

Efficient Procedure of Building University Campus Models for Digital Twin Simulation

Talha Azfar¹, Jeffrey Weidner, Adeeba Raheem, Ruimin Ke², *Member, IEEE*,
and Ruey Long Cheu, *Senior Member, IEEE*

Abstract—Realistic digital geographical models of real-world locations are a necessary starting point for digital twin applications, especially for simulation and visualization. However, the visual fidelity of this first step is often neglected, since the effort involved is counterproductive to the main research focus. In this paper, we explore different tools and resources to assemble an efficient and convenient procedure to create a 3D digital model of a university campus that can support digital twin applications. Specifically, the terrain, buildings, and road network are combined into the CARLA project on Unreal Engine, enabling computer vision, traffic simulation, and autonomous driving experimentation.

Index Terms—Digital twin, 3D model, intelligent transportation systems, autonomous vehicles.

I. INTRODUCTION

DIGITAL twins (DT) are fast becoming popular across many disciplines as more and more applications are explored. Internet of Things (IoT) devices have simplified sensing, data acquisition, communication, and actuation in the digital twin pipeline [1]. While those are important components, much of the perceived value of a DT lies largely in the digital model for visualization and user interface. In manufacturing and robotics applications, digital models are highly precise representations of real world objects, with the fidelity of surrounding environment depending on the intended use case [2]. Infrastructure systems DT rely on building information modeling (BIM) for precise 3D structural plans, including utilities, to represent the form and function of an existing or proposed system [3], [4].

Urban infrastructure planning has seen a rise in DTs of real geographical locations as an important aspect of smart cities [5]. For example, Zürich has a detailed 4D digital twin of the city based on data recorded by the government about buildings and infrastructure historically, and future projections [6], allowing users to see a 3D replica of the city at various points

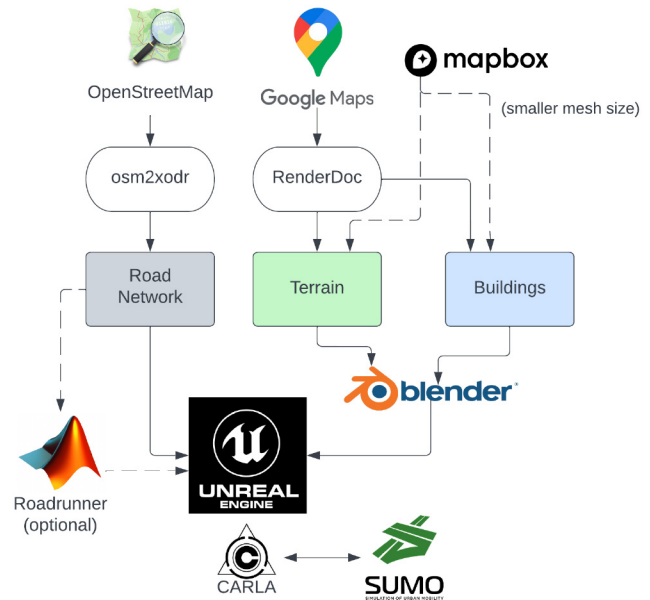


Fig. 1. Summary of the proposed procedure.

in time. The data is made freely available under the international open data charter [7], which is developed according to the “open by default” principle. The UK has also launched a national digital twin program for building, maintaining, and decommissioning infrastructure. It purports to provide benefits to society, economy, business, and the environment [8].

Transportation systems can benefit greatly from DT applications, but highly specialized digital modeling, lack of interoperable software, and virtually no standard sensing and simulation solutions create a wide gap for DT research to fill [9]. Current efforts to digitize transportation infrastructure are reliant on proprietary enterprise software using custom private data which makes collaboration, research, and prototyping difficult for individuals and academia. Some commercially available enterprise services include Microsoft Azure Digital Twins [10], Amazon Web Services IoT TwinMaker [11], PTC [12], Bentley iTwin [13], and Autodesk Tandem [14]. However, open source data, free software (including software that is free in universities), and online communities have created a few ways to get started with digital modeling and simulation.

Road networks are often simulated and analyzed in two dimensions. For example, highly customizable road network

Manuscript received 26 July 2022; accepted 26 September 2022. Date of publication 10 October 2022; date of current version 12 December 2022. This work was supported by the USDOT C2SMART University Transportation Center. (Corresponding author: Ruimin Ke.)

Talha Azfar is with the Electrical and Computer Engineering, University of Texas at El Paso, El Paso, TX 79968 USA (e-mail: tazfar@miners.utep.edu).

Jeffrey Weidner, Adeeba Raheem, Ruimin Ke, and Ruey Long Cheu are with the Department of Civil Engineering, University of Texas at El Paso, El Paso, TX 79968 USA (e-mail: jweidner@utep.edu; aaraheem@utep.edu; rke@utep.edu; rcheu@utep.edu).

Digital Object Identifier 10.1109/JRFID.2022.3212957

planning based on GIS data is presented in [15] but the representation of the virtual information is limited to 2D. A road infrastructure DT method is discussed in [16], with 360° video, object detection, and GPS as the data transmitted. There is no digital model in this twin, as the 360° video is used for visualization. Terrain and building 3D modeling is performed using a few different methods in [17] based on the kind of available data including point clouds, 3D object models, digital terrain model, and footprint + height of buildings. Aerial LiDAR data is used in [18] by modeling complex roof structures as a combination of predefined simple parametric shapes, assuming rectilinear outlines, and eliminating non-flat objects like trees. Detailed 3D models were used in a DT approach [19] involving the integration of augmented reality (AR) and drones, giving planners and decision makers an immersive aerial view. Tools like Robot Operating System in the Gazebo environment have been extensively used in DTs of manufacturing and warehouses to aid in the simulation of automated guided vehicles [20], which can also be applied to autonomous vehicles research. The examples discussed so far do not provide a clear and efficient way of creating a university campus DT without extensive 3D modeling or compromising visual fidelity.

In the subsequent section we discuss the major challenges for digital model of university campus, followed by an efficient procedure for simulating the existing transportation network of a real world location, the University of Texas at El Paso (UTEP) campus, and its 3D visualization using freely available data and software as a quick start base for transportation DT research. Then, we show the results with some comments on the practical requirements, followed by a conclusion highlighting future directions.

II. CHALLENGES FOR DIGITAL UNIVERSITY CAMPUS MODELS

It is often time consuming for researchers to create 3D digital models from scratch, and DT research focuses more on the simulation, data collection, and system analysis. Therefore, the efficient use of the most convenient tools which allow multiple avenues of research, while also maintaining faithful resemblance to the real world location is necessary for certain parts of research, especially presentation, visualization, and real world applications.

Compared to major urban roadways, the available information on university campus roads is incomplete, for, e.g., disconnected links on digital maps, as campus roads are not managed by a state or local transportation agency. Campus road information may also be outdated in digital model creation, such as having an incorrect number of lanes or one-way information. Besides, there are often not as many traffic sensors (both fixed sensors and mobile sensors) on a university campus as on an arterial road or freeway. This issue creates additional barriers to building an accurate campus digital model.

Many universities do have certain types of sensors and data logging, but it usually is not available in a usable format or the accessibility is protected due to privacy concerns. 3D data like LiDAR point clouds and CAD models are also

unavailable unless specific work has previously been done to create them. Generally, as-built BIM models are not obtainable for university buildings, as these are not commonly part of the deliverables. Although, the adoption of campus BIM is expected to grow in the coming years [21] due to increased demands of smart monitoring and sustainability solutions which can leverage DT technology.

It is challenging to have more elaborate digital campus models towards supporting a real-time full-scale campus DT. It would need even more detailed and unique information. For example, campus operation data (e.g., classroom schedule and occupancy) is necessary in order to integrate the major campus activities in the digital model; building entrances/exits and connection information would be helpful to understand campus-unique optimization problems and statistical analysis, such as to model the pedestrian volume and travel time.

III. PROPOSED PROCEDURE

While Google has one of the most up to date and accurate maps, the non-proprietary collaborative project OpenStreetMap (OSM) [22] uses data from freely available sources presented in a convenient manner with options to export the data in OSM files. These files are XML formatted, containing road network information in the form of nodes, ways, and relations, which can be used for multimodal microscopic transportation simulations in free software like SUMO or proprietary software like VISSIM. Some preprocessing needs to be performed since missing information like the number of lanes, or speed limits can be added or updated before simulation. The OSM file can be converted to a popular JSON based GIS format, General Modeling Network Specification (GMNS) using a tool called osm2gmns [23]. This allows visualization, editing, and simulations in free geospatial analysis software like QGIS, as shown in Fig. 2(a).

However, this approach only yields the road network with, at best, a satellite image of the area as background. The incorporation of contextual information like 3D buildings and terrain is usually manual and constrained by the capabilities of the software used. Mapbox [24] provides a service to export terrain and 3D buildings from OpenStreetMap. The buildings lack texture, which can be added in 3D graphics software like Blender [25], but are otherwise to scale and representative of the actual location. This 3D model can be improved by manually adjusting and editing in Blender, but a quick way of using preexisting 3D data is to extract it from Google maps.

RenderDoc [26] is a graphics debugging tool that can be used to store and examine 3D graphics data that an application is displaying, for example Google maps 3D view. It is recommended to zoom into the map to view the highest level of detail. These debugging captures can be saved to disk and imported to Blender using MapsModelsImporter plugin [27]. While the output of this process is relatively “low poly” buildings (low 3D resolution), and objects like trees and parked cars add to the noise in the 3D mesh, this is the most visually appealing result that does not involve manual 3D sculpting and texturing. It can be used as a starting point for a higher quality and smoother mesh if edited in Blender. It must be

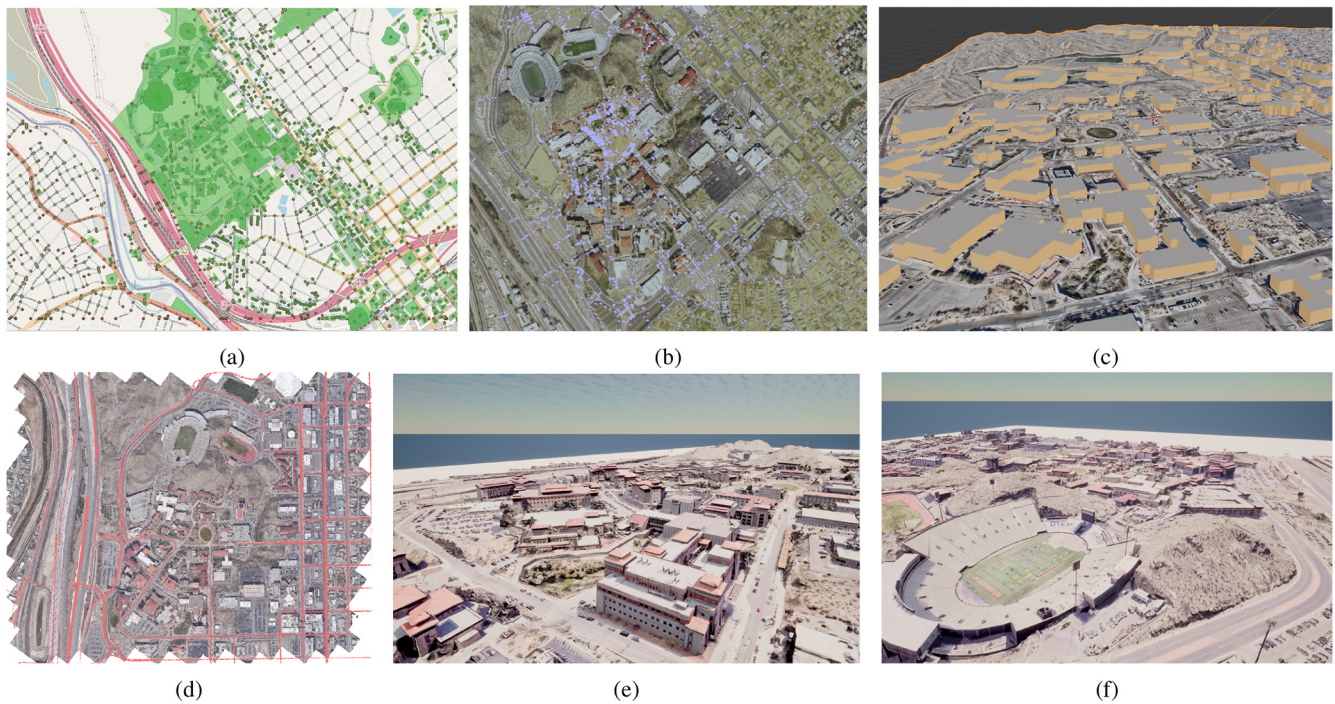


Fig. 2. (a) OpenStreetMap data converted with osm2gmns, viewed in QGIS, (b) OpenStreetMap imagery and road network imported to RoadRunner, (c) Mapbox terrain and 3D buildings based on OpenStreetMap in Blender, (d) Google maps 3D data and OpenStreetMap road network imported to Unreal Engine, (e)&(f) Overview of final CARLA map.

noted that data obtained from Google maps cannot be shared or used for commercial purposes.

IV. USE CASES

A. Road Network Customization

This model can be used to create a map in CARLA, a driving simulation software which runs on Unreal Engine [28]. The road network, traffic lights and signs, and road markings can be customized and applied in RoadRunner, a proprietary software from Mathworks [29], which provide a campus wide license for universities and research institutes. RoadRunner itself can be used to import GIS terrain and aerial LiDAR to create the model, which can then be used by CARLA or SUMO for simulation. This method was performed by [30] to create a co-simulation based on aerial video of traffic of a few locations. The limitation of this method is that it requires manual creation of the 3D buildings and sacrifices visual fidelity for simplicity.

The exported map from RoadRunner can be ingested into CARLA. Alternatively, the OSM file can be converted to xodr (opendrive format [31]) using CARLA python library, and the road network can be merged with the terrain and building meshes in Unreal Engine. If CARLA is compiled from source, the 3D model and opendrive road network may be imported into the Unreal Engine project with the possibility of editing and customizing the road movements or the 3D mesh. OpenStreetMap does not always have the most up to date road links, especially in the university campus with limited access to some areas. The ability to edit waypoints manually in the Unreal project solved these issues.

B. Sensors and Autonomous Driving

Once the model is imported into the CARLA project in Unreal Engine as a map, different cars can be driven on the roads and even off the road as shown in Fig. 3(a). it offers many capabilities including multiple camera views, dynamic weather, and different sensors like optical flow (Fig. 3(c)), LiDAR, depth (Fig. 3(b)), and multiple camera angles. Large 3D models are very computationally intensive, so a powerful computer is required for smooth operation. Simulations and experiments related to autonomous vehicles, traffic congestion, signal lights, and computer vision can be performed with quick revisions and variations.

C. Co-Simulation

Co-simulation with SUMO was also performed, as seen in Fig. 3(d), which is made possible by CARLA's integration with SUMO. Some built in maps already have SUMO configurations, but it is simple to create the road network XML files by following the CARLA documentation. Large networks with a lot of vehicles might cause performance issues but overall, the experience is seamless and cars can be seen moving on both the SUMO and CARLA interfaces simultaneously, while adhering to lanes and traffic signals.

Since Unreal Engine is used for video game development, there are endless possibilities for automation and configuration. Logic for dynamic elements like vehicles, gates, or detours can be programmed using blueprints. Multiple levels can be built and combined together for expandability, for example, the insides of buildings can be modeled, real time data from IoT sensors can be incorporated, and

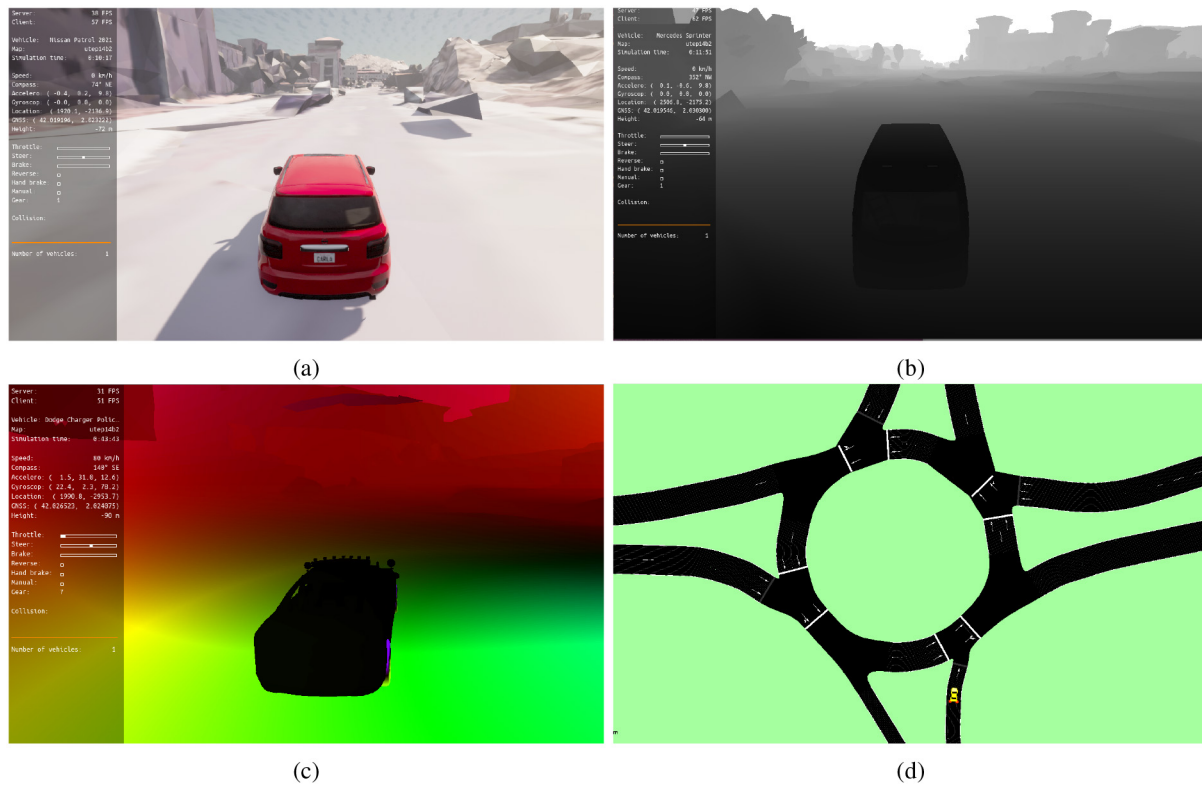


Fig. 3. (a) Driving a car in CARLA project on UTEP campus model, (b) Depth camera view, (c) Optical flow view, (d) Co-simulation in SUMO.

multiple levels of details can be supported similarly to BIM [32].

V. DISCUSSION

CARLA supports a number of different types of vehicles by default, such as sedans, SUVs, trucks, and ambulances, and these were able to run on the campus model, both manually and autonomously, as shown in Fig. 3(a). The dimensions and grades of the roads are faithfully represented in the model, with some visual and collision glitches in the 3D mesh that can be easily edited. It is also possible to go off the road in manual driving mode, allowing exploration and unique perspectives. One use case for this could be to generate a synthetic dataset for autonomous driving reinforcement learning, where going off road gives a negative reward.

In computer vision studies, ground truth for 3D bounding boxes are hard to label in real world video, but much simpler in video game environments. 3D object detection methods use data augmentation such as rotation and jitter [33] for point cloud data, or symmetry preserving transformations [34] in synthetic 3D data to prevent overfitting in deep neural networks. The model presented in this paper is ripe for such use cases with full control of weather and lighting conditions, and multiple camera angles. Integration of realtime sensor data, physics simulation, and control of actuators, for example in parking gates, can extend it into a true campus digital twin.

The goal of this exploration was to find the more efficient ways to create a 3D model of a geographical location along with the road network, but some hurdles were discovered during the implementation. First, the versions of software used are

very important. Renderdoc version 1.19 was used with Chrome (latest version) on Windows to download the 3D data, then Blender 2.93 with MapsModelImporter 0.4 plugin was used to import the captures, which were stitched together to export the FBX file. On Ubuntu Linux, CARLA version 0.9.13 was compiled from source along with Unreal Engine 4.26, while the latest version of RoadRunner was used.

Exporting the terrain and 3D buildings of UTEP and surrounding areas using Mapbox (see Fig. 2(c)) yielded a 3D model with 48,182 triangles. Comparatively, the model exported from Google maps, shown in Fig. 2(d), is somewhat bulky, containing 1.59 million triangles. At least 64 GB of RAM is needed to deal with a large model such as the one simulated in this paper, especially the road network can be quite slow to manipulate. RAM usage as high as 70 GB was observed when transforming the road network to fit the terrain in RoadRunner, so for large projects 128 GB of RAM may be suitable. Compiling CARLA and Unreal Engine also takes a long time, even on a CPU with 10 cores. The machine used for this paper has an Intel Core i9-10900X CPU, 256 GB RAM, and NVIDIA A6000 GPU.

VI. CONCLUSION AND FUTURE WORK

In general, 3D modeling and graphics work for large geographical regions is a time consuming and laborious task that detracts researchers from their main objectives. Thus, a convenient and practical method to set up the 3D model of a geographical location, with its known road network, in Unreal Engine CARLA project was discussed in this paper. This process is not just useful for digital twin modeling, it is also

beneficial for transportation and infrastructure research and visualization, along with computer vision applications involving monocular depth sensing, 3D object detection, and optical flow applications.

The model can be extended for a variety of use cases and experiments as the base of Unreal Engine is highly versatile. Specific alterations and improvements like removing cars from the 3D mesh, inserting missing traffic signs, and adding support for functional, interactive objects like gates is also possible. With minimum efforts, a VR showcase for the campus could be implemented. More advanced applications could incorporate realtime data from sensors like traffic, temperature, indoor air quality, power consumption, and building occupancy could be displayed.

REFERENCES

- [1] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.
- [2] F. Tao and M. Zhang, "Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing," *IEEE Access*, vol. 5, pp. 20418–20427, 2017.
- [3] Q. Lu, X. Xie, J. Heaton, A. K. Parlikad, and J. Schooling, "From BIM towards digital twin: Strategy and future development for smart asset management," in *Proc. Int. Workshop Serv. Orient. Holonic Multi-Agent Manuf.*, 2019, pp. 392–404.
- [4] R. Alonso, M. Borrás, R. H. E. M. Koppelaar, A. Lodigiani, E. Loscos, and E. Yöntem, "SPHERE: BIM digital twin platform," *Proceedings*, vol. 20, no. 1, p. 9, 2019.
- [5] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020.
- [6] G. Schrotter and C. Hürzeler, "The digital twin of the city of Zurich for urban planning," *PFG J. Photogrammetry Remote Sens. Geoinf. Sci.*, vol. 88, no. 1, pp. 99–112, 2020.
- [7] "International open data charter." Open Government Partnership Global Summit. 2015. [Online]. Available: https://opendatacharter.net/wp-content/uploads/2015/10/opendatacharter-charter_F.pdf
- [8] "National digital twin programme." University of Cambridge. 2022. [Online]. Available: <https://www.cdbb.cam.ac.uk/what-we-do/national-digital-twin-programme>
- [9] Y. Gao, S. Qian, Z. Li, P. Wang, F. Wang, and Q. He, "Digital twin and its application in transportation infrastructure," in *Proc. IEEE 1st Int. Conf. Digit. Twins Parallel Intell. (DTPI)*, 2021, pp. 298–301.
- [10] "What is azure digital twins?" Microsoft. 2022. [Online]. Available: <https://docs.microsoft.com/en-us/azure/digital-twins/overview>
- [11] "AWS IoT TwinMaker." Amazon. Accessed: Jul. 20, 2022. [Online]. Available: <https://aws.amazon.com/iot-twinmaker/>
- [12] "Digital twin: A primer for industrial enterprises." PTC. Accessed: Jul. 20, 2022. [Online]. Available: https://www.ptc.com/-/media/Files/PDFs/IoT/digital_twin_industrial-enterprises-6-11-19.pdf
- [13] "Solutions and applications powered by iTwin." Bentley Systems, Incorporated. 2022. [Online]. Available: <https://www.bentley.com/en/products/product-line/digital-twins/itwin>
- [14] "Autodesk tandem." Autodesk. 2022. [Online]. Available: <https://intandem.autodesk.com/>
- [15] F. Jiang, L. Ma, T. Broyd, W. Chen, and H. Luo, "Digital twin enabled sustainable urban road planning," *Sustain. Cities Soc.*, vol. 78, Mar. 2022, Art. no. 103645.
- [16] O. El Marai, T. Taleb, and J. Song, "Roads infrastructure digital twin: A step toward smarter cities realization," *IEEE Netw.*, vol. 35, no. 2, pp. 136–143, Mar./Apr. 2021.
- [17] J. Yan, S. Zlatanova, M. Aleksandrov, A. Diakite, and C. Pettit, "Integration of 3D objects and terrain for 3D modelling supporting the digital twin," in *Proc. ISPRS Ann. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 4, 2019, pp. 147–154.
- [18] V. Verma, R. Kumar, and S. Hsu, "3D building detection and modeling from aerial LIDAR data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, 2006, pp. 2213–2220.
- [19] N. Kikuchi, T. Fukuda, and N. Yabuki, "Future landscape visualization using a city digital twin: Integration of augmented reality and drones with implementation of 3D model-based occlusion handling," *J. Comput. Design Eng.*, vol. 9, no. 2, pp. 837–856, Apr. 2022.
- [20] A. Martínez et al., "Digital twin for the integration of the automatic transport and manufacturing processes," in *Proc. IOP Conf. Ser. Mater. Sci. Eng.*, Oct. 2021, Art. no. 12107.
- [21] M. Judge. "Introducing building information modeling for campuses." 2022. [Online]. Available: <https://campusinsights.aramark.com/resources-blog/introducing-building-information-modeling-for-campuses>
- [22] "Planet dump retrieved from <https://planet.osm.org/>" OpenStreetMap contributors. 2017. [Online]. Available: <https://www.openstreetmap.org>
- [23] J. Lu and X. S. Zhou. "OSM2GMNS." 2022. [Online]. Available: <https://github.com/jiaulu/OSM2GMNS>
- [24] "Maps, geocoding, and navigation APIs and SDKs." Mapbox. 2022. [Online]. Available: <https://www.mapbox.com/>
- [25] "Blender—A 3D modelling and rendering package." Blender Online Community. 2018. [Online]. Available: <http://www.blender.org>
- [26] B. Karlsson. "RenderDoc." 2019. [Online]. Available: <https://renderdoc.org/>
- [27] E. Michel. "Maps models importer." 2022. [Online]. Available: <https://github.com/eliemichel/MapsModelsImporter>
- [28] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.
- [29] "RoadRunner." Mathworks. 2022. [Online]. Available: <https://www.mathworks.com/products/roadrunner.html>
- [30] O. Zheng. "UCF-SST-CitySim-dataset." 2022. [Online]. Available: <https://github.com/ozheng1993/UCF-SST-CitySim-Dataset>
- [31] ASAM OpenDRIVE, *Open Dynamic Road Information for Vehicle Environment*, Assoc. Stand. Autom. Meas. Syst., Höhenkirchen-Siegertsbrunn, Germany, 2020.
- [32] M. L. Trani, M. Cassano, D. Todaro, and B. Bossi, "BIM level of detail for construction site design," *Procedia Eng.*, vol. 123, pp. 581–589, Jan. 2015.
- [33] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1708–1716.
- [34] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2686–2694.