Sensors and Materials, Vol. 34, No. 12 (2022) 4813–4825
MYU Tokyo

S & M 3145

Development of System for Collecting User-specified Training Data for Autonomous Driving Based on Virtual Road Environment

Min-Soo Kim[1*] and In-Sung Jang[2]

[1]Department of Computer Engineering, Daejeon University, 62 Daehak-ro, Dong-gu, Daejeon 34520, Korea
[2]Intelligent Convergence Research Laboratory, ETRI, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea

(Received April 30, 2022; accepted July 5, 2022; online published September 20, 2022)

Keywords: autonomous driving, high definition road, virtual environment, training data, deep learning

Deep learning technologies that use road images to recognize autonomous driving environments have been actively developed. Such deep-learning-based autonomous driving technologies need a large amount of training data that can represent various road, traffic, and weather environments. However, there have been many difficulties in terms of time and cost in collecting training data that can represent various road environments. Therefore, in this study, we attempt to build a virtual road environment and develop a system for collecting training data based on the virtual environment. To build a virtual environment identical to the real world, we convert and use two kinds of existing geospatial data: high-definition 3D buildings and high-definition roads. We also develop a system for collecting training data running in the virtual environment. The implementation results of the proposed system show that it is possible to build a virtual environment identical to the real world and to collect specific training data quickly and at any time from the virtual environment with various user-specified settings.

1. Introduction

Deep learning technologies that use road images as training data to recognize autonomous driving environments have been actively developed. Such deep learning technologies can detect and classify not only static environmental information such as lanes, traffic signs, and traffic signals, but also dynamic environmental information such as movements of vehicles and pedestrians.[1–5] Recently, deep learning technologies that use not only a camera, but also a light detection and ranging (LiDAR) sensor have made it possible to realize autonomous driving technologies that exceed human cognitive abilities. Such deep-learning-based autonomous driving technologies need a large amount of training data that can represent various road, traffic, and weather environments. However, there have been many difficulties in terms of time and cost in collecting training data using a vehicle equipped with a camera, a GPS, and a LiDAR sensor on real roads. It has also been difficult to collect various types of training data based on a variety of user-defined scenarios.

*Corresponding author: e-mail: minsoo@dju.ac.kr
https://doi.org/10.18494/SAM3966

ISSN 0914-4935

Therefore, in this study, we develop a method that can build a virtual road environment identical to the real world to collect various types of training data. We build the virtual road environment using two kinds of existing geospatial data: high-definition (HD) 3D buildings and HD roads. Specifically, we use the 3D building data of VWorld(6) and the HD road data of National Geographic Information Institute (NGII).(7) In the process of building the virtual road environment, we first extract the 3D building data from VWorld in Filmbox (FBX) format and directly import the data into our virtual environment. Second, we convert the HD road data of NGII into a new data model of OpenDRIVE(8) and import the data. We also develop a system for collecting training data based on the virtual environment. This system can easily collect training data for roads with various types of environments, sensors, and scenarios. We implement the virtual road environment and the system for collecting training data by modifying and extending an open-source autonomous driving simulator called Carla. Carla supports the OpenDRIVE and OpenSCENARIO(9) specifications for representations of the HD road data and the driving scenarios, respectively. The implementation results show that our method can build a virtual environment identical to the real world and that the data collection system can collect specific training data under various types of user-specified scenario at any time using the virtual environment.

The remaining sections of this paper are structured as follows. We review related works regarding the HD road data model and the autonomous driving simulator in Sect. 2. In Sect. 3, we present the method used to build a virtual road environment identical to the real world by converting and integrating the 3D building data of VWorld and the HD road data of NGII. In Sect. 4, we present the implementation details and results of our data collection system that can collect various types of training data. In Sect. 5, we conclude this work.

2. Related Works

Recently, with the increasing importance of the role of HD road data in autonomous driving, various types of HD road data model such as ISO 14296,(10) GDF5.1,(11) NDS open lane model,(12) OpenDRIVE, HERE HD live road,(13) and NGII HD road have been announced. In addition, there have been many studies in which these data models have been compared, improved, and applied to autonomous driving. Poggenhans *et al.*(14) presented the HD road data model Lanelet2, which can provide accurate situation information about vehicles and pedestrians. Lanelet2 defines attribute, location, and topology data for each lane on the road. Liu *et al.*(15) presented a brief review of navigation maps and then an extensive review of the development of HD road models for autonomous driving, focusing on the structure, functionality, accuracy, and standardization requirements of HD maps. Hirabayashi *et al.*(16) presented a reliable method of recognizing the state of traffic lights from images and an HD 3D map based a self-localization method to enable autonomous driving. Kim(17) presented a reference data model for HD roads consisting of five objects (roads, lanes, intersections, road signs, and road facilities) for autonomous driving and performed an analysis for conformance with ISO 14296, NGII HD road, and OpenDRIVE using the reference data model. Kang and Magdy(18) presented an HD road data model consisting of two parts: road network objects and landmark objects. This model is an extension of the node-link model of existing navigation maps to enable efficient autonomous driving. Specifically, the road network object is composed of a set of each lane, and the landmark object is composed of road facilities such as road signs, traffic lights, and buildings.

In most of the studies related to HD roads, the data models have been compared, improved, and standardized for application to efficient autonomous driving. In contrast, we present a method of building a virtual road environment for autonomous driving simulation using HD road data. In particular, we present a method of converting NGII HD roads into the OpenDRIVE data to build a virtual road environment.

In addition, there have been many studies on autonomous driving simulation using HD road data. Na *et al.*[19] presented a method of building an autonomous driving simulation environment by building OpenDRIVE data using RoadRunner software and simply applying the data to the Carla[20] simulator. Yao *et al.*[21] presented a virtual-reality-based test platform for autonomous driving built with AirSim[22] software and the UE4 engine to reduce the cost and risk of road testing for autonomous driving. The platform establishes a model library that includes a vehicle dynamics model, a sensor model, and a traffic environment model. Motta *et al.*[23] presented a system for generating synthetic datasets including driving scenarios that were based on OpenSCENARIO, a well-established and vendor-independent standard. The user can specify parameter values, such as vehicle speed, distance, and weather condition, which represent the intended variability within a scenario through the system configuration. Rong *et al.*[24] proposed an LGSVL simulator for autonomous driving that allows a user to easily customize sensors, create new types of controllable objects, and create digital twins of particular environments. Osiński *et al.*[25] applied a reinforcement learning result for autonomous driving based on a simulation environment using virtual synthetic data to a real-world vehicle. Amini *et al.*[26] proposed a data-driven simulation and training system capable of reinforcement learning of end-to-end autonomous vehicle control policies using Carla. They argued that the system is scalable, leverages reinforcement learning, and is applicable to situations requiring effective perception and robust operation in the real world.

In most of the autonomous driving simulation studies, a virtual environment was built by applying 3D geospatial data or HD roads to Carla or the AirSim simulator, and autonomous driving technologies based on deep learning that are applicable to real-world vehicles were developed. In this study, we implement a simulation system identical to the real world that applies both HD road and 3D building data to Carla. In particular, we implement a system that can collect various types of road image with user-specified driving environments and scenarios such as time, weather, vehicles, and pedestrians as well as sensor settings.

3. Building Virtual Road Environment Identical to Real World

In this study, we use existing HD geospatial data and an open-source simulator for autonomous driving to build a virtual road environment identical to the real world. Figure 1 shows the process of building a virtual road environment, consisting of data acquisition, data model conversion, and virtual environment creation.

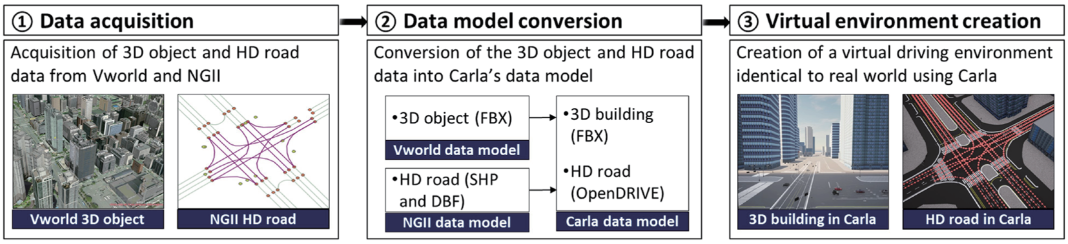

Fig. 1. (Color online) Process of building a virtual road environment identical to the real world using existing HD building and road data and the autonomous driving simulator Carla.

As shown in Fig. 1, we first collect the HD 3D building data and HD road data separately. Specifically, we collect the 3D building data of VWorld and the HD road data of NGII that are easily accessible in Korea. Second, we perform a data model conversion to apply the raw data from VWorld and NGII to our virtual road environment. In this study, since we use the autonomous driving simulator Carla, we should convert the raw data into a data model available in Carla. Carla uses an FBX data model for 3D building data and a de facto standard model called OpenDRIVE for HD road data. We can import the 3D building data in the FBX model from VWorld without modifying the data, but we should convert the HD road data of NGII composed of SHAPE and base database file (DBF) into OpenDRIVE data. Third, we build a virtual road environment identical to the real world by integrating and visualizing the two datasets. In Sect. 3.1, we present the design and implementation details of the data conversion, and in Sect. 3.2, we present the result of a virtual road environment based on Carla.

3.1 Design and implementation of NGII data conversion

VWorld provides realistic 3D data including buildings, road facilities, and road marks in the form of an FBX model that can be used in Carla. However, NGII does not support the OpenDRIVE data model, which can be used in Carla for HD road data. Therefore, in this study, we convert the NGII HD road data into OpenDRIVE data. In this section, we describe the conversion process of the NGII HD road data into OpenDRIVE data in detail.

The NGII HD road data model extends the node-link model used in existing car navigation systems. Therefore, it defines a node-link layer for each lane and adds new layers for various types of road facility and road structure. Table 1 shows the NGII HD road data model configuration, which consists of 14 layers representing road structures, road facilities, road signs, and road marks in addition to nodes and links. From Table 1, we can see that only A1_Node and A2_Link layers are used for actual connections between roads and intersections for a vehicle driving. We can also see that the NGII HD road data model does not define a road layer and an intersection layer as distinct. In contrast, the OpenDRIVE data model defines a road and an intersection as distinct classes. Then it hierarchically defines road structures, road facilities, road signs, and road marks as subclasses of the Road and Junction classes. From Table 2, we can see that the Road class defines subclasses of Planview, Elevationprofile, Lateralprofile, Lanes, Objects, and Signals to represent various road information. We can also see that the Junction class defines subclasses of Connection and Priority to represent intersection information.

Table 1
NGII HD road data model composed of 14 layers of road structures, road facilities, road signs, and road marks, in addition to nodes and links.

Layer name	Classification	Features
A1_Node (A1)	Road lane	Specific points located in center in lateral direction of roads (entry, exit, and stop points of roads, tunnels, bridges, underpasses, tollgates, etc.)
A2_Link (A2)	Road lane	Links between A1_Nodes
A3_Drivewaysection (A3)	Road structures	Road structures such as tunnels, bridges, underpasses, etc.
A4_Subsidlarysection (A4)	Road structures	Road structures such as rest areas
A5_Parkinglot (A5)	Road structures	Road structures such as parking lots
B1_Safetysign (B1)	Road sign	Road signs
B2_Surfacelinemark (B2)	Road mark	Road surface markings
B3_Surfacemark (B3)	Road mark	Road surface markings
C1_Trafficlight (C1)	Road facilities	Traffic lights
C2_Kilopost (C2)	Road sign	Road signs indicating distances on highway
C3_Vehicleprotectionsafety (C3)	Road facilities	Road safety facilities such as central dividers, guard rails, concrete barriers
C4_Speedbump (C4)	Road facilities	Road safety facilities such as speed bumps
C5_Heightbarrier (C5)	Road facilities	Road facilities with height such as overpasses and bridges
C6_Postpoint (C6)	Road facilities	Road facilities such as traffic light poles and road sign poles

Table 2
OpenDRIVE data model composed of Road and Junction classes.

Class	Subclass	Features	NGII layers
Road	Type	Road type such as rural, town, motorway, pedestrian, bicycle, low speed, etc.	Not defined
	Planview	Road reference line (road center line)	A1 and A2
	Elevationprofile	Road elevation in direction of reference line	Not defined
	Lateralprofile	Road surface perpendicular to reference line	Not defined
	Lanes	Lanes with link, width, boundary, road mark, material, speed limit and rule information	A1, A2, B2, and B3
	Signals	Road signs and traffic signals	B1 and C1
	Objects	Road structures or road facilities such as poles, trees, barriers, parking lots, crosswalks, street lights, etc.	A3, A4, A5, C2, C3, C4, C5, and C6
Junction	Connection	Actual links between roads in junction	A1 and A2
	Priority	Priority of links at junction	Not defined

It is impossible to create all classes in the OpenDRIVE model from the NGII HD road data model. This is because the OpenDRIVE model has a more detailed definition using the hierarchical definition of subclasses than the NGII HD road model. Moreover, it is not easy to convert layers of the NGII HD road data based on a relational model to classes of the OpenDRIVE data based on an object-oriented model. Therefore, in this study, we perform the NGII HD road-to-OpenDRIVE data conversion at a level that can build a Carla-based virtual road environment. Specifically, we create the Road and Junction classes of the OpenDRIVE model by converting only the layers for roads and intersections required for vehicle driving from the NGII HD road data.

Figure 2 shows the NGII HD road-to-OpenDRIVE data conversion system composed of an Import module, a Conversion module, and an Export module.

The Import module, which consists of the NGII Import and Correction submodules, imports the HD road data from NGII and corrects errors in the data. The NGII Import module imports two-layer files with layers A1 and A2 composed of geospatial data of SHAPE and attribute data of DBF, and forms A1 and A2 objects. However, sometimes there are errors in the lane numbers of the A2 layer. The Correction module checks and corrects these errors before executing the Conversion module. Specifically, the Correction module finds and corrects the case where the lane number is incorrect by checking the RoadNo and LaneNo attributes of the A2 objects. Here, RoadNo is a number identifying a road and LaneNo is a number identifying a lane included in a road. Since LaneNo of a road always starts from 1, we can find roads whose LaneNo does not start from 1, and correct the incorrect LaneNos of the roads. Figure 3 shows examples of an incorrect LaneNo, the correction of the incorrect LaneNo, and an actual image of the lanes. Figure 3(a) suggests that the road includes five lanes with LaneNos from 1 to 5. However, when we check all RoadNos of the five lanes, we see that the RoadNos are set to 1, 2, and 3 for each lane as shown in Fig. 3(b). Therefore, in this study, we correct the LaneNos of {1, 2, 3, 4, 5} to {1}, {1, 2}, and {1, 2} with the LaneNos for each road of 1, 2, and 3. Finally, we can confirm that the five lanes are included in three different roads in the actual image in Fig. 3(c). In addition, the Correction module finds and corrects the A2 objects having incorrect LaneNos by comparing L_LinkID and R_LinkID of the A2 objects, where L_LinkID and R_LinkID store the LaneNos of the left and right lanes of an A2 object, respectively.

The Conversion module, which consists of the Road and Junction submodules, creates the Road and Junction objects of the OpenDRIVE model from the A1 and A2 layers of the NGII model. The Road module first creates Planview objects, then creates Lanes objects, and finally creates Road objects, which include Planview and Lanes objects. Regarding Planview object creation, the OpenDRIVE model defines a Planview object as a road reference line indicating the absolute coordinates of a road centerline, then it defines other objects such as Lanes, Signals, and Objects using the relative coordinates of the Planview object. Therefore, the OpenDRIVE model must create the Planview object, but the NGII HD road does not define any layers corresponding to the Planview object. In this study, we propose a method of creating Planview objects from the A2 layer. The proposed method finds the A2 objects corresponding to the first lane in the A2 layer, performs a linear transformation on them to a road centerline, and finally

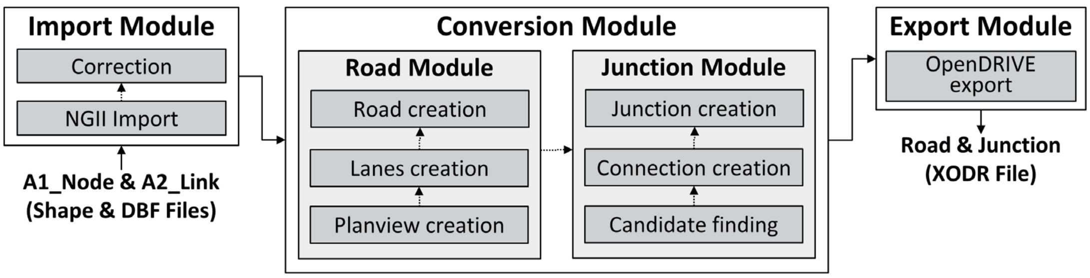

Fig. 2. Configuration of the NGII HD road-to-OpenDRIVE data conversion system.

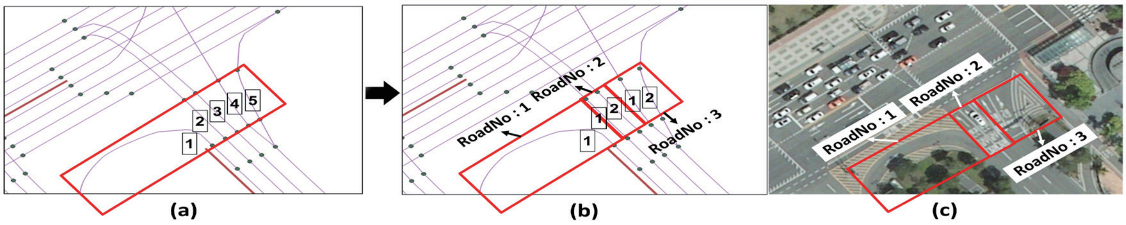

Fig. 3. (Color online) Correction process of incorrect LaneNo: (a) incorrect LaneNos setting, such as five lanes included in one road, (b) LaneNo correction based on five lanes included in three different roads, and (c) actual image of the lanes.

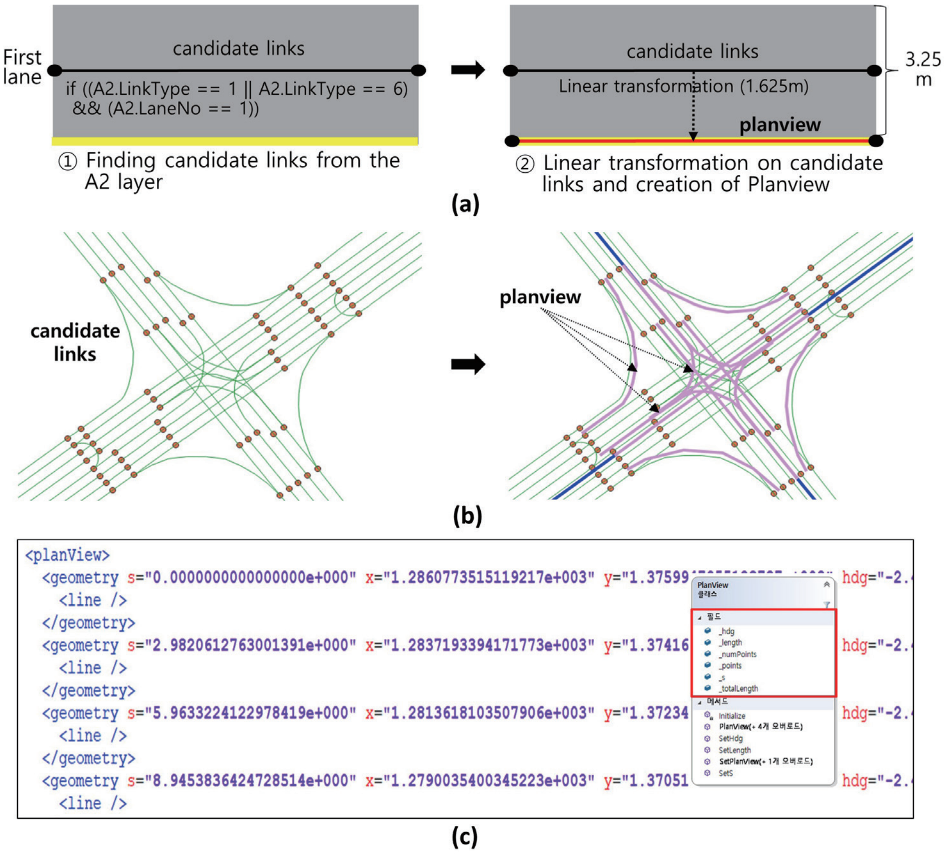

Fig. 4. (Color online) Creation process of Planview objects: (a) example of creation of Planview object from A2 object, (b) visualization of Planview objects, and (c) Planview objects represented by the OpenDRIVE model.

forms the Planview objects. Specifically, the method creates candidate links by finding the A2 objects representing the first lane on roads and intersections, i.e., LinkType is 1 or 6 and LaneNo is 1, as shown Fig. 4(a). After that, the method creates Planview objects by linearly transforming the candidate links into a road centerline by 1.625 m, considering that the downtown road width is 3.25 m. We can create Planview objects in various types of straight-line set, spiral, arc, and cubic polynomial, but we create them as a straight-line set for simplicity in this study. Figure

4(b) shows the Planview objects created from the candidate links, and Fig. 4(c) shows the Planview objects in an XML format.

After the creation of the Planview object, we create the Lanes object. Regarding Lanes object creation, the OpenDRIVE model defines the Lanes object hierarchically, which includes a large amount of information such as the lane width, lane boundary, road mark, speed limit, and surface material as well as the lane link. In particular, the Lanes object should include the lane width along with the lane link, but the A2 object of the NGII model does not include lane width information. Therefore, in this study, we create the lane width using the A2 object. We first find all A2 objects located in the center of the lane on roads and intersections, i.e., LinkType is 1 or 6. Then, we linearly transform them to the left side of each lane by 1.625 m, as shown in Fig. 5(a), and create the Lanes objects using the transformed A2 objects. Finally, we set the lane width of the Lanes objects to 3.25 m, as shown in Fig. 5(b).

After the creation of the Planview and Lanes objects, we finally create a Road object that includes them. In this study, we create a Road object containing the minimal amount of information enabling a vehicle driving simulation. Therefore, we exclude ElevationProfile and LateralProfile, which are not defined in the NGII HD roads, and Signals and Objects, which define a different model from that of NGII HD roads.

The Junction module first creates Connection objects and finally creates Junction objects, which include Connection objects. As shown in Table 2, the Connection class defines links within an intersection and the Junction class defines a road intersection, which contains the Connection object. Such Connection and Junction objects are essential in the OpenDRIVE model, but the NGII HD roads do not define a specific layer for the road intersections. The NGII HD roads only define whether the A2 objects are links within any intersections. Therefore, in this study, we propose a method to create the Connection and Junction objects using the A2 objects. The proposed method finds all A2 objects located within all road intersections from the A2 layer and groups the objects by each intersection. Specifically, the method finds all A2 objects representing lanes within the road intersections, i.e., LinkType is 1, and groups the A2 objects by each intersection using the connection information among the objects. It finally creates the Connection objects including the A2 objects included in each group. In this study, we found 321 A2 objects representing lanes within road intersections from the NGII HD road of Yeouido area and created 36 Junction objects from the A2 objects as shown in Fig. 6.

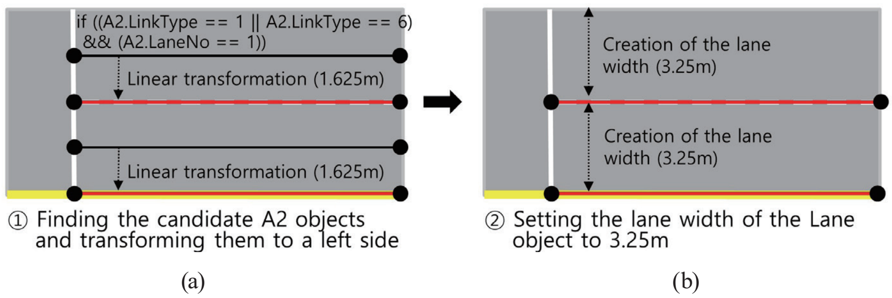

Fig. 5. (Color online) Creation process of the Lanes object: (a) finding and linear transformation of the candidate A2 objects and (b) creation of the Lanes object having the lane width.

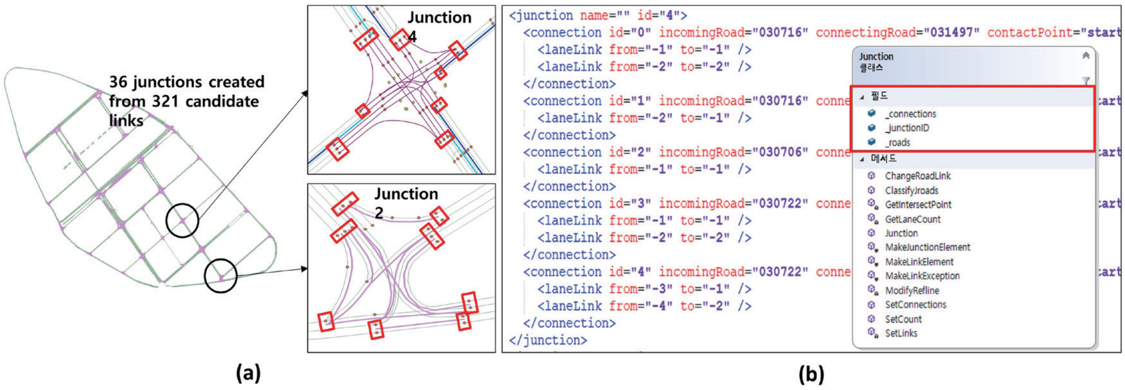

Fig. 6. (Color online) Creation process of the Junction object: (a) example of creation of the Junction object of Yeouido area and (b) Junction object represented by the OpenDRIVE model.

Finally, the Export module exports an XODR file of the OpenDRIVE model, such as the models in Figs. 4(c) and 6(b), for the Road and Junction objects created in this Conversion module.

3.2 Virtual road environment based on Carla

In this section, we show the implementation results of the NGII HD road-to-OpenDRIVE data conversion system of Sect. 3.1. The implementation results show the generation of a virtual road environment identical to the real world by converting and integrating the NGII HD road data and the VWorld 3D building data of Yeouido area. Figure 7 shows an example of the virtual road environment built on the basis of the Carla simulator.

Figure 7(a) shows the visualization result of the entire Yeouido area. Figures 7(b) and 7(c) respectively show the visualization result of the Road and Junction objects having texture. In the process of building the virtual road environment, we transformed the WGS84 coordinate system in the FBX data to UTM-K and modified its texture scale for realistic rendering. Nevertheless, Fig. 7 appears to be considerably different from the real world. This is because the scenes and the textures of the original FBX data, such as buildings, road facilities, road markings, and crosswalks, are out of date or the quality of the original data is low. Such problems can be solved by updating the original data.

4. System for Collecting Training Data

In this section, we present a system for collecting training data based on the virtual road environment described in Sect. 3.2. This system can collect AI training data for roads with various types of environments, sensors, and scenarios. As shown in Fig. 8, this system is composed of an environment module, a sensor module, and a scenario module. The data collection process is performed in the order of environment setting, sensor setting, and scenario setting.

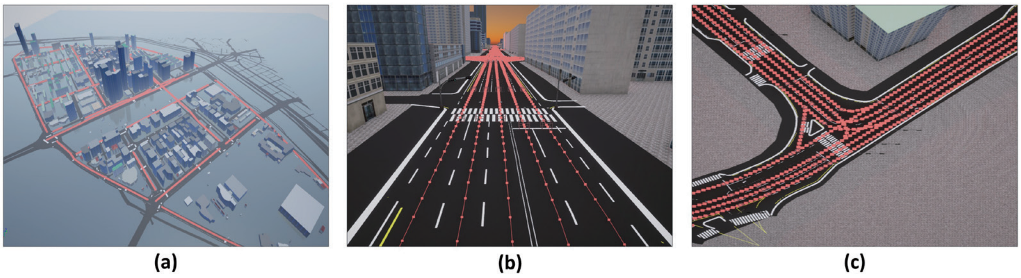

Fig. 7. (Color online) Example of the virtual driving environment for Yeouido area: (a) visualization of the entire Yeouido area, (b) visualization of the Road objects, and (c) visualization of the Junction objects.

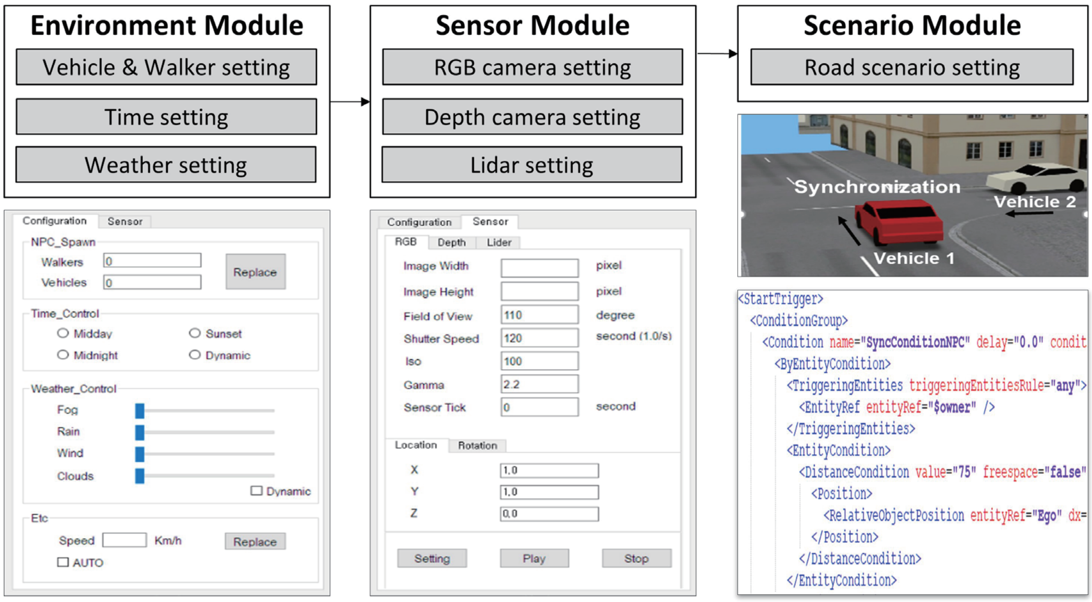

Fig. 8. (Color online) Configuration and implementation of the system for collecting training data.

The environment module sets several road environments such as time, weather, and non-player characters (NPCs). It can set the NPCs of vehicles and pedestrians in random locations; set the time to morning, midday, sunset, and night; and set the weather to fog, rain, wind, and clouds. It enables the user to set various types of virtual road environment identical to the real world when collecting various types of training data.

The sensor module employs an RGB camera, a depth camera, and a LiDAR sensor to collect various types of road image. It can set an installation location using 3D coordinates (x, y, z) and gyroscope values (roll, pitch, yaw). In addition, it can set options of the image size, field of view (FOV), shutter speed, and sensitivity of the RGB and depth cameras, and can set options of the detection range, number of point clouds, and FOV of the LiDAR sensor. It enables the user to collect various types of training data of RGB images, depth images, and point clouds.

The scenario module sets routing of the data collection vehicle and the driving scenarios among vehicles. It can set the routing of the data collection vehicle using the starting point, a waypoint, and the destination. It also can set various driving scenarios such as overtaking, cutting in, lane changes, speed changes, and intersection synchronization among vehicles. Such scenario setting enables the user to collect road images assuming specific traffic situations. In particular, we used the OpenSCENARIO standard for the standardized scenario setting in our system. From the XML-based OpenSCENARIO, we can easily set various actions, such as route changes, lateral movement, longitudinal movement, vehicle synchronization, and traffic signal changes, and various events using the speed, distance, and time of vehicles.

Figure 9 shows examples of executing the system for collecting training data. Figures 9(a)–9(c) show simulation results of Yeouido built as a virtual road environment. Figure 9(a) shows a virtual road environment of Yeouido composed of HD roads, 3D buildings, and NPC vehicles. In particular, we can see the 3D and 2D simulation results for the same area at the same time as respectively shown in Figs. 9(b) and 9(c). Using the image in Fig. 9(c), we can also dynamically move our data collection vehicle to other locations to collect training data in other areas. Figures 9(d)–9(f) show examples of various types of road image collected using various driving environments and sensors. Specifically, Fig. 9(d) shows a road image obtained from the RGB sensor at midday, and Fig. 9(e) shows the road image in Fig. 9(d) after changing the environment from midday to late afternoon. Figure 9(f) shows an example of point cloud data obtained using the LiDAR sensor rather than the RGB sensor.

As shown in Fig. 9, our proposed system can collect various types of training data by using the virtual road environment identical to the real world built with the VWorld 3D data and the NGII road data. Therefore, we can collect AI training data quickly and at a low cost at any time provided the VWorld 3D Data and the NGII road data for a specific region are available.

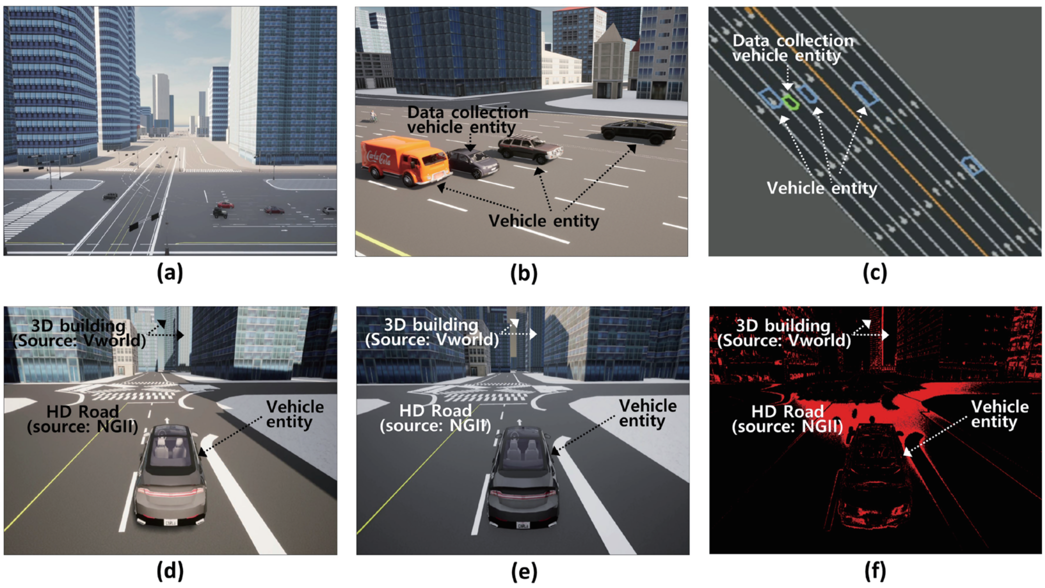

Fig. 9. (Color online) Examples of executing the system for collecting training data: (a) virtual road environment of Yeouido, (b) 3D simulation result around the data collection vehicle, (c) 2D simulation result around the data collection vehicle, (d) road image collected using RGB sensor, (e) road image collected using RGB sensor by changing time to late afternoon, and (f) point cloud data collected using LiDAR sensor.

Moreover, we can collect specific training data by setting various user-specified environments of roads, sensors, and scenarios.

5. Conclusions

In this study, we built a virtual road environment and developed a system for collecting training data based on the virtual environment. To build a virtual road environment identical to the real world, we integrated both NGII HD road data and VWorld 3D building data and developed an NGII HD road-to-OpenDRIVE data conversion system. We also developed a system for collecting training data composed of an environment module, a sensor module, and a scenario module running in the virtual environment. The implementation results of the NGII HD road-to-OpenDRIVE data conversion system showed that it is possible to build a virtual road environment identical to the real world. The implementation results of the data collection system showed that it is possible to collect specific AI training data quickly and at a low cost at any time from the virtual road environment with various user-specified settings.

We plan to extend this work in two directions. First, we would like to create practically usable AI training data through labeling specific objects for our collected road images. We expect that various types of training data can be created using various road environments and scenarios. Second, we would like to validate the AI training data by applying the data to various deep learning models. Specifically, we would like to establish whether our training data is applicable to deep learning models for object detection or recognition, such as traffic signals and traffic signs, compared with datasets such as Common Objects in Context (COCO)[27] and Laboratory for Intelligent and Safe Automobiles (LISA) Datasets.[28] We expect that the training data collected in our virtual road environment can be used in a deep learning model.

Acknowledgments

This work was supported by Grant 22DRMS-C47287-05 from the Development of Customized Realistic Three-Dimensional Geospatial Information Update and Utilization Technology Based on Consumer Demand project, funded by the Ministry of Land, Infrastructure and Transport of the Korean government.

References

1 D. Kim, Y. Kang, Y. Park, N. Kim, and J. Lee: Spatial Inf. Res. **28** (2020) 241. https://doi.org/10.1007/s41324-019-00285-x
2 A. Hamoudzadeh and S. Behzadi: Spatial Inf. Res. **29** (2021) 379. https://doi.org/10.1007/s41324-020-00358-2
3 H. Lee and Y. Kang: Spatial Inf. Res. **29** (2021) 825. https://doi.org/10.1007/s41324-021-00397-3
4 T. Lee, M. Jeong, and A. Peter: Sens. Mater. **34** (2022) 251. https://doi.org/10.18494/SAM3732
5 X. Zhang, H. Huang, W. Meng, and D. Luo: Sens. Mater. **32** (2020) 4505. https://doi.org/10.18494/SAM.2020.3128
6 VWorld: https://map.vworld.kr/map/ws3dmap.do (accessed April 2022).
7 NGII HD road: https://map.ngii.go.kr/ms/map/NlipMap.do?tabGb=hdmap (accessed April 2022).
8 ASAM OpenDRIVE: https://www.asam.net/standards/detail/opendrive/ (accessed April 2022).
9 ASAM OpenSCENARIO: https://www.asam.net/standards/detail/openscenario/ (accessed April 2022).

10 ISO 14296:2016 Intelligent Transport Systems - Extension of Map Database Specifications for Applications of Cooperative ITS: https://www.iso.org/standard/54587.html (accessed April 2022).
11 ISO 20524-2:2020 Intelligent Transport Systems - Geographic Data Files (GDF) 5.1: https://www.iso.org/standard/72494.html (accessed April 2022).
12 NDS Open Lane Model: https://www.openlanemodel.org/ (accessed April 2022).
13 HERE HD Live Map: https://www.here.com/platform/HD-live-map (accessed April 2022).
14 F. Poggenhans, J. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr: Proc. 2018 Int. Conf. Intelligent Transportation Systems (ITSC 2018) 1672. https://doi.org/10.1109/ITSC.2018.8569929.
15 R. Liu, J. Wang and B. Zhang: J. Navig. **73** (2019) 324. https://doi.org/10.1017/S0373463319000638
16 M. Hirabayashi, A. Sujiwo, A. Monrroy, S. Kato, and M. Edahiro: Rob. Autonom. Syst. **111** (2019) 62. https://doi.org/10.1016/j.robot.2018.10.004
17 M. Kim: J. Geospatial Inf. Sci. **28** (2020) 79. https://doi.org/10.7319/kogsis.2020.28.4.079
18 Y. Kang and A. Magdy: Proc. 2020 IEEE Int. Conf. Data Engineering Workshops (ICDEW 2020) 26. https://doi.org/10.1109/ICDEW49219.2020.00-11
19 Y. Na, S. Kim, Y. Kim, J. Park, J. Jeong, K. Jo, S. Lee, S. Cho, M. Sunwoo, and J. Oh: KASE **28** (2020) 797. https://doi.org/10.7467/KSAE.2020.28.11.797
20 A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun: Proc. 2017 Conf. Robot Learning (CoRL 2017) 1. https://doi.org/10.48550/arXiv.1711.03938
21 S. Yao, J. Zhang, Z. Hu, Y. Wang, and X. Zhou: J. Eng. **16** (2018) 1768. https://doi.org/10.1049/joe.2018.8303
22 AirSim: https://microsoft.github.io/AirSim/ (accessed April 2022).
23 J. Motta, F. Bellotti, R. Berta, A. Capello, M. Cossu, A. D. Gloria, L. Lazzaroni, and S. Bonora: LNEE **866** (2022) 310. https://doi.org/10.1007/978-3-030-95498-7_43
24 G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafono, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim: Proc. 2020 IEEE Int. Conf. Intelligent Transportation Systems (ITSC 2020) 1. https://doi.org/10.1109/ITSC45102.2020.9294422
25 B. Osiński, A. Jakubowski, P. Zięcina, P. Miłoś, C. Galias, S. Homoceanu, and H. Michalewski: Proc. 2020 IEEE Int. Conf. Robotics and Automation (ICRA 2020) 6411. https://doi.org/10.1109/ICRA40945.2020.9196730
26 A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus: IEEE Rob. Autom. Lett. **5** (2020) 1143. https://doi.org/10.1109/LRA.2020.2966414
27 COCO : Common Objects in Context: https://cocodataset.org/#home (accessed April 2022).
28 LISA : Laboratory for Intelligent & Safe Automobiles Datasets: https://avindata.ca/dataset/lisa-laboratory-for-intelligent-safe-automobiles-datasets?activity_id=a16ffa0f-d5e3-4f5e-9b65-0cc32e01520e (accessed April 2022).