JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 26, No. 2: 278~287, Feb. 2022

드라이빙 시뮬레이터 시나리오 개발을 위한 동적 도로환경 데이터 융합

권준호[1] · 전연수[2] · 염춘호[3*]

Integration of Dynamic Road Environmental Data for the Creation of Driving Simulator Scenarios

Joonho Gwon[1] · Yeonsoo Jun[2] · Chunho Yeom[3*]

[1]Undergraduate Student, Department of Computer Science and Engineering, University of Seoul, Seoul, 02504 Korea
[2]Researcher, International School of Urban Sciences, University of Seoul, Seoul, 02504 Korea
[3*]Professor, International School of Urban Sciences, University of Seoul, Seoul, 02504 Korea

요 약

기술발전에 따라 드라이빙 시뮬레이터는 다양한 용도로 활용되고 있다. 드라이빙 시뮬레이터 실험에서 시나리오 개발은 실험결과의 신뢰를 높이고 연구목표를 달성하며 운전자에게 보다 실제같은 경험을 제공하는데 필수적이다. 그러나 시나리오를 개발하는데 데이터베이스 형성과 실시간 시나리오 운영 등에는 아직도 제약이 많다. 본 연구는 이러한 환경에서 실제 도로에서 실시간 주행속도와 기상데이터를 수집하고 활용하는데 가능성을 확인하고자 한다. 또한 본 연구를 통해 아두이노 센서 데이터와 공공API 데이터를 연계하는 방안도 제시하고자 한다. 연구결과의 검증을 위해 실제도로에서 시험을 실시했으며 본 연구를 통해 드라이빙 시뮬레이터에서 실시간 데이터를 활용한 시나리오 개발에 도움이 될 것으로 기대한다.

ABSTRACT

With the development of technology, driving simulators have been used in various ways. In driving simulator experiments, scenario creation is essential to increase fidelity, achieve research aims, and provide an immersive experience to the driver. However, challenges remain when creating realistic scenarios, such as developing a database and the execution of scenarios in real-time. Therefore, to create realistic scenarios, it is necessary to acquire real-time data. This study intends to develop a method of acquiring real-time weather and traffic speed information for actual, specific roads. To this end, this study suggests the concatenator for dynamic data obtained from Arduino sensors and public open APIs. Field tests are then performed on actual roads to evaluate the performance of the proposed solution. Such results may give meaningful information for driving simulator studies and for creating realistic scenarios.

키워드 : 도로환경데이터, 드라이빙 시뮬레이터, 아두이노, 공공API

Keywords : Road environmental data, Driving simulator, Arduino, Open API

Received 6 December 2021, **Revised** 24 December 2021, **Accepted** 29 December 2021

*** Corresponding Author** Chunho Yeom(E-mail:chunhoy7@uos.ac.kr, Tel:+82-2-6490-5154)
Professor, International School of Urban Sciences, University of Seoul, Seoul, 02504 Korea

Open Access **http://doi.org/10.6109/jkiice.2022.26.2.278** print ISSN: 2234-4772 online ISSN: 2288-4165

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.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. Introduction

As computer technologies have improved in recent years, driving simulators (DSs) are widely used in research for training, assessing the impact of new regulations, human factor monitoring, designing, and evaluating roadway configuration [1-4]. The DS is a virtual reality facility consisting of subsystems, such as real-time vehicle simulations, motion systems, operation consoles, and video and audio systems [5]. DSs are a safe and efficient method to replicate actual driving situations in a controlled laboratory environment [5-8]. In terms of physical aspects, the DS conveys perceptual cues to the driver based on representations of three methods: mathematical structures, physical analogs, or a combination of the two [4, 9]. Here, perceptual cues consist of vehicles, scenarios, controls, roadways, and environmental cues [9].

To mockup an actual or future road project in DSs, a scenario is required [3]. Creating a DS scenario requires the setting and environment of the simulation to be determined and the simulation actions to be choreographed [10]. Although the definition of scenario differs depending on the study, this study describes the DS scenario as a virtual static layout (i.e., scene) and the dynamic environment or activity of the simulation [10-12]. The need for scenario complexity is determined according to driving tasks and research questions; this can affect the workload of participants [13, 14].

The scenarios are essential in DS experiments. First, researchers generate scenarios to evaluate the hypotheses [10, 15]. Second, developing road models ignoring factors such as signs and markings may negatively affect vehicle performance and driver behavior [11]. Similarly, the behavior of DS drivers can be affected by the richness of visual scenes, the density of virtual environments, and other users' behavior [3]. Therefore, to effectively use DSs, the scenario specification is important because it determines when and where dynamic objects will take action at the experimental design stage [10]. Third, the physical validity of the DSs can be affected by the information provided to drivers through different sensory modalities of the virtual environment and movement [3]. Finally, the scenario can also affect the fidelity of DSs. With flight simulation studies, the concept of visual fidelity can be defined by the visual detail level of the simulators' virtual environment [16]. In general, fidelity is defined as how the senses perceive the simulation as real [14, 17]. Regarding physical fidelity, including visual fidelity, when the DS driver's experience seems to match the sensory experience in real situations, it is expressed as high-fidelity [9]. In summary, the detail of the virtual environment refers to visual fidelity. When the visual detail level is high, the experience of the DS driver matches the actual sensory experience.

The usual scenario development process is as follows. The static environment of the DSs scenario reproduces the site by converting 2D computer-aided design (CAD) drawings into 3D models [18]. It includes specific road elements such as markings, buildings, and traffic signs [18]. Moreover, it creates effects such as texturing, shading, and lighting [4, 18]. Commercial programs such as Google Sketchup and Blender are used for developing such a scenario [18-19]. In addition, 3D conversion (e.g., of 2D CAD) software such as Autodesk Civil 3D can utilize 3D CAD files representing roadway alignments [18-21]. When the 3D model is prepared for simulation, the simulator converts the 3D model into the scenario and roadway metadata. Specifically, behaviors and events are coded in the simulator [18]. Meanwhile, scenario manipulation is performed through a scenario control subsystem, including object placement, traffic generation, event activation, and the setting of weather conditions [10, 22]. The scenario control system manages the execution of scenario models and actions and the database to respond to static or dynamic queries of the virtual environment [23]. These control systems should efficiently represent complex and dynamic virtual environments. In addition, the systems need to present current events repetitively and provide real-time performance [22].

The scenario related requirements to represent virtual driving environments are modeling and controlling reliable scenarios and large scale virtual environment database modeling [23]. The database includes visual, logical, and terrain components; These components represent the virtual environment of DS [23]. Specifically, visual components are coordinated to create images, representing the scene at multiple levels of detail; The logical database is used for scenario control, including high level representations such as networks, vehicles, signs, and lanes. The terrain component includes surface type, roughness, and friction, and it is used for computations of vehicle dynamics [23]. After all, different levels of static and dynamic information are needed to represent the processes and objects of a DS scenario [24].

However, scenario creation has some problems, including scene modeling for the DS virtual reality environment [23]. Although the visual environment must be modeled and rendered accurately, generating 3D databases is difficult and time consuming [3]. To specify, simulators cannot precisely replicate the scenarios experienced in reality [25]. Apart from this, the primary constraint is deterministic, real-time execution. In other words, since the scenario system is part of a large set of cooperating modules, the control and real-time execution of the scenario is limited [23]. Additionally, if there are unexpected interactions between the driver and the surrounding traffic or scripted behaviors, the timing of the task may be disrupted, and the scenario may fail [10]. In summary, it is challenging to develop a database that represents the visual environment realistically. Even if it is developed, real-time or intended execution of the scenario is challenging.

Consequently, it is necessary to acquire actual road data, in real-time, during the primary stage to create the DS scenarios realistically. It is because data driven scenario creation can influence the improvement of visual fidelity and the immersive experience of the driver. Regarding road scene creation, static road information is usually obtained through road design documents or photographic logs [18, 19]. Thus, this study proposes a method for acquiring dynamic road environmental data by developing a prototypical solution to get real-time data of actual roads. In addition, this study suggests an approach for acquiring weather information with Arduino sensors and acquiring traffic speed information of public open API [26] to collect these two types of data in Comma Separated Values (CSV) format.

Ⅱ. Proposed Solution

2.1. Project Structure

Before defining the methodology of this study, previous methods are reviewed. In previous studies, road data integration methodologies and data sources are varied [27-35]. Similar to this research, a study uses weather conditions and traffic as data sources [34]; It proposes a comprehensive classifier system that integrates historical and real-time data for real-time traffic accident prediction. Weather conditions and traffic updates from the PARAMICS micro-simulator are utilized to develop the classifier system. Before the simulation, PARAMICS' parameters are adjusted to reflect recent public traffic data or observed data. On the other hand, static and historical data are stored in different databases according to their characteristics, and the location parameter links these databases. The Bayesian Network for prediction is constructed based on historical data. It then predicts accident risk, in real-time, based on real-time input data [34].

Also, there is another effort to develop the LDM platform that integrates various information, from a permanent static geographic information system layer to a dynamic object information layer [27-30]. LDM can be implemented on a computer using PostgreSQL, PostGIS, and PL/pgSQL, based on the LDM specification suggested by the SAFESPOT project [31]. This previous study specifies the design and implementation of LDM as follows. As static data, OpenStreetMap data is stored in the LDM road element table [31]. The vehicle sensor model and driving scenarios are created using a simulation platform to acquire the dynamic traffic data. Then, it creates a CSV file with data such as location information, speed, and acceleration as the final output. Finally, as dynamic data, the vehicle data CSV file is read by the update program and registered in the vehicle data tables of LDM based on the sensing period [31].

As shown in Fig. 1, this study presents a method to acquire weather data with Arduino and traffic data through public open API. Then, the result of this study is organized in a CSV file, referring to the study of LDM design based on CSV format [31]. Considering these previous studies, even if this study result is saved in CSV format, it will be helpful in the future. Also, the method of acquiring road data is different from other studies for integration road data.

There are two independent loops. Sensor Loop runs on the Atmega32u4 once every two seconds. In this loop, Atmega32u4 continuously gets the humidity and temperature data from the DHT11/DHT22 sensor and then pushes it into the hardware bridge.

Main Loop runs on a laptop and also runs once every two seconds. This loop does the following steps. First, the laptop requests humidity and temperature data from AR9331 on Arduino Yun via a local Wi-Fi connection. Then, AR9331 reads the required data from the hardware bridge and responds to the laptop. When the required data is received, the computer requests traffic data from the public open API server via the internet. If both data

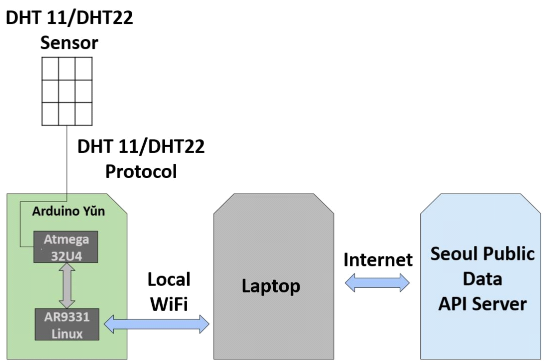

Fig. 1 Project structure

are obtained, the program concatenates the two data with a timestamp and appends them to a CSV file.

The specific execution process is as follows. Firstly, Arduino Yun and a PC are connected to the same Wi-Fi network. It is recommended to use a PC as a hotspot AP and connect Arduino to it. Second, update required files on Arduino Yun. The Sketch file can be uploaded via Sketch IDE, and the Python [36] script can be uploaded via Secure Copy Protocol. Next, execute the Sketch file and Python file on Arduino. If the Sketch file is uploaded correctly, it will be executed automatically, and the Python file should be run manually via Secure Shell Protocol (SSH). Finally, execute Python script for PC. Then it will automatically connect to Arduino, obtain the humidity, temperature, and traffic data from Arduino and public open API, and generate a CSV file containing timestamps. Additionally, the HOST variable in the Python script for PC should be changed to the proper IP address. For the entire code, visit the following GitHub repository: https://github.com/unknownpgr/dhtiot.

2.2. Data Source

First, this study develops an IoT sensor based on Arduino Yun to acquire the weather data of the road link. Arduino has been widely used in various studies as a prototype. It has the advantages of flexibility, platform independence, cost effectiveness, ease to program, open source hardware, and software [37-40].

Arduino Yun is an IoT board with Ethernet and Wi-Fi connectivity options. Additionally, it has the unique ability to integrate Linux for stability and security. It is based on a dual processor system; the Atmega32u4 is a microcontroller, and Atheros AR9331 supports Linux and Wi-Fi [40-43]. Atmega32u4 and Atheros AR9331 communicate with each other through the bridge module. Arduino Yun also provides a convenient communication function between the sensor and the internet [43].

Accordingly, Arduino Yun is installed on an actual road in this study, and DHT11/DHT22 sensors [39] are used as weather sensors. Temperature and humidity are acquired as weather data. The schematic figure for hardware configuration is shown in Fig. 2.

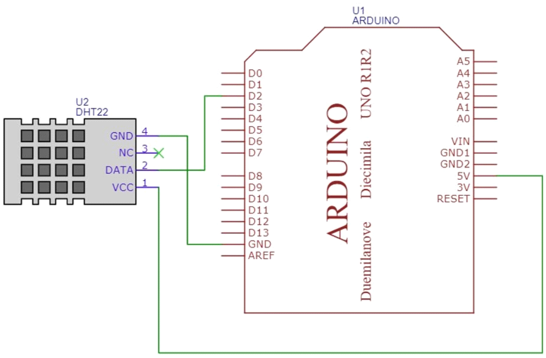

Fig. 2 Schematic of hardware configuration

Second, a public open API is used to acquire traffic speed data. According to the definition of traffic data, traffic speed data uses the vehicle global positioning system information of about 70,000 card taxis in Seoul. The public agency collects link speeds every 5 minutes and processes them to provide statistical speed. As shown in Fig. 3, speed data is acquired in this study based on the link ID of the road where Arduino Yun is installed. There are two types of link IDs, Standard link and Service link. The public open API server uses the Service link.

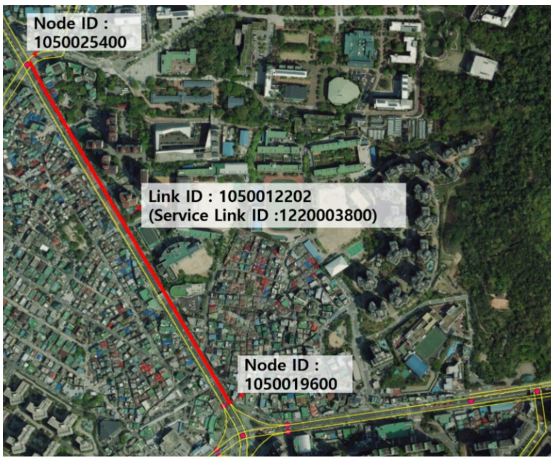

Fig. 3 Node and link in GIS objects for traffic speed data

Ⅲ. Results

As shown in Fig. 4, the sensor was installed on the road, and a field test was performed to model the proposed solution. The location of the field test was 135-24 Jeonnong-dong, Dongdaemun-gu, Seoul. The sensor was installed on the roadside at the intersection close to the traffic speed data link. It will be possible to be installed on a road drain or roadside through the development of a sensor case. The experiment was conducted from 14:45 on 2021.02.26.

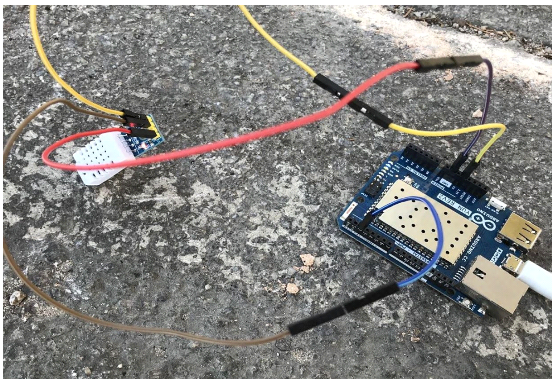

Fig. 4 Arduino sensor in the field test

The field test results indicated that the sensor and overall solution work well, and the results were displayed as a CSV file, as shown in Fig. 5. The CSV file shows the humidity, temperature, and traffic speed data along with a timestamp in seconds. Humidity and temperature are data acquired based on DHT11/DHT22 sensors. Temperature data are read with an accuracy of ±0.5℃ in the range from -40 to 80℃; Humidity data are from 0% to 100% humidity range with 2-5% accuracy. Traffic speed data means the 5-minute average speed of the Seoul card taxi that passed the link section. The public API's limitation is that it cannot provide the real-time speed of all vehicles passing through the actual area.

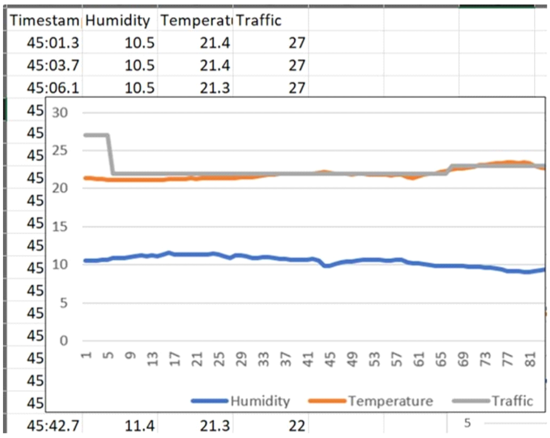

Fig. 5 CSV file result of the field test

The experimental weather data results, the sensor data, can be compared with the public open data. AWS(Automatic Weather System) data are public weather data provided in real-time in Korea. AWS data is acquired every minute, and there is a limit to providing only data corresponding to representative points. Specifically, in the case of a neighborhood unit, the AWS representative point value becomes the value of the neighborhood weather forecast considering the distance and altitude within the grid. In other words, the representative point data is deemed to be representative of the corresponding neighborhood. The comparison between the sensor values and the representative AWS data of Dongdaemun-gu is shown in Table 1. About the sensor value, the temperature is higher, and the humidity is lower than the representative AWS data. This difference comes from the difference in data acquisition points and the difference in the time unit. But the future research needs to verify the sensor values and find out precisely the reason for the difference.

Table. 1 Comparison between sensor data and AWS data (source: [44])

Contents (14:45)	Sensor data(Average)	AWS data
Temperature data	21.2	14.9
Humidity data	11.2	14.0

Ⅳ. Discussion

4.1. Assessment and explanation of the results

The subjects of this study, weather conditions and ambient traffic, have been studied using DSs. Mainly, weather condition scenarios have been conducted through various studies [45-49]. Studies usually analyze the effect of weather conditions on driving behavior. Weather conditions such as no fog, light fog, heavy fog, clear, snow, and rain are suggested [45-49]. In studies related to fog or hazy conditions, visibility is explicitly specified [45, 46, 49]. Significant research classifies the levels of rain, snow, and fog then presents the related actual weather grade in terms of visibility and rain (mm/24h) [47].

Regarding traffic, programmed vehicles become a backdrop to ambient traffic or are utilized to enable research conditions in the simulator [10]. Here, the DS scenario can be more realistic by using measurements describing traffic behaviors [18]. In some traffic behavior algorithms, each vehicle can represent parameters such as speed, position, safety distance, maximum desired acceleration, and braking in front and rear vehicles [24]. In another scenario control software, the vehicle elements are deterministic dynamic objects (DDOs) and autonomous dynamic objects (ADOs) [12]. DDOs are objects that follow a scripted path and speed control without autonomous motion, while ADOs are those derived from autonomous driver models that show driver like driving behavior [12].

Consequently, simulated traffic and interacting agents must act realistically to provide immersion to DS drivers and not affect their behavior [3, 24]. However, since driving behavior is very complex, it is challenging to simulate real traffic and control scenarios [10]. In DS experiments, researchers use the traffic level or a microsimulation model that adjusts the surrounding vehicle speed [15, 50]. Additionally, a study explicitly suggests adopting three traffic flow states produced by combining elements, such as car following situations, speed, and headway time [47].

Most DS related studies do not use real-time, dynamic data of actual road links, such as weather conditions and ambient traffic speed. Nevertheless, this study can be compared with a previous study that presented a specific case of weather and traffic scenario creation. The previous study matches the similarity between simulated weather conditions and actual weather grades through a visual experiment. It uses related studies to present the speed as a configuration of traffic flow states [47]. However, this previous study does not acquire real-time data and does not use actual road link weather and speed data. The weather or ambient traffic data used in DS scenarios is usually based on the available data in the previous studies. This is the main difference between this study and previous research. In other words, using dynamic data is common to both the current and previous work, but this study distinctively suggests a method to acquire real-time data of actual roads.

4.2. Research Limitations

The study has some limitations. First, it does not verify the sensor values as mentioned in 4.1. section. Second, there is no DS scenario creation based on the results of this study. Third, the software has some limitations; it is impossible to perform complicated tasks on Arduino Yun because of the lack of storage space. Therefore, most of the tasks are performed on a computer, causing a bottleneck. Also, installing Python3 is impossible because of the same reason. Thus, only Python2, a deprecated version of the language, can be used. Overall, except from the software point of view, future research is proposed to verify the sensor results and create a realistic DS scenario.

4.3. Future Research

At first, to verify the sensor values, there is a method of measuring the variability of a sensor value for a certain period. And, it can also be compared to the values of various types of sensors to minimize the influence of external conditions such as sunlight.

Secondly, there is the need for the incorporation of sensor data into simulator experiments. For the creation DSs scenarios based on data, future research could lead to the adjustment of the resulting data according to the traditional DS scenario setting system. With considering the related studies, there are some methods to incorporate the data into simulators. For example, a survey of drivers' experiments can match sensor values to the weather setting of the DS software regarding the weather conditions. Also, it is possible to compare field data and DS data on driving behavior such as traffic speed [50]; In other words, it is necessary to develop various DS scenarios based on data and compare each scenario. In addition, to evaluate the realism of the data used in the simulation, the simulator driving needs to be compared with the actual driving. Specifically, future research can perform NASA-TLX and participation realism survey with Likert scale [51, 52].

Ⅴ. Conclusion

This study develops a solution to collect weather conditions and traffic speed data in real-time. Since this solution acquires actual road environmental data, it can be used as a visual data source for creating realistic DS scenarios. Furthermore, because the solution has real-time properties, it can help establish the basis for real-time control and execution of DS scenarios. As a result, the authors believe that the research should be helpful for researchers to develop more reliable solutions while using a driving simulator.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2020S1A5C2A01092978).

References

[1] R. Zha and W. Tang, “Optimal design of driving simulator scenario system based on speed perception,” in *2019 IEEE International Conference on Computation, Communication and Engineering (ICCCE)*, Fujian, China, pp. 178-180, 2019.

[2] T. M. Granda, G. W. Davis, V. W. Inman, and J. A. Molino, *The use of high-fidelity real-time driving simulators for geometric design*, CRC Press: Boca Raton, FL, USA, 2011.

[3] F. Vienne, S. Caro, L. Désiré, J. Auberlet, F. Rosey, and E. Dumont, “Driving simulator: an innovative tool to test new road infrastructures,” in *TRA-Transport Research Arena*, Paris, France, pp. 10, 2014.

[4] H. S. Kang, M. A. Jalil, and M. Mailah, “A PC-based driving simulator using virtual reality technology,” in *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, Los Angeles: CA, pp. 273-277, 2004.

[5] W. Lee, J. Kim, and J. Cho, “A driving simulator as a virtual reality tool.” In *Proceedings of 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, Leuven, Belgium, pp. 71-76, 1998.

[6] P. A. Hancock and T. B. Sheridan, *The future of driving simulation*, In Handbook of Driving Simulation for Engineering, Medicine, and Psychology, 1st ed. CRC Press: Boca Raton, FL, USA, 2011.

[7] Y. Zhang, Z. Guo, and Z. Sun, “Driving simulator validity of driving behavior in work zones,” *Journal of Advanced Transportation*, vol. 2020, no. 1, pp. 1-10, Jun. 2020.

[8] F. Bella, “Effects of combined curves on driver's speed behavior: Driving simulator study,” *Transportation Research Procedia*, vol. 3, no. 1, pp. 100-108, Jul. 2014.

[9] J. A. Greenberg and M. Blommer, *Physical fidelity of driving simulators*, In Handbook of Driving Simulation for Engineering, Medicine, and Psychology, 1st ed. CRC Press: Boca Raton, FL, USA, 2011.

[10] J. K. Kearney and T. F. Grechkin, *Scenario authoring*, In Handbook of Driving Simulation for Engineering, Medicine, and Psychology, 1st ed. CRC Press: Boca Raton, FL, USA, 2011.

[11] D. Evans, *The importance of proper roadway design*, In Handbook of Driving Simulation for Engineering, Medicine, and Psychology, 1st ed. CRC Press: Boca Raton, FL, USA, 2011.

[12] Y. Papelis, O. Ahmad, and M. Schikore. (2001, June). Scenario definition and control for the national advanced driving simulator. *SAE Technical Paper* [Online]. Available: https://www.sae.org/publications/technical-papers/content/2001-06-0219.

[13] H. A. Stoner, D. L. Fisher, and M. Mollenhauer, *Simulator and scenario factors influencing simulator sickness*, In Handbook of Driving Simulation for Engineering, Medicine, and Psychology, 1st ed. CRC Press: Boca Raton, FL, USA, 2011.

[14] J. Mueller, L. Stanley, T. Martin, and C. Gallagher, “Driving simulator and scenario effects on driver response,” in *Proceedings of IIE Annual Conference*, Toronto, Canada, pp. 507, 2014.

[15] J. A. Mueller, “Driving in a simulator versus on-road: the effect of increased mental effort while driving on real roads and a driving simulator,” Ph. D. dissertation, Montana State University, Bozeman, MT. 2015 [Online]. Available: https://scholarworks.montana.edu/xmlui/handle/1/9068.

[16] D. Liu, N. D. Macchiarella, and D. A. Vincenzi, “Simulation fidelity,” in *Human Factors in Simulation and Ttraining*, 1st ed. CRC Press: Boca Raton, FL, USA, pp. 61-73, 2008.

[17] J. De Winter, P. A. Wieringa, J. Dankelman, M. Mulder, M. M. Van Paassen, and S. De Groot, “Driving simulator fidelity and training effectiveness,” in *Proceedings of the 26th European Annual Conference on Human Decision Making and Manual Control*, Lyngby, Denmark, pp. 20-22, 2007.

[18] D. A. Noyce and M. V. Chitturi. (2018, February). Virtual road safety audits: Recommended procedures for using driving simulation and technology to expand existing practices. *Repository & Open Science Access Portal* [Online] pp. 1-47. Available: https://rosap.ntl.bts.gov/view/dot/35567.

[19] K. Santiago-Chaparro, M. DeAmico, A. Bill, M. Chitturi, and D. Noyce, “Realistic-scenario creation process for virtual road safety audits,” *Advances in Transportation Studies*, vol. 2011, pp. 19-28, Dec. 2011.

[20] J. F. Dols, J. Molina, F. J. Camacho, J. Marín-Morales, A. M. Pérez-Zuriaga, and A. Garcia, “Design and development of driving simulator scenarios for road validation studies,” *Transportation Research Procedia*, vol. 18, no. 2016, pp. 289-296, Jun. 2016.

[21] D. Nabors and J. Soika. (2013, July). Road safety audit case studies: Using three-dimensional design visualization in the road safety audit process. [Online]. Available: https://safety.fhwa.dot.gov/rsa/case_studies/fhwasa14003/fhwasa14003.pdf.

[22] J. Cremer, J. Kearney, Y. Papelis, and R. Romano, “The software architecture for scenario control in the Iowa

driving simulator," *Proceedings of the 4th Computer Generated Forces and Behavioral Representation*, Orlando, FL, pp. 1-9, 1994.

[23] J. Cremer, J. Kearney, and Y. Papelis, "Driving simulation: challenges for VR technology," *IEEE Computer Graphics and Applications*, vol. 16, no. 1, pp. 16-20, Sep. 1996.

[24] S. Bayarri, M. Fernandez, and M. Perez, "Virtual reality for driving simulation," *Communications of the ACM*, vol. 39, no. 5, pp. 72-76, May. 1996.

[25] S. Sahami and T. Sayed, "How drivers adapt to drive in driving simulator, and what is the impact of practice scenario on the research?," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 16, no. 2013, pp. 41-52, Jan. 2013.

[26] Seoul. Public open API [Internet]. Available: http://data.seoul.go.kr/dataList/OA-13291/A/1/datasetView.do.

[27] T. Eiter, H. Füreder, F. Kasslatter, J. X. Parreira, and P. Schneider, "Towards a semantically enriched local dynamic map," *International Journal of Intelligent Transportation Systems Research*, vol. 17, no. 1, pp. 32-48, Mar. 2019.

[28] ISO. ISO/TS 18750:2015: Intelligent Transport Systems - Cooperative Systems - Definition of a Global Concept for Local Dynamic Maps [Internet]. Available: https://www.iso.org/standard/63273.html.

[29] ETSI. ETSI TR 102 863 (V1.1.1): Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and Guidance on Standardization [Internet]. Available: https://www.etsi.org/deliver/etsi_tr/102800_102899/102863/01.01.01_60/tr_102863v010101p.pdf.

[30] ETSI. ETSI EN 302 895 (V1.1.0): Intelligent transport systems - Extension of map database specifications for Local Dynamic Map for applications of Cooperative ITS [Internet]. Available: https://www.etsi.org/deliver/etsi_en/302800_302899/302895/01.01.01_60/en_302895v010101p.pdf.

[31] H. Shimada, A. Yamaguchi, H. Takada, and K. Sato, "Implementation and evaluation of local dynamic map in safety driving systems," *Journal of Transportation Technologies*, vol. 5, no. 2, pp. 102-112, Mar. 2015.

[32] A. Imawan and J. Kwon, "A timeline visualization system for road traffic big data," in *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara: CA, pp. 2928-2929, 2015.

[33] B. Cao, Z. Yang, and H. Zhu, "Fusion and integration framework study on road network and dynamic traffic data in vehicle navigation," in *2009 Third International Symposium on Intelligent Information Technology Application Workshops*, Nanchang, China, pp. 70-73. 2009.

[34] E. Paikari, M. Moshirpour, R. Alhajj, and B. H. Far, "Data integration and clustering for real time crash prediction," in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, Redwood City: CA, pp. 537-544, 2014.

[35] El Abdallaoui, H. El Alaoui, A. El Fazziki, F. Z. Ennaji, and M. Sadgal, "A system for collecting and analyzing road accidents big data," in *2019 15th International Conference on Signal-Image Technology &Internet-Based Systems (SITIS)*, Sorrento, Italy, pp. 663-671, 2019.

[36] Python Software Foundation. Python [Internet]. Available: https://www.python.org/.

[37] C. Alberca, S. Pastrana, G. Suarez-Tangil, and P. Palmieri, "Security analysis and exploitation of arduino devices in the internet of things," in *Proceedings of the ACM International Conference on Computing Frontiers*, New York: NY, pp. 437-442, 2016.

[38] M. Bogdan, "Traffic light using arduino uno and LabVIEW," in *Proceedings of the 12th International Conference on Virtual Learning ICVL*, Bucharest, Romania, pp. 286-290, 2017.

[39] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. El Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25-32, Jan. 2021.

[40] A. Nayyar and V. Puri, "A review of Arduino board's, lilypad's & arduino shields," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 1485-1492, 2016.

[41] V. M. Cvjetković and U. Stanković, "Arduino based physics and engineering remote laboratory," in *International Conference on Interactive Collaborative Learning*, Belfast, UK, pp. 560-574, 2016.

[42] Y. Lin, Y. Lin, M. Yang, and J. Lin, "ArduTalk: An arduino network application development platform based on IoTtalk," *IEEE Systems Journal*, vol. 13, no. 1, pp. 468-476, Nov. 2017.

[43] A. Kurniawan. (2015, March). The hands-on arduino yún manual lab. *PE Press* [Online]. pp. 1-93. Available: https://play.google.com/store/books/details?id=YDcABwAAQBAJ.

[44] Korea Meteorological Administration. Weather AWS [Internet]. Available: https://www.weather.go.kr/m/obs/aws.jsp.

[45] X. Li, X. Yan, and S. C. Wong, "Effects of fog, driver experience and gender on driving behavior on s-curved road

segments," *Accident Analysis &Prevention*, vol. 77, no. 1, pp. 91-104, Apr. 2015.

[46] Q. Shangguan, T. Fu, and S. Liu, "Investigating rear-end collision avoidance behavior under varied foggy weather conditions: A study using advanced driving simulator and survival analysis," *Accident Analysis & Prevention*, vol. 139, no. 1, pp. 1-14, May. 2020.

[47] C. Chen, X. Zhao, H. Liu, G. Ren, Y. Zhang, and X. Liu, "Assessing the influence of adverse weather on traffic flow characteristics using a driving simulator and VISSIM," *Sustainability*, vol. 11, no. 3, pp. 1-16, Feb. 2019.

[48] G. Yang, M. M. Ahmed, and S. Gaweesh, "Impact of variable speed limit in a connected vehicle environment on truck driver behavior under adverse weather conditions: driving simulator study," *Transportation Research Record*, vol. 2673, no. 7, pp. 132-142, Apr. 2019.

[49] H. Tu, Z. Li, H. Li, K. Zhang, and L. Sun, "Driving simulator fidelity and emergency driving behavior," *Transportation Research Record*, vol. 2518, no. 1, pp. 113-121, Apr. 2015.

[50] M. Sarvi, M. Kuwahara, and A. Ceder, "Freeway ramp merging phenomena in congested traffic using simulation combined with a driving simulator," *Computer Aided Civil and Infrastructure Engineering*, vol. 19, no. 5, pp. 351-363, Jun. 2004.

[51] I. Kim, G. S. Larue, L. Ferreira, A. Rakotonirainy, and K. Shaaban, "Traffic safety at road - rail level crossings using a driving simulator and traffic simulation," *Transportation Research Record*, vol. 2476, no. 1, pp. 109-118, Jan. 2015.

[52] X. Zou, S. O'Hern, B. Ens, S. Coxon, P. Mater, R. Chow, M. Neylan, and H. L. Vu, "On-road virtual reality autonomous vehicle (VRAV) simulator: An empirical study on user experience," *Transportation Research Part C: Emerging Technologies*, vol. 126, no. 1, May. 2021.

권준호(Joonho Gwon)

2019년 서울시립대 컴퓨터과학부 학부생
※관심분야 : 시뮬레이션, AI설계, 센서개발

전연수(Yeonsoo Jun)

2017년 숭실대학교 법학과/경영학과 법학사/경영학사
2019년 서울시립대 국제도시과학대학원 국제도시과학석사
※관심분야 : 도시개발. 데이터 분석, 시뮬레이터

염춘호(Chunho Yeom)

1998년 동아대학교 토목공학과 공학사
1998년~2018년 한국도로공사 차장/책임연구원
2006년 노쓰캐롤라이나주립대 토목공학과 공학석사
2015년 노쓰캐롤라이나주립대 토목공학과 공학박사
2018년~ 서울시립대 국제도시과학대학원 교수
※관심분야 : 도로안전, 도시개발, 교통분석