 허용 위치 오차를 0.288 [m] 이상으로 두었다. 아래 Table 1에서 보는 바와 같이 허용 위치 오차가 클수록 유사도가 커지는 것을 알 수 있다. 또한 가상 K-City를 이용한 자율 주행 결과는 실제 K-City와 92.5 % 이상의 유사도를 가지는 것을 알 수 있다.

가상 K-City와 실제 K-City의 자율 주행 데이터 차가 발생하는 원인으로서는 센서의 지연시간 오차, 가상 환경에서 구현한 K-City 주행로의 정확도, 차량 모델의 정확도, 그리고 풍속 및 풍향 변화를 들 수 있다.

Table 1 Virtual K-City performance evaluation

Allowable error [m]	Coincidence rate (K-City, Virtual)		
	Coincidence point	Total point	Coincidence rate
0.288	838	906	92.5 %
0.504	904		99.8 %
1.008	904		99.8 %

## 5. 결론

본 논문에서는 자율 주행 시스템 개발을 위하여 실제 K-City를 가상 환경으로 구축하고 검증하였다. 이를 위해 자율 주행 환경의 구성 요소와 자율 주행 제어 시스템의 구성을 확인하였다. 이러한 구성 요소를 바탕으로 가상 환경을 설계하였다. 가상 환경은 K-City 고정밀 지도 (High-definition map)를 기반으로 구현하였다. 구현한 가상 환경과 자율 주행 시스템을 구동하였고, 실제 K-City와 가상 K-City 주행 결과를 비교하였다. 그 결과, 0.288 [m] 오차 범위에서 92.5 %의 일치도를 나타내었다.

추후 연구에서는 가상환경에서 차량과 보행자와 같은 객체를 회피하는 시나리오와 교통 신호 변화 시나리오를 구성하고, 성능을 평가하고자 한다. 이를 위해 가상환경에서 출력하는 운전자 시점의 이미지, 차량 위치 및 방향, 차량과 객체 사이의 거리를 이용한다. 이를 바탕으로 차량 이동 경로 정확도, 객체 감지율, 객체 감지까지의 처리 시간, 객체 위치 정확도, 신호등 감지율, 생성한 차량 주행 경로의 안전성 평가에 대한 연구를 진행 할 필요가 있다.

## 후 기

본 연구는 국토교통부 도심도로 자율협력주행 안전·인프라 연구사업의 연구비 지원(과제번호 20PQOW-B152473-02)에 의해 수행되었습니다.

## References

- 1) J. Wang, J. Liu and N. Kato, "Networking and Communications in Autonomous Driving: A Survey," IEEE Communications Surveys & Tutorials, Vol.21, No.2, pp.1243-1274, 2019.
- 2) R. Okuda, Y. Kajiwara and K. Terashima, "A Survey of Technical Trend of ADAS and Autonomous Driving," Proceedings of Technical Program - 2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA), Hsinchu, 2014.

- 3) A. Paul, R. Chauhan, R. Srivastava and M. Baruah, "Advanced Driver Assistance Systems," SAE 2016-28-0223, 2016.
- 4) S. Brodsky, A. Panferov, A. Nebylov and D. Chikrin, "Integrated Navigation and Distributed Control Intelligent Transport System," 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russia, 2020.
- 5) I. S. Weon and S. G. Lee, "Environment Recognition Based on Multi-sensor Fusion for Autonomous Driving Vehicles," Journal of Institute of Control, Vol.25, No.2, pp.125-131, 2019.
- 6) K. W. Min, S. J. Han, D. J. Lee, D. S. Choi, K. B. Sung and J. D. Choi, "SAE Level 3 Autonomous Driving Technology of the ETRI," 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, pp.464-466, 2019.
- 7) J. A. Sokolowski and C. M. Banks, Principles of Modeling and Simulation: A Multidisciplinary Approach, John Wiley & Sons, Hoboken, 2009.
- 8) N. Fouladinejad, N. Fouladinejad, M. K. Abd Jalil and J. M. Taib, "Modeling Virtual Driving Environment for a Driving Simulator," 2011 IEEE International Conference on Control System, Computing and Engineering, Penang, pp.27-32, 2011.
- 9) CARLA Team, <https://carla.org>, 2020.
- 10) M. Park and J. Son, "Reference Test Scenarios for Assessing the Safety of Take-over in a Conditionally Autonomous Vehicle," Transactions of KSAE, Vol.27, No.4, pp.309-317, 2019.
- 11) KATRI(Korea Automobile Testing & Research Institute), <http://katri.or.kr/web/contents/katri2030205.do>, 2020.
- 12) OSRF(Open Source Robotics Foundation), <https://www.ros.org>, 2020.
- 13) Y. S. Na, S. K. Kim, Y. S. Kim, J. Y. Park, J. M. Jeong, K. C. Jo, S. Lee, S. J. Cho, M. Sunwoo and J. Oh, "HD Map Usability Verification for Autonomous Car," Transactions of KSAE, Vol.28, No.11, pp.797-808, 2020.
- 14) ASME, OpenDRIVE, <https://www.asam.net/standards/detail/opensdrive/>, 2020.
- 15) MathWorks, <https://www.mathworks.com/products/roadrunner.html>, 2020.