



Article

A Geospatial Platform to Manage Large-Scale Individual Mobility for an Urban Digital Twin Platform

Ahyun Lee *, Kang-Woo Lee, Kyong-Ho Kim and Sung-Woong Shin

City & Transportation ICT Research Department, Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; kwlee@etri.re.kr (K.-W.L.); kkh@etri.re.kr (K.-H.K.); sshin@etri.re.kr (S.-W.S.)

* Correspondence: ahyun@etri.re.kr

Abstract: Urban digital twin (UDT) technology can be used to digitize physical urban spaces. Previous UDT or smart city research reconstructed the three-dimensional topography of urban spaces, buildings, and facilities. They collected various multimodal sensor data from cities and monitored conditions such as temperature, humidity, fine dust, and real-time road traffic. However, these studies lacked ways to manage individual mobility data, such as those of the vehicles and pedestrians, which constitute major components of a city. Here, we propose a geospatial platform based on the universal game engine Unity3D, which manages large-scale individual mobility data for a UDT platform. The proposed platform stores and manages individual vehicles or pedestrians using information from public closed-circuit television. It also allows the generation of long-term route information for a unique vehicle based on its license plate. We also propose methods to anonymize license plates, to ensure the security of individuals, and to compress individual mobility data. Unique UDT models with individual mobility functionalities can be built and visualized using our proposed geospatial platform.

Keywords: 3D maps; digital twin; spatial information; individual mobility



Citation: Lee, A.; Lee, K.-W.; Kim, K.-H.; Shin, S.-W. A Geospatial Platform to Manage Large-Scale Individual Mobility for an Urban Digital Twin Platform. *Remote Sens.* **2022**, *14*, 723. <https://doi.org/10.3390/rs14030723>

Academic Editors: Soheil Sabri, Abbas Rajabifard, Yiqun Chen, Nengcheng Chen and Hao Sheng

Received: 13 December 2021

Accepted: 31 January 2022

Published: 3 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the way that humans developed civilizations by storing knowledge as letters, the digitization of the physical “real” world that people can see and touch could be one possible next step. We live in an era that would have been difficult to imagine in the past, where information and technology are combined with computers and the Internet. What kind of world could be created if physical realities, as perceived with the five senses, could become digital data that could be updated to resemble reality? Such technology, in which the updated digital data are known as a digital twin, is not common [1,2]. Initially, digital twins were created using sensors attached to complex machines or parts, and used to manage replacement cycles or failure; however, they are now used in smart factories [3–5]. For example, in the complex process of assembling a car, the only moving element is a robot—there are no humans involved—and information on the components of the car are managed digitally.

Cities can also be subjects of digital twin technology [6,7]. Virtual Singapore, a Dassault Systèmes [8] project involving the creation of a three-dimensional (3D) model of the entire city-state of Singapore, has been used for various urban planning processes that require an actual physical model [9,10], where previously only geometric and physical models were available for urban topography. It has also been used as a navigation service for the elderly and disabled, to predict energy generated from solar power, and to simulate the construction of new buildings based on sunlight or city temperature [11,12].

Google Earth is the easiest way for the public to access a 3D digitalized city [13]. Google Earth builds and provides 3D geospatial data for major global cities on a planetary scale. Users can experience different views of these cities, such as flying over them, just by searching Google Earth in an HTML5 standard-based web browser; no pre-installation is

required. Cesium ion allows users to upload various 3D models of their own to a cloud server and place them on the surface of planetary-scale 3D geospatial maps [14–16]. VWorld provides the most precise 3D city model in the Republic of Korea [17]; unlike Google Earth or Cesium, it provides individual 3D building and facility models at a high resolution, and enables more precise city planning and simulation [18]. Having data on individual buildings or facilities enables the construction of networks and improves radio wave efficiency and fine dust simulation [19]. However, it is difficult to judge whether a city has been digitized when only 3D topography, buildings, and facilities are part of it. The most important component, the people, are currently missing.

Virtual Singapore conducted an experiment to collect and analyze individual mobility information from pedestrians using a sensor called SENsg [20]. It was a national science experiment, wherein data on the daily lives of Singaporean students with access to SENsg and a global positioning system were collected, using approximately 50,000 nodes, for approximately 3 months [21]. As a result, it was possible to quantify the times at which students were indoors or outdoors and analyze their point of origin, destination, or travel time. A device-holding approach has enabled individual mobility-sensing projects to work on a large scale.

A web-based 3D city model of New York, Vision Zero, shows a small white moving sphere on the road [22]. The white sphere is not a moving vehicle or public transportation facility, but an animation of the speed in a specific section of the road. In addition, Vision Zero provides dynamic data including types of traffic accidents, casualties, and air quality information [23]. The urban digital twin (UDT) platform we propose aims to support and provide data that are equal to the number of actual vehicles present in a space, without representing it as a single circle only. The UDT platform can also reproduce individual spatiotemporal mobility.

Urban problems can also be solved by detecting, tracking, and analyzing individual mobility objects. Hangzhou city, China, reduced traffic congestion during peak hours by 9.2% and increased average traffic speed by 15.3% by using 3500 traffic cameras that were part of the Alibaba cloud-based City Brain project to detect and analyze vehicles passing through 1300 intersections [24]. City Brain provides major functions such as real-time traffic event detection, tracking target vehicles or persons, and improving traffic by predicting volumes thereof [25].

A UDT is a digital model of a city and consists of physical assets and multimodal sensor data. Using real data in urban planning can lead to better decision-making that is efficient in terms of cost and operation, and allows better management of problems related to urbanization. A UDT platform should provide geospatial data that can be stored, managed, analyzed, and visualized in three dimensions by reproducing various viewpoints. We propose a UDT platform that can also achieve this for object models of vehicles and pedestrians; these are dynamic data based on large-scale sensing of individual mobility. The proposed platform constructs static and dynamic geospatial data using a tile-based system and includes data storage and management methods, detection of individual mobility data, and 3D visualization of UDT models.

In this study, we present a geospatial platform that manages large-scale individual mobility for UDTs. The remainder of this paper is organized as follows: In Section 2, we describe the type and format of the urban geospatial data covered by this platform; in Section 3, we explain the structure and procedures of the UDT geospatial platform for managing large-scale individual mobility. A UDT platform-based procedure for managing private information, and some issues that may arise therefrom, are discussed in Section 4; and concluding remarks are given in Section 5.

2. Urban Geospatial Data

Our proposed geospatial platform for UDT can provide mobility information for individual objects such as vehicles or pedestrians. All types of urban geospatial data were located within a city, and in this study the urban geospatial data for the UDT model were

classified as static or dynamic. Static data are fixed data sets that do not change after being recorded, such as aerial images, digital elevation models (DEMs), 3D terrain models, 3D buildings, road networks, and administrative information. Static data can be set to update at specific periods. However, the value of the static data is not changed; a new static dataset with similar properties is recorded as a new additional layer. Urban geospatial data are managed as layers by grouping similar data properties.

Figure 1 shows an example of static geospatial data collected over time using layers. Our proposed platform uses geospatial data from VWorld. Aerial images from this platform use years as layers, and users can combine and utilize images from various time zones based on the purpose for which they are needed. The aerial images in Figure 1, showing Yeouido in Seoul from 1978 to 2020 and sorted by year, have similar properties as aerial images; however, they can be classified as `aerial_1978`, `aerial_1989`, `aerial_1996`, and `aerial_2020` according to each time zone established. These aerial images can show the changes in topography, facilities, and buildings made in Yeouido over the past 40 years.

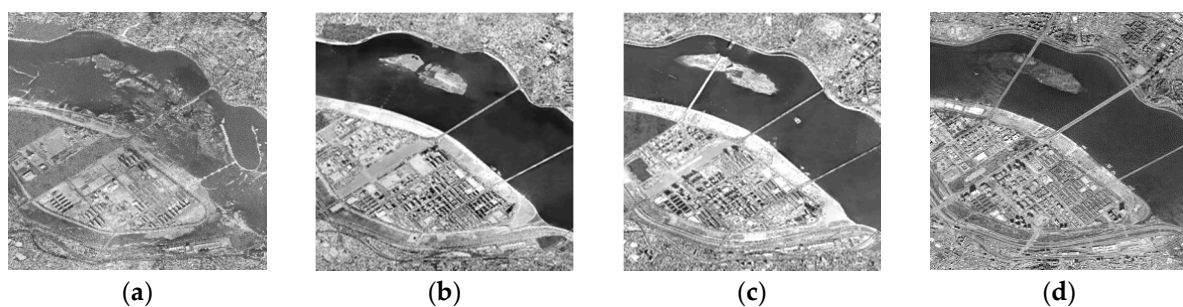


Figure 1. Example of yearly aerial image management: aerial images of the same region as layers based on the year the image was taken: (a) 1978; (b) 1989; (c) 1996; (d) 2020. All images are layer data, based on the year the image was taken, at the tile position of {level: 10, IDX: 8730, IDY:3627}.

Dynamic geospatial data are those whose values change at regular intervals. They are typically collected from traffic lights around a city or sensors focusing on weather, temperature, humidity, fine dust, and traffic volume. In previous UDT studies, dynamic geospatial data collected from sensors were combined and used with 3D static geospatial data such as topography, facilities, and buildings. However, this creates a representation that differs from the city seen in real life. Figure 2 shows a city as it is in reality and in a 3D model. Although the terrain and buildings are similar, the biggest difference is the presence of vehicles. Although topography, buildings, and roads are the main components of urban spaces, the most important factor therein is people. Vehicles, bicycles, and motorcycles that people use daily for transportation are the main elements comprising a city.

For a UDT to closely resemble the real world, data on vehicles and pedestrians—reflections of human movement in a city—must be collected. This is different from analyzing traffic volume or signal latencies to improve traffic jams on roads or detect and analyze temporary vehicle objects as part of developments in autonomous driving. Our aim is to reproduce a desired time and place in 3D by adding individual pedestrians and vehicles to the UDT model. In this study, we defined individual mobility data as vehicles or pedestrians that can move in an urban space. If such data can be stored and managed as individual objects, it will expand the possibility of utilizing UDT-based urban planning.

In this study, both static and dynamic geospatial data included location information. The unit of “tiles” is used to manage a large quantity of high-capacity data. A tile is the minimum unit in which geospatial data can be stored and managed. By managing data in this way, one can quickly search for and access the data desired using location information. Therefore, 3D geospatial platforms such as planet-scale Google Earth manage data as tiles [13,26].



Figure 2. Differences between the real world and 3D city models: roads and buildings in (a) roads and the real world; (b) a 3D terrain and building model of Yeouido-daero, Yeongdeungpo-gu, Seoul, Republic of Korea.

The quadtree tile system is constructed according to how the surface of Earth is divided or how many levels are built. The VWorld tile system that our proposed platform uses has a quadtree-based structure, and a level 0 tile is defined as one that is 50 equal parts of Earth's surface divided into five latitude steps and ten longitude steps, as shown in Figure 3. The latitude ranges from -90° to 90° and longitude ranges from -180° to 180° . Conceptually, this means that all tiles should be square; however, the actual 3D form rendered is a curved surface, as shown in Figure 4, a level 0 tile divided into four level 1 tiles or sixteen level 2 tiles. All aerial images and tile DEMs have the same resolution. Therefore, if a low level (highest: 0; lowest: 15) is displayed as the camera approaches the surface of Earth, geospatial data can continuously be visualized at high-precision resolutions.

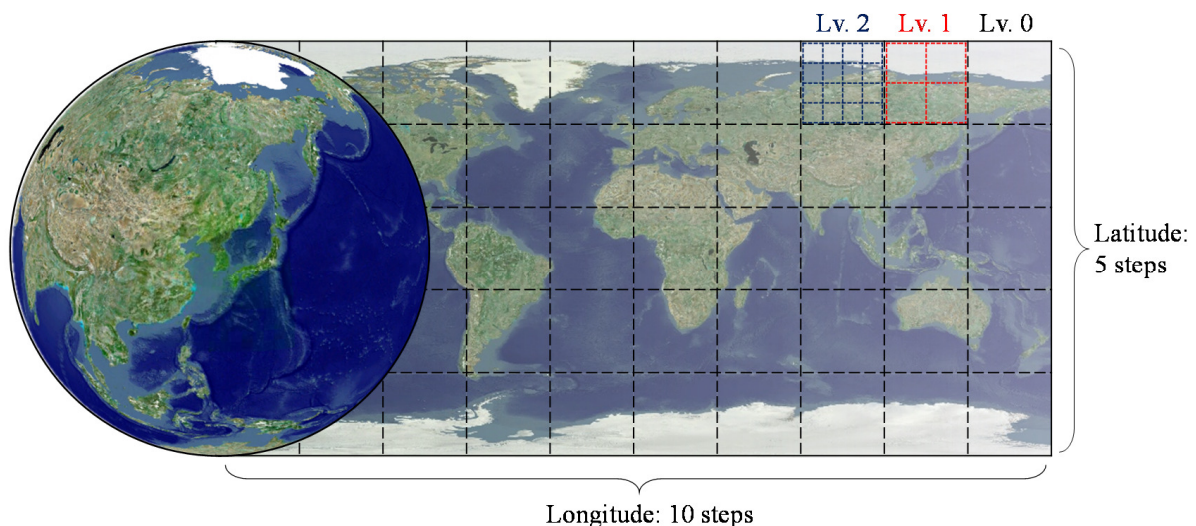


Figure 3. Tile system: level 0 tiles divide the surface of Earth into 5 latitude steps and 10 longitude steps. When a level 0 tile is divided into four or sixteen smaller tiles, the smaller tiles become, respectively, level 1 or level 2 tiles.

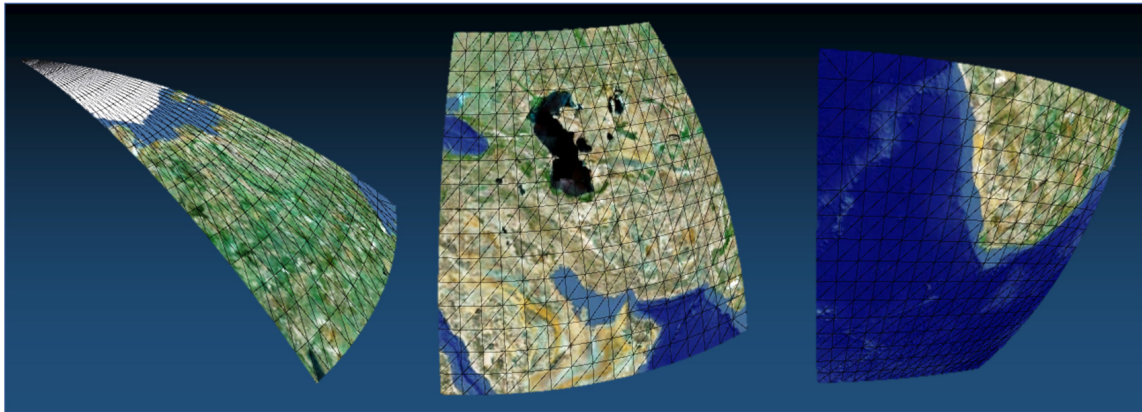


Figure 4. Example of the implementation of our 3D terrain model: tile surfaces are curved to represent a spherical globe.

In our platform, a tile is the smallest unit by which geospatial data are managed; all of the latter are located in a tile based on the central position of the data model. The geospatial data managed in a tile were divided into various layers based on their properties. Figure 5 shows how various types of 3D geospatial data are visualized in our platform with the 3D terrain, building, and bridge; facility name; and road name layers activated. The advantage of tile-based data management is that a target data path can be acquired without a special search procedure once a service area and the tiles to be included are determined. Additional user-defined geospatial data can be added as layers or a unique UDT model can be composed using the layers provided. Unique level of detail (LOD) systems can also be composed using our proposed tile-based management system. Table 1 lists the geospatial data layer types used in the proposed platform [17,18].

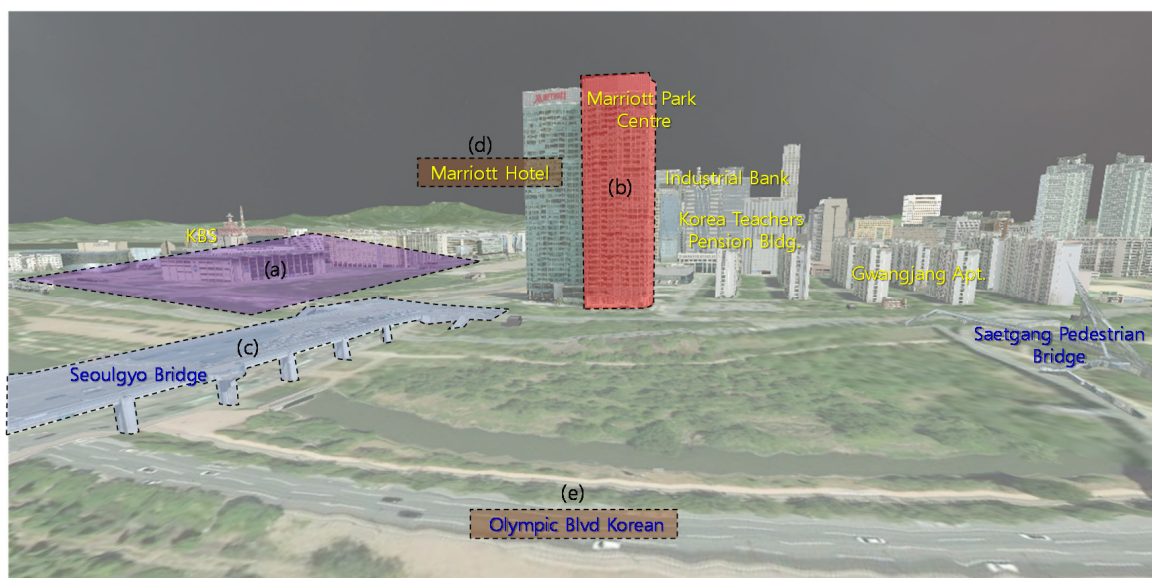


Figure 5. Example of the results of 3D rendering using our urban digital twin model: one of each type of layer data is indicated by the dotted lines. (a) A 3D terrain tile model generated by combining an aerial image and a digital elevation model; (b) a 3D building model; (c) a 3D bridge model; (d) facility name; (e) road name.

Table 1. Urban geospatial data layer classification.

	Layer Name	Type	Request URL Sample	Remarks
1	aerial	image	Layer=tile&Level=level&IDX=idx&IDY=idy	Default aerial image
2	aerial_1978	image	Layer=tile_1978&Level=level&IDX=idx&IDY=idy	Past aerial image
3	aerial_1989	image	Layer=tile_1989&Level=level&IDX=idx&IDY=idy	Past aerial image
4	aerial_1996	image	Layer=tile_1996&Level=level&IDX=idx&IDY=idy	Past aerial image
5	aerial_2020	image	Layer=tile_2020&Level=level&IDX=idx&IDY=idy	Past aerial image
6	DEM	binary	Layer=Dem&Level=level&IDX=idx&IDY=idy	Digital elevation model
7	XDO index	binary	Layer=dat&Level=level&IDX=idx&IDY=idy	XDO file list in a tile
8	Facility build	3D model	Layer=facility_build&Level=level&IDX=idx&IDY=idy DataFile=file.xdo	3D model
9	Facility build_texture	3D model	Layer=facility_build&Level=level&IDX=idx&IDY=idy DataFile=file.jpg	3D model texture
10	Bridge	3D model	Layer=facility_bridge &Level=level&IDX=idx&IDY=idy DataFile=file.jpg	3D bridge model
11	Building B3DM	3D model	facility_LOD4/Seoul-2021/Yeongdeungpo-gu-yeouido/file.b3dm	Separation of folder locations by region (Seoul)
12	Building B3DM	3D model	facility_LOD4/Busan/Busan/file.b3dm	Separation of folder locations by region (Busan)
13	Area name	text	Layer=poi_bound&Level=level&IDX=idx&IDY=idy	Image showing administrative boundaries with lines
14	Road	text	Layer=poi_road&Level=level&IDX=idx&IDY=idy	
15	facility name	text	Layer=poi_base&Level=level&IDX=idx&IDY=idy	
16	administrative boundary	image	Layer=hybrid_bound&Level=level&IDX=idx&IDY=idy	Image showing administrative boundaries with lines
17	User model	3D model	Layer=user_model&Level=level&IDX=idx&IDY=idy	3D model added by user
18	User marker	image, text	Layer=user_marker&Level=level&IDX=idx&IDY=idy	2D marker (image and text) added by user

All tiles include one aerial image and one DEM layer. An aerial image and a DEM were used to create a 3D terrain model in the form of a curved surface, as shown in Figure 4. The other geospatial data were requested and generated according to the layer active in the application. All geospatial data were managed using the basic unit of a tile. The index of a tile is one of its parameters calculated using its three properties—*level*, *IDX*, and *IDY*—as in Equation (1) below. All target tiles managed by the proposed platform using the tile index can be managed as a single list-type variable.

$$index = 50 \cdot \frac{2^{level*2} - 1}{3} + 10 \cdot 2^{level} \cdot IDY + IDX \quad (1)$$

Each tile level stored was given a designation based on geospatial data properties. The text of country and major city names are, respectively, stored in level 0 tiles and included in level 1 tiles. The aerial image and DEM are included at all tile levels from 0 to 15. Buildings or facilities are stored in tiles from levels 13 to 15, depending on the size of the model. It is impossible to request and render all buildings located in a country in real time. However, our approach has the advantage of having a small model, which can only be perceived when the camera is close to the ground, in the lower-level tiles.

3. Urban Digital Twin Platform

The proposed platform includes functions such as collecting, analyzing, storing, and managing urban geospatial data to represent a UDT model. It updates this model via camera sensors that detect and analyze individual mobility objects. It also provides a function for visualizing 3D individual mobility objects based on the time and space a user requires.

Our focus is on individual mobility data for a UDT model; therefore, the proposed platform has a procedure and structures by which individual mobility objects can be detected and the UDT model updated. Figure 6 shows the structure of the proposed UDT platform. In previous UDT studies, sensors collected a simple numerical value and their

positions were fixed to update the UDT model [22]. In contrast, we use monocular camera lens-based closed-circuit television (CCTV) to collect data on individual mobility objects. It is inefficient to store and manage video data from CCTVs on a central server. However, unanalyzed CCTV video—which contains a substantial amount of data—cannot be used alone to update the UDT model. To update the UDT model efficiently, it is necessary to store and manage meaningful analysis information from CCTV videos. In addition, by simplifying the analyzed data, the size of the UDT data must also be scaled down to manage the city-scale UDT model.

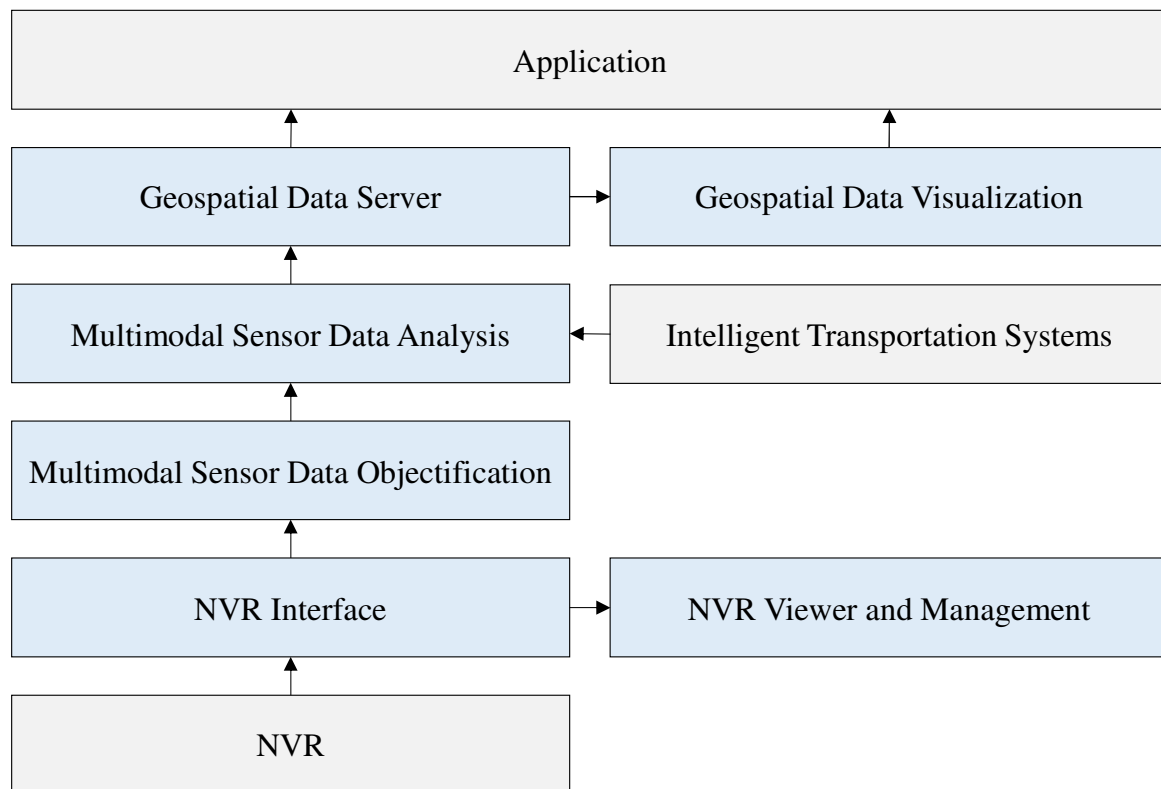


Figure 6. Structure of the proposed urban digital twin platform. The blue blocks are the modules proposed in our platform.

3.1. Objectification

Objectification refers to the detection of individual mobility objects in two-dimensional (2D) RGB images, estimating their locations, and classifying objects such as pedestrians and types of vehicles, with the latter being able to distinguish between, for example, buses, cars, and trucks. Objectification converts high-capacity video data into low-capacity objectified data, enabling the storage and management of the locations, types, speed, or direction of moving objects. The network video recorder (NVR) interface module transfers the video data from CCTVs installed in a city to the multimodal sensor data objectification module. NVRs are devices that include hardware that can store data by connecting one or more Internet protocol (IP) cameras, and software that can control them and manage their video. Most NVRs provide a real-time streaming protocol (RTSP) interface, and each live video from IP cameras can be streamed through the RTSP application programming interface. Previously recorded videos can be requested from a specific camera ID or time zone; however, these video data come in large file sizes. For example, the XRN-1610A NVR equipment with a 32-terabyte (TB) hard disk drive and four XNO-6080R IP cameras used in this experiment can store approximately eight months of full high definition (FHD) video at 30 frames per second (FPS) from the four IP cameras. Approximately 1 TB is required to record monthly videos in FHD from an IP camera. Therefore, the proposed platform detects,

analyzes, and stores only necessary CCTV video data using the NVR interface module. Furthermore, because data interface standards vary among different digital video recorder or NVR manufacturers, we developed an interface module for various NVR devices most commonly used in the Republic of Korea. When it is used in various fields in the future, the interface should be modified according to each product.

Figure 7 shows the data processing procedure that the proposed image-based multimodal sensor data objectification and analysis module uses. The module's target data may use a different type of video sensor, such as a car dashboard camera, instead of a CCTV. One or more CCTVs is connected to one NVR, and one or more NVRs is linked to one NVR interface module. One NVR interface module is connected to an objectification module; one or more of the latter are, in turn, connected to a multimodal sensor data analysis module.

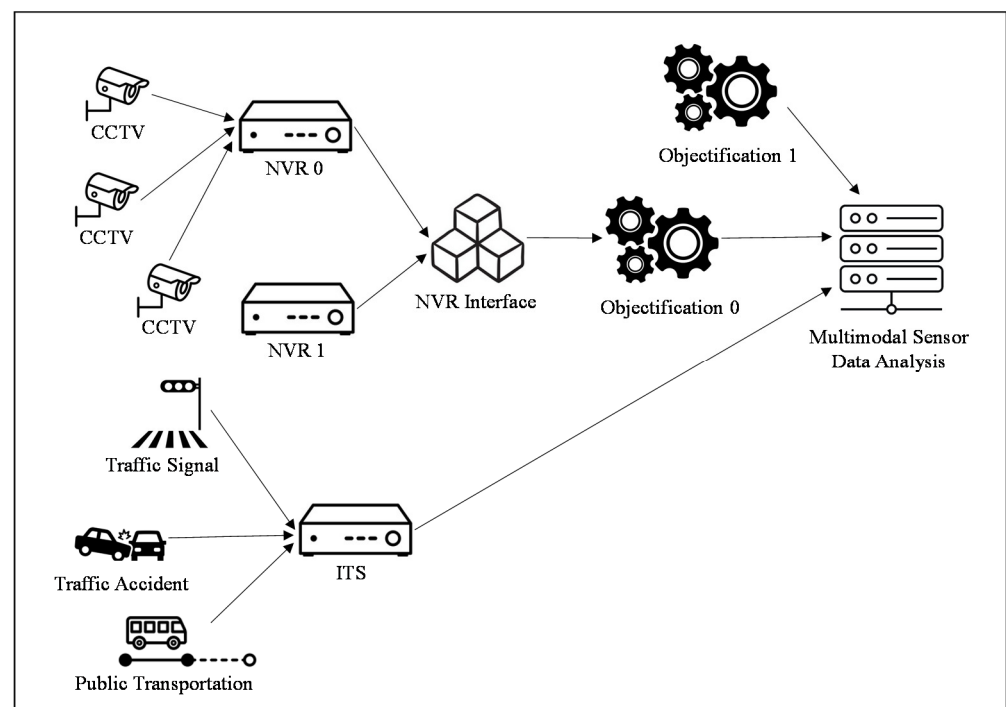


Figure 7. Image-based multimodal sensor data objectification and analysis.

It is inefficient to store and manage raw public CCTV video data on a central server for a UDT, particularly given that there were 1,148,770 public CCTV units (those operated by public institutions) in the Republic of Korea as of 2019. This value is an approximately 1.7-fold increase compared to 2011, and it is expected to increase annually by 5–10% in the future. The size of data handled by a server can be greatly reduced using the proposed method of storing and managing image-based multimodal sensor data via objectification. Executing the objectification module produced the objectified data for each frame, configured as shown in Table 2.

Table 2. Objectification of vehicles in a frame of closed-circuit television video.

Name	Content
Camera ID	2580-01
Global ID	0xEF5bef53CCDAaA4b6A68
Local ID	034
Type	car
Position	127.38500, 36.35111, 15.4, 127.38500 (latitude/longitude/elevation)
Time	2021:07:02:13:54:35:01 (year:month:day:hour:minute:second:frame)

In Table 2, the camera ID is a unique value of a specific camera (IP CCTV ID). Global ID is a unique value assigned to a vehicle license plate for the entire platform and was used to identify the vehicle. In this study, the global ID was linked to the license plate to prevent the leakage of personal information; it is encrypted and converted into a form that cannot be reversed. Local ID is the ID of an object that is captured and detected in successive frames from the same camera. Objects can be distinguished by local ID only in continuous frames of a specific section of one camera. The category of vehicle type is divided into cars, trucks, and buses. We used SiamMOT [27] to detect vehicles or pedestrians. Figure 8 shows the detection results of the testbed designed to demonstrate the proposed platform. It is assumed that the longitude, latitude, and altitude can be estimated using the position of the 2D image pixels in the videos because cameras are calibrated. The proposed platform is not limited to a specific region or country; it uses a sphere-based coordinate system axis [19] and can therefore be used anywhere worldwide. Figure 9 shows the coordinate system used in the proposed platform.

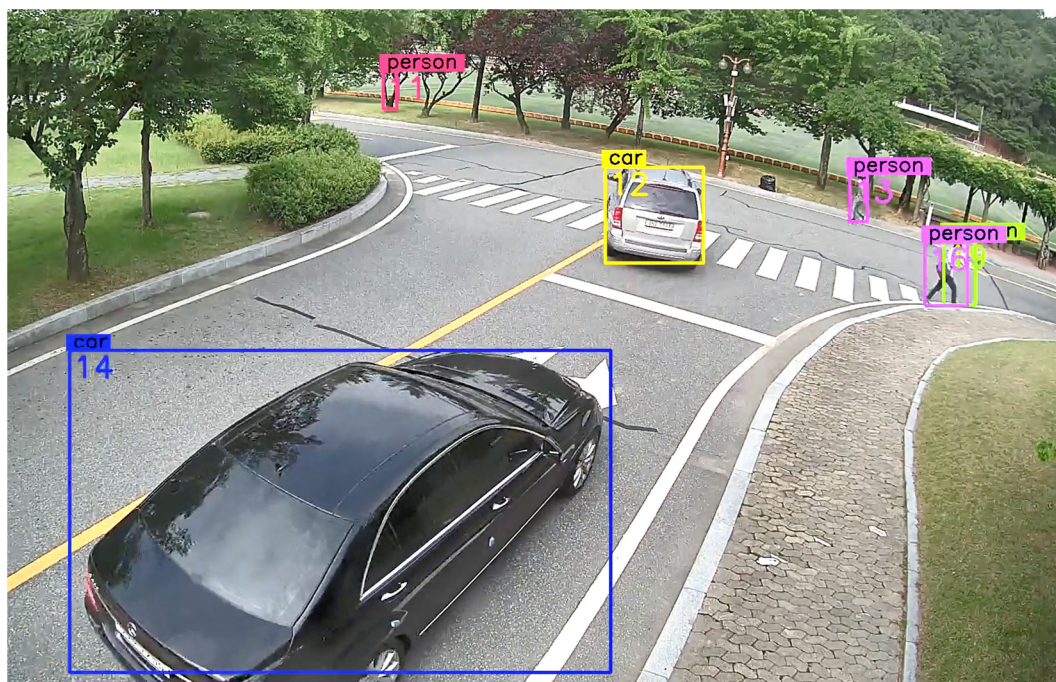


Figure 8. Objectification result for individual mobility in a testbed built to test the proposed platform.

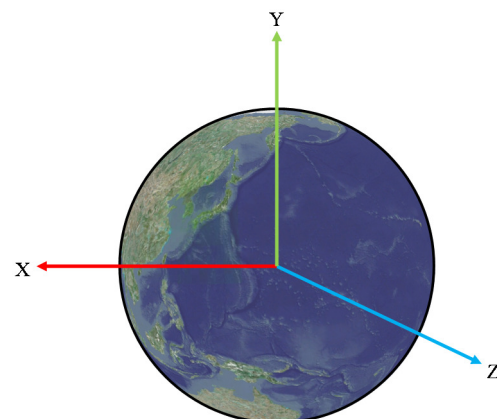


Figure 9. Sphere-based 3D coordinate system used in the proposed platform.

The origin of the coordinate system is the center of Earth, which is assumed to be a perfect sphere. The positive direction of the y -axis passes through the North Pole (latitude

90°), and the negative direction passes through the south pole (latitude -90°). The positive direction of the z-axis passes through latitude 0° and longitude 180° , and the negative direction passes through latitude 0° and longitude 0° . The positive direction of the x-axis passes through latitude 0° and longitude 90° , and the negative direction passes through latitude 0° and longitude -90° . The latitude and longitude coordinates were transformed into a three-dimensional coordinate vector using Equation (3) below [18]:

$$s = UR + elev * UR / 6378137 \quad (2)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} s * \cos lat * \sin lon \\ s * \sin lat \\ -s * \cos lat * \cos lon \end{bmatrix} \quad (3)$$

where *lat* and *lon* are latitude and longitude, respectively, both measured in radian angles; *elev* is elevation; and *UR* is the radius of Earth used within the proposed platform. In the Unity3D engine, all *GameObject* values—the basic units used in Unity3D to represent characters, props, and landscapes, and act as a container for all components in the project—can only be stored in the float type. Values with more than six to nine digits in the float type are approximated; this limits the size of a map that can be used in a Unity3D project. Therefore, the proposed platform uses the downscaled radius of Earth in the Unity3D-based 3D coordinate system.

3.2. Multimodal Sensor Data Analysis

The multimodal sensor data analysis module analyzes continuous frames of objectified models detected from CCTV video via objectification and combines this with traffic signals and road network information to provide behavioral statuses. A vehicle can be classified as going through basic behavior patterns such as moving forward or backward, decelerating or accelerating, turning left or right, changing to the left or right lane, and stopping or starting [28]. We expect that integrated analysis is possible based on using objectified model data that estimate the location of vehicles. Traffic light and traffic accident information or road network data can also be imported and analyzed based on the location of the vehicles. Table 3 presents an example of individual mobility object analysis data.

Table 3. An example of individual mobility object analysis data.

Name	Content
Camera ID	2580-1
Global ID	0xEF5bef53CCDAaA4b6A68
Local ID	034
Type	car
Position	127.38500, 36.35111, 15.4,
(number of frames)	 127.38501, 36.35112, 15.5, (latitude/longitude/elevation)
Time	2021:07:02:13:54:35:01 ~
(number of frames)	2021:07:02:13:54:36:15 (year:month:day:hour:minute:second:frame)
Direction	(0.96, 0.24, 0.06),
(number of frames)	 (0.92, 0.27, 0.02)
Speed	30, 30.1,, 21.4 km/h
(number of frames)	
Action Label	forward, forward,, enter intersection,
(number of frames)	deceleration + turn left
Driving Analysis	after entering the intersection, turn left while decelerating

Objectified data have only the ID, type, time, and position information out of what is given in Table 2. An individual mobility object is labeled with its direction, speed, and action using the location of the objectified data from two or more frames and the road lane

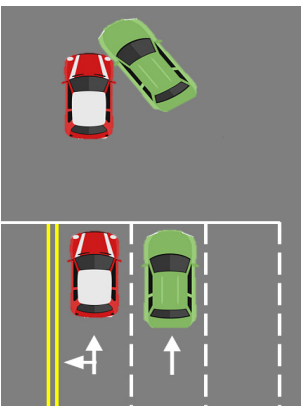
where it is located. Direction is a unit vector of a direction based on the changes in position for each frame. Speed is calculated using the difference in position from the previous frame. Thus, the direction vectors and speed parameters from all frames except the first were recorded. For a vehicle, a basic action is classified into one or more action labels per frame. In the proposed platform, the basic behaviors of vehicles were classified into 15 categories, as shown in Table 4.

Table 4. List of action labels to apply to a vehicle.

Name	Content
Forward	Moving forward
Backward	Moving backward
Deceleration	Speed is reduced
Acceleration	Speed is increased
Turn right	Turn right
Turn left	Turn left
Change to the right lane	From starting to change into the right lane until it ends
Change to the left lane	From starting to change into the left lane until it ends
Make a U-turn	From starting and ending a U-turn crossing the center lane
Stop	When the car is at zero speed
Park	Stopped state after a specified period has elapsed
Collide with another vehicle	Collide with another vehicle
Collide with road facility	Collide with road facility
Collide with pedestrian	Collide with pedestrian
Signal violation	Includes various violations of road signals

The action label can be granted one or more frames per object. For example, when a vehicle decelerates it may be classified as decelerating while moving forward within a lane, decelerating while changing lanes, or decelerating while turning left at an intersection. By integrating more than two action labels, it is possible to define the complex situation of an individual mobility object that is captured and detected in successive frames of one camera. Basic action labels can be combined to create new situations. For example, ‘forward + deceleration’ occurs when the vehicle decelerates while moving within the lane. ‘Lane change + deceleration’ means a case of decelerating while changing lanes. It can be used to determine the cause of, and responsibility for, an accident by complexly analyzing the behavior of individual vehicles, as shown in Table 5. In this case, road network data or traffic light sensor information are required. Therefore, public data from intelligent transportation systems are imported and used in the multimodal sensor data analysis module.

Table 5. Example of vehicle accident analysis; the green vehicle perspective is on the right.

Situation	Action Labels	Place	Visualization
Collision after turning left in the straight lane at an intersection	Deceleration + turn left + lane violation	Straight lane → intersection	

3.3. Compressing Data for an Individual Mobility Object

The individual mobility object data analyzed through the objectification and multimodal sensor data analysis modules connected to CCTVs installed in urban spaces are stored in a data server that a user can access to create their own UDT model. Our proposed platform can update corresponding UDT models by collecting multimodal sensor data. Therefore, UDT models that use individual mobility objects should be updated continuously using the proposed server.

The tile unit where the detected and analyzed individual mobility object data are stored is based on the location of the corresponding sensor. The location where the individual mobility object layers are stored is defined as the lowest level (15) of the 16 steps of tile levels from 0 to 15. The tile structure is a quadtree with the highest level being 0. Other geospatial data, such as buildings, bridges, and facilities, are stored and managed as a 3D model. Only objectified and analyzed data for an individual mobility object are stored and managed on the server. Depending on the user, individual mobility objects can be expressed using points, lines, or 3D vehicle models. To store and manage large-scale data, objectified and analyzed individual mobility data must be compressed and managed according to the geometric properties of each moving object.

Figure 10 shows examples of the three types of individual mobility object data compression. To compare capacity, commonly stored information such as ID, time, start, and end points were excluded. Figure 10a–d shows approximately 1 MB per frame of FHD video, a total of 150 MB based on 150 frames. By sampling from 150 frames in which objectification was performed, 20 pieces of objectified data (ID, type, time, and location) were compressed as shown in Figure 10e. Figure 10f minimizes the capacity of the data sampled in Figure 10e to 8 bytes (double) $\times 4 = 32$ bytes per object through the curve-fitting method [29]. In the curve-fitting method, the number of parameters is determined as four, or the value set by the user, per object. The parameter w is derived by performing polynomial curve-fitting using the two-dimensional coordinates (x, t) of the sampled data, as shown in Equations (4) and (5) below:

$$y(x, w) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M \quad (4)$$

$$E(w) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, w) - t_n\}^2 \quad (5)$$

where N is the number of sampling data points and the value of w , for which $E(w)$ is the lowest is used. The value of the parameter stored according to the polynomial order is $M + 1$. The proposed platform uses M as three for the first time. If $E(w)$ in Equation (5) is judged to be beyond a certain threshold, the curve-fitting method is not used and the values sampled in Figure 10e are compressed. As a result, a car has a relatively gentle change in movement, whereas the left and right fluctuations would occur irregularly for a pedestrian, as shown in Figure 10e. For the latter, the curve-fitting compression method can dilute the raw position values.

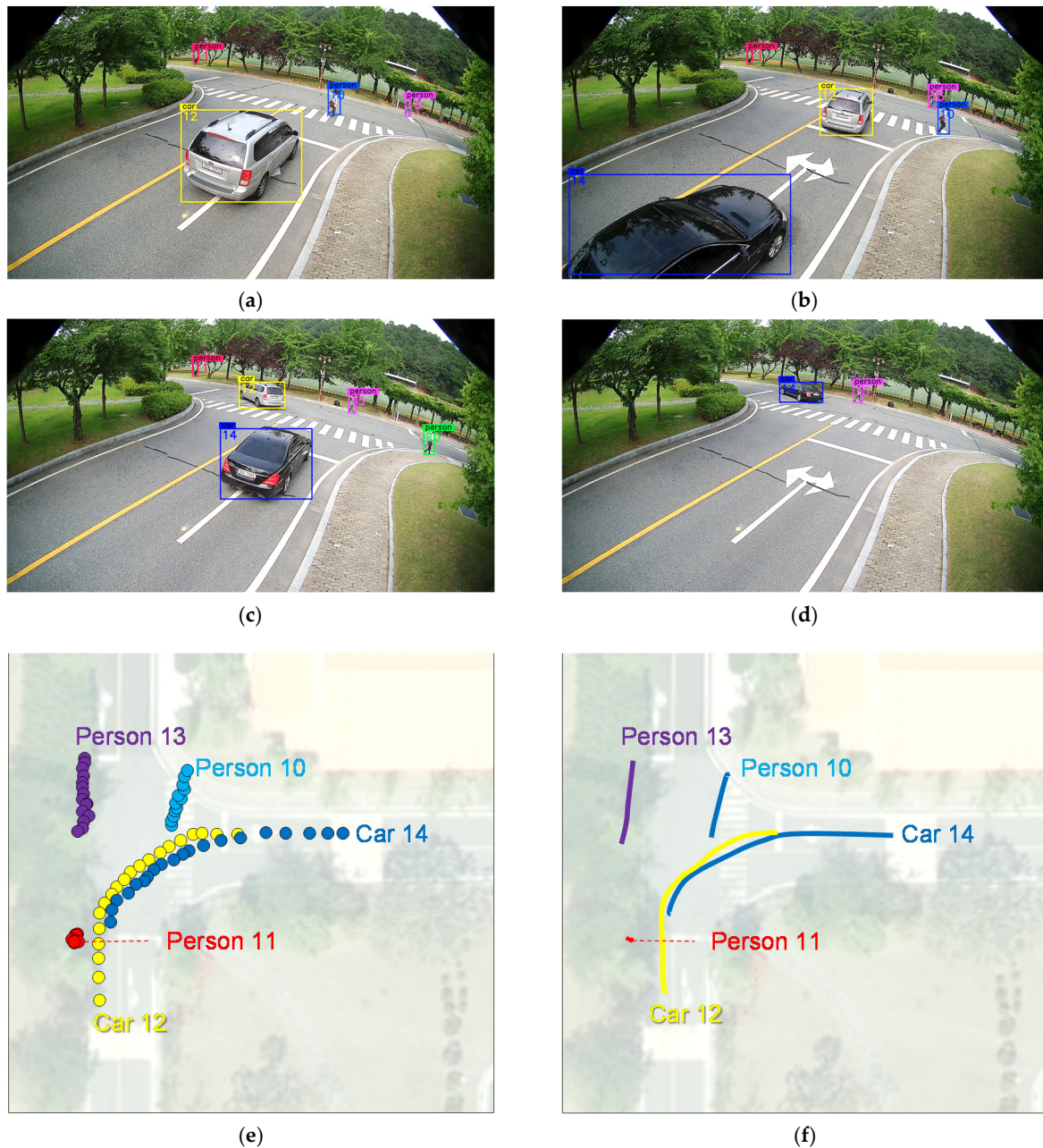


Figure 10. Approaches to compressing data for individual mobility objects: the (a) 1st; (b) 33rd; (c) 66th; (d) 100th frame of the closed-circuit television video in our testbed; (e) result of sampling the entire frame to 1/5 size; (f) curve-fitting result of the sampled positions [29].

3.4. Vehicle License Plate De-Identification

The purpose of our proposed UDT platform is to store and manage long-term individual mobility object data. Unlike pedestrians, which are difficult to identify using only CCTV cameras, vehicles can generate a unique global ID through their license plates. If the unique ID of a vehicle is managed by a central server, it can utilize the long-term route and analysis results for a specific vehicle for application in various ways. For example, long-term analysis data for vehicles operating in a specific area can be used in simulations of new road construction and causes of traffic congestion. Several individual mobility objects can be utilized through a unique global ID. However, it is necessary to prevent users from accessing individual personal information. It is illegal for anyone except specific authorized persons to view data containing personal information on public CCTVs.

Therefore, we propose a method to generate an unidentifiable global ID for vehicles such that they cannot be identified via their license plate, i.e., we propose a vehicle license plate de-identification method. A global ID would have a unique global property and the license plate, which is personal information linked to a vehicle, will not be able to be estimated inversely. However, the movement and trajectory information for each vehicle to which a global ID is assigned will still be browsable. Figure 11 shows the data flow of the proposed vehicle license plate de-identification and the global ID generation method.

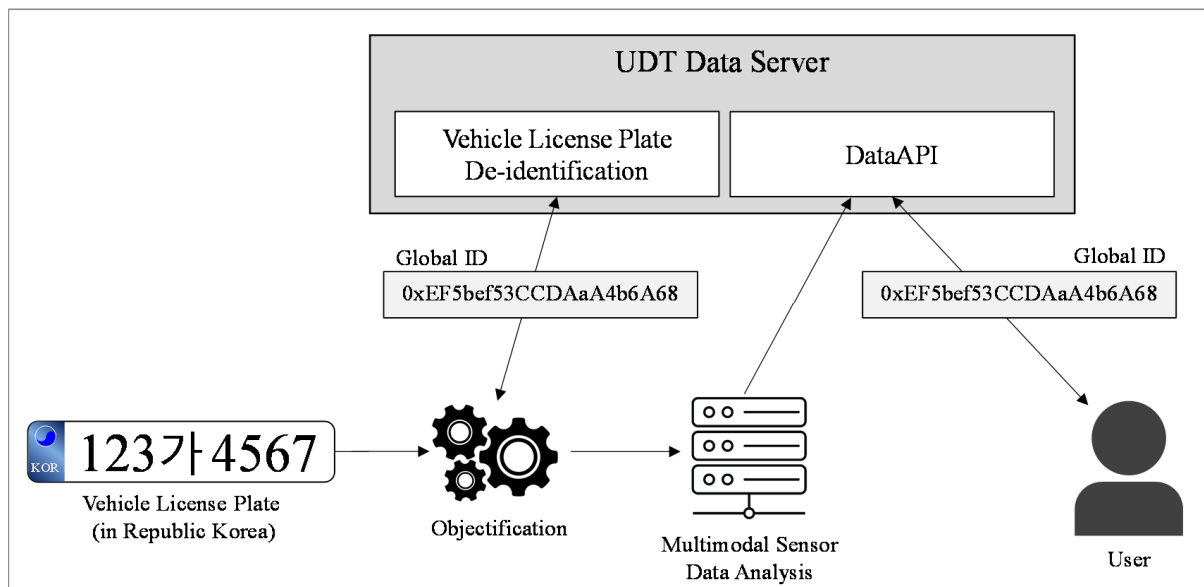


Figure 11. Data flow for the vehicle license plate de-identification and global ID generation method.

The objectification module is interlinked with one or more NVRs, and vehicle license plates in CCTV video can be checked using data stored therein. Therefore, the NVR administrator has the right to use and manage the objectification module. Once the objectification module detects a license plate, the latter will be given a global ID through the vehicle license plate de-identification module in the UDT Data Server. The objectification module will then store and manage license plates and global unique IDs via an interlink. However, only the objectified data excluding the license plates, as shown in Table 2, will be transmitted to the multimodal sensor data analysis module. A user will be able to view the movement and trajectory information of a specific vehicle using a global ID, or use the former to simulate traffic; however, the objectification module or NVR will not be accessible and the license plates of the vehicle will thus not be able to be checked. The data that a user will be provided with through the UDT data server are analyzed in Table 3. The proposed method is based on the following principles:

- De-identification of the license plate will only be possible through the proposed vehicle license plate de-identification module in the UDT Data Server. The de-identification method is being kept private for security purposes.
- The objectification module manages the license plate in conjunction with the global ID.
- Conditions to access the objectification module accessor are the same as those needed to access the NVR. Regulations in the Republic of Korea stipulate that the police are required to be present when a person views such video that they did not film themselves.
- Users of the UDT Data Server can check the long-term movement and trajectory information of one vehicle using the global ID but will not be able to read its license plate.
- If access to NVR is permitted, the movement information on a specific vehicle can be provided by checking the global ID linked to the license plate.
- There are two possible ways to find license plate information linked with the global ID: (1) browsing the list in the objectification module or (2) uploading all possible number

plates that can be combined to the UDT data server through the objectification module and receiving all types of global IDs that can be created. To prevent this, the objectification module can only make a limited number of requests to the UDT data server.

- When the vehicle license plate de-identification module is changed, all vehicle license plates and global IDs recorded in the past in the objectification module must be updated. Otherwise, the global ID will only be valid from the time of issuance until the vehicle license plate de-identification module is changed.

The proposed de-identification method associates a global ID with the license plate of a vehicle, while making it impossible to determine the latter in reverse using the global ID. The global ID is matched 1:1 with the license plate. However, if the method for issuing a global ID is disclosed, the vehicle license plate can be calculated using the global unique ID in reverse. In this study, the UDT Data Server independently managed the module that received the vehicle license plates and output the global ID to prevent this. The method of generating a global ID is encrypted so that it cannot be reversed; therefore, the only way to estimate license plates is to create all possible combinations of license plates and global IDs. A license plate can be obtained by searching for a value that matches with a global ID from the entire list in the table; however, the global ID issuance function is limited by the server. In addition, because the de-identification method is also private on the server, the entire table list cannot be created from outside. To create the global ID, a unique key must be authenticated by the UDT data server, and only the objectification module connected to the NVR through the local network can receive the unique key. Moreover, those who can access the NVR input module can access the original CCTV videos. Therefore, a person who could track license plates in reverse will not need to attempt to infer backward, because they will already have access to the personal information related to the license plates. The UDT Data Server does not store the vehicle license plate numbers as this is classified information. Only a global ID and the vehicle model, movement route, and situational analysis information based on continuous frames are stored, as shown in Table 3.

3.5. Visualizing the Urban Digital Twin Model

As previously noted, the platform proposed in this study was implemented using Unity3D, the most widely used game engine in the world, because of the accessibility and extensibility of its various internal or external libraries. Unity3D is also supported across more than 25 platforms and development environments based on virtual or mixed reality equipment. We developed a UDT platform using a VWorld-based geospatial engine library [19]. The UDT platform implemented provides planetary-scale geospatial rendering. Figure 12 shows an example of our implemented UDT model visualization based on Unity3D.

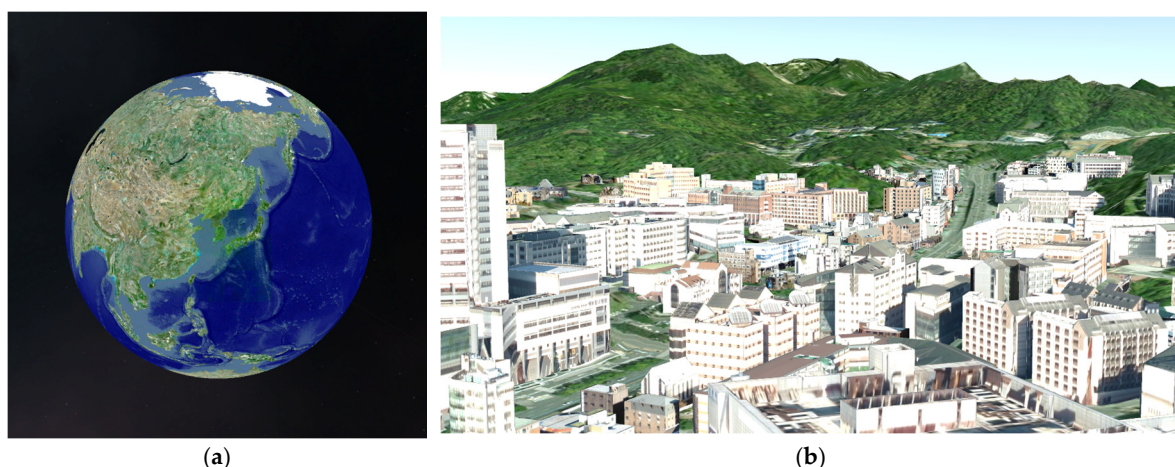


Figure 12. Examples of implemented planetary-scale UDT model visualization: (a) Earth visualized in planetary-scale; (b) close-up camera shot of the surface of Earth, rendering 3D terrain, buildings, and road layers.

We implemented the proposed visualization platform as a Unity3D asset library so that it could be utilized in various UDT platforms. Creating a UDT model will be as simple as importing our proposed asset library into a Unity3D project. The main tasks will be to control the camera and requests, and to manage the tile-based geospatial data based on the position and direction of the controlled camera. The data layers will be activated according to the application used to visualize the UDT model. Additional functions can easily be accessed using Unity3D, which has several development samples, depending on the UDT model visualization library implemented.

The tile-based data management method in the UDT visualization engine is based on a quadtree-based tile attribute or principle. The VWorld data used on this platform provide approximately 30 TB of geospatial data. Planetary-scale data can be visualized by selecting the target tile required based on the position of a camera. All types of geospatial data used to visualize the UDT are streamed; however, real-time rendering is possible because only necessary data are retrieved and requested quickly. Each tile determines whether it is visible. First, camera back-face culling and camera frustum tests are performed on 50 level 0 tiles to determine whether the target tile should be drawn on the screen. The level of a target tile to be displayed is determined based on the distance between the center of the tile and the center of the camera. For example, lower-level tiles are generated when the camera is close to the ground; however, because all tiles have the same aerial image and DEM resolution, geospatial resolution can be maintained above a certain quality even in this case.

The principle of tile-unit-based data management is to draw the data contained in a tile of the rendering target. However, depending on the properties of a geospatial data layer, data that the rendering target tile does not contain may be drawn. In Table 6, layer-rendering properties are classified according to layer type.

Table 6. Rendering properties according to data layer.

Layer	Type	Rendering Properties
Aerial	image	C
Building (XDO)	image	SU2AL
Building (B3DM)	image	C
Bridge	image	SU2AL
Area boundary	image	C2AL
Administrative name	text	C2AL
Administrative boundary	image	C
Road	text	SU2AL
Facility name	text	C2AL
User model	3D model	A or SU2SL
User marker	image, text	A or SU2SL
Individual mobility	text	A or SU2AL

Note: C (current level), C2AL (from the current level to all lower levels), C2SU (from the current level to a specific upper level), SU2AL (from a specific upper level to all lower levels), SU2SL (from a specific upper level to specific lower level), and A (all level).

For example, aerial images are classified as C (current level) because they use only data that the render target tile has. Methods to render country or city names are different, and level 0 tiles manage country names. When the camera approaches the surface of the globe, multiple lower-level tiles are rendered instead of a single level 0 tile to resolve data. In this case, the name of the country included in the level 0 tile must also be displayed in tiles below level 1. This gives the rendering property C2AL (from the current level to all lower levels), which renders data from the current level to all lower levels, such as country or administrative names. As the lower tiles are created as shown in Figure 13, text data—including the location of names—are assigned to lower levels.

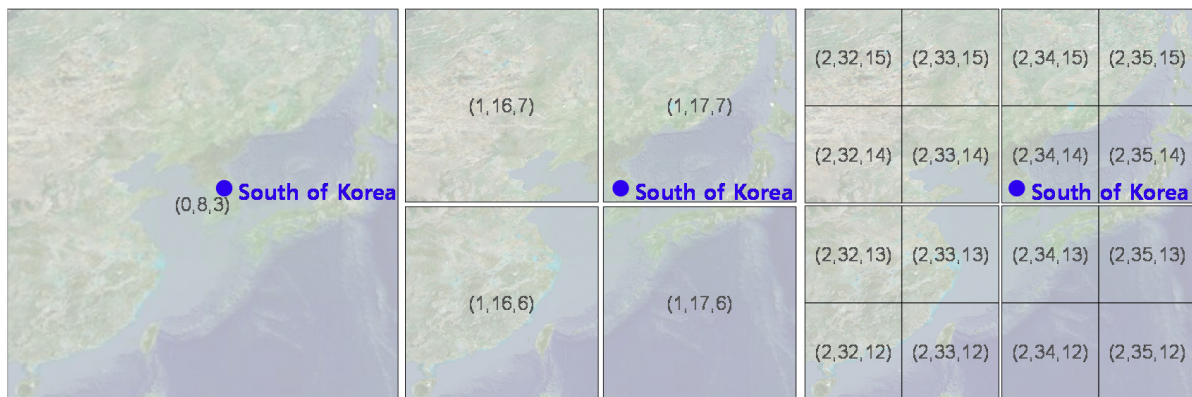


Figure 13. Country name in the level 0 tile with rendering property C2AL given to lower level tiles (level, IDX, IDY).

VWorld tiles are stored and managed at level 0 for country names, level 2 or 4 for city names, level 13 for facilities, and level 14 or 15 for buildings. Individual XDO-type buildings are stored in level 14 or 15 tiles. If only the 3D building models that the rendering target tile has are drawn, the buildings may appear or disappear depending on the changes in camera position and direction. The display level of the tile was determined using the distance between the camera and the center of the tile. The display level changed frequently based on how the user moved the camera. In addition, when there was a great distance between the camera and the ground, the display level of the tile will be higher than the level 14 or 15 tiles that contain the building models. This prevents the building data from being drawn. XDO-type buildings are classified as SU2AL (from a specific upper level to all lower levels). If the specific upper level is set to level 11, all building models owned by the child tiles included in the level 11 tile are searched for and drawn. In B3DM, a building with a different LOD for each tile is stored in a way similar to that of Google Earth. Only the model containing the display tile must be rendered using rendering property C, because the former is a digital surface model-based approach that knows both the terrain and LOD of the building models in advance, according to the level of a tile.

The individual mobility data are drawn at the upper and lower tile levels specified by SU2AL, similar to the rendering property of the building layer. A user can adjust the number of individual mobility objects drawn through using upper tile level setting, based on the limited computing environment. The purpose of this visualization engine is 3D visualization of a UDT; therefore, planetary-scale geospatial information is not always required. The purpose of the implemented platform is that it can be integrated from the small scale of a town to the large scale of a country or continent. At the town scale, we focused on the movements of vehicles or pedestrians. However, at country and continent scales, data on individual mobility objects such as airplanes, ships, or artificial satellites can be used. Figure 14 shows the visualization results of individual mobility objects using our implemented UDT platform.

The area of the testbed was selected, and a 3D point cloud model was constructed via a drone survey. The point cloud survey model was created as a relatively lightweight mesh model. The mesh-up targets are buildings in the testbed and road facilities such as lanes, crosswalks, speed bumps, and curbs. The terrain model was rendered by importing VWorld geospatial data. We also installed a CCTV in the testbed and collected individual mobility objects using our proposed object detection and analysis. Figure 14b shows the results of reproducing the location of the vehicles detected in CCTV in three dimensions.

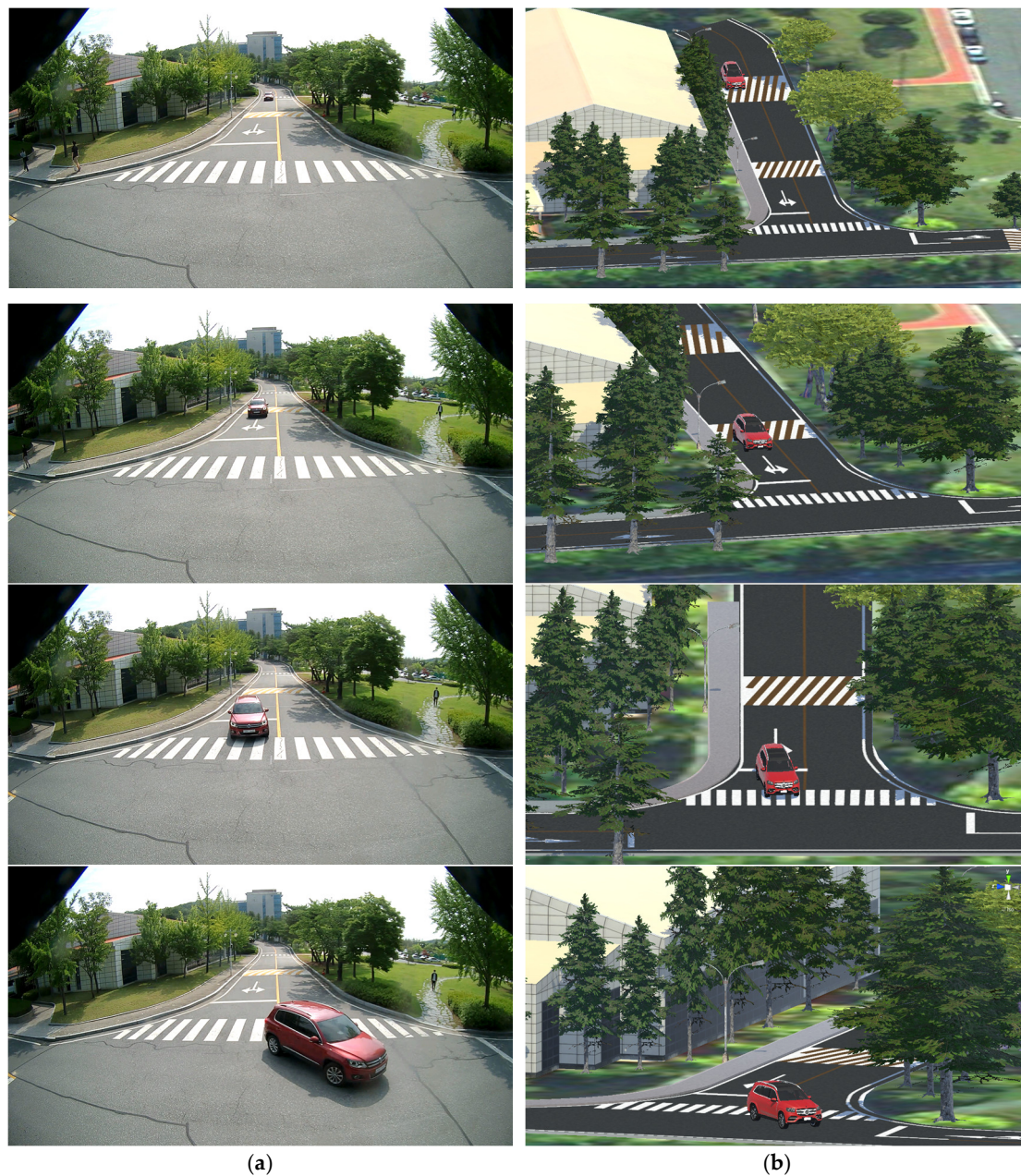


Figure 14. Visualization result for an urban digital twin model with individual mobility objects in our testbed: (a) closed-circuit television video; (b) digitized terrain, buildings, road facilities (e.g., lanes, crosswalks, speed bumps, curbs) and individual mobility objects.

4. Discussion

The proposed UDT platform supports applications based on a digital twin model that is close to reality, including individual mobility data. It is expected that real world changes could be observed if long-term records of the individual mobility of pedestrians and vehicles from the past were stored and managed. As noted in the Introduction, the Hangzhou City Brain—which used the most advanced UDT model—enabled traffic congestion in that city to be reduced by 15.3%. The model also provided a function to quickly search for a specific person or vehicle using approximately 3500 CCTVs.

There are concerns and resistance to privacy violations. The development of information and communication technology has produced side effects such as the indiscriminate distribution of personal information and invasions of privacy. Our proposed platform, which is only in its initial stages, has the potential to be used for the latter. Therefore, in

this study, we proposed a method to de-identify a vehicle license plate using a global ID, and another to manage the IDs. However, the proposed approach is experimental and we believe that the possibility of data leakage or information infringement is still high. However, such concerns do not go beyond the risks related to existing CCTV management systems. Therefore, we only considered the possibility of license plate information leaking within the current system of rights and restrictions related to the ability to access CCTV data. Figure 15 shows the data utilization pipeline of the proposed UDT platform.

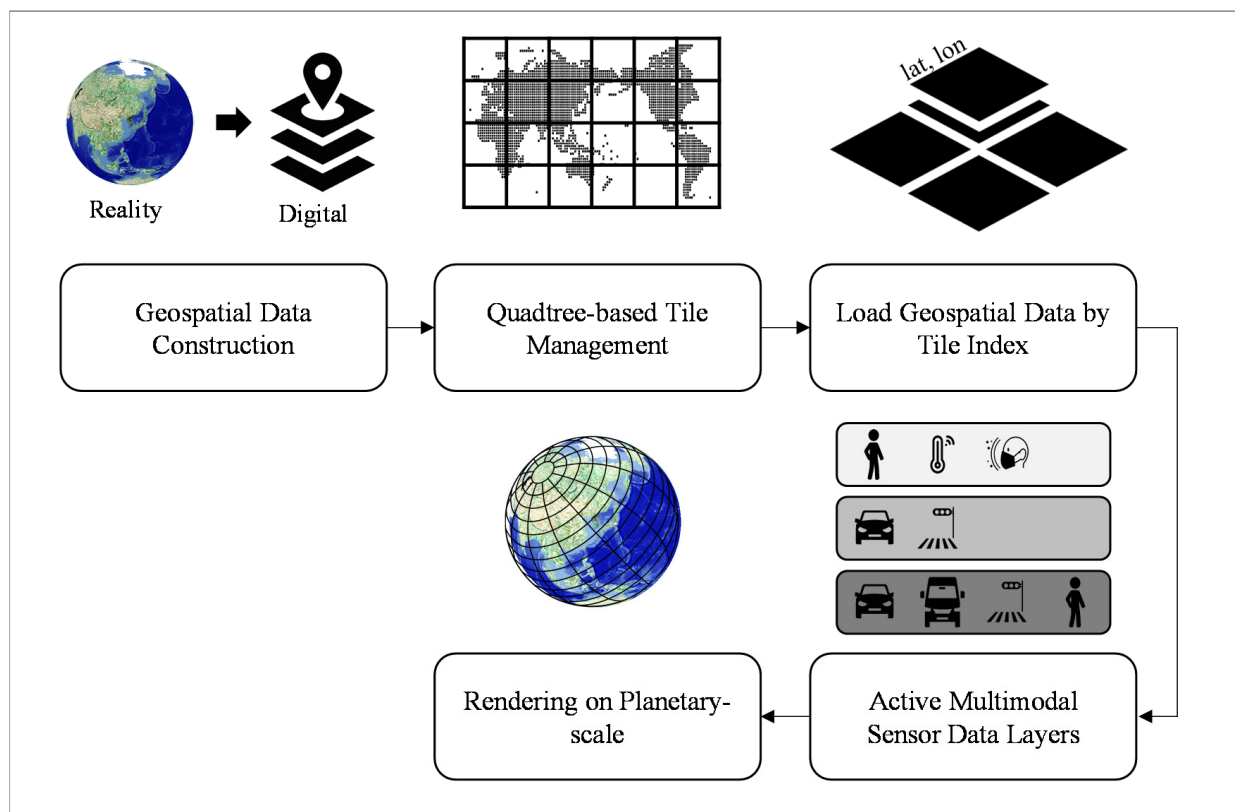


Figure 15. Data utilization pipeline of the proposed UDT platform.

Information in the real world is stored as digitized data using geospatial data construction. Geospatial data that include locations are stored and managed in units of tiles based on the quadtree tile system. Data provided by the platform are requested through the tile index, and a user can activate any multimodal sensor data layer provided as part of their application. The data provided are based on planetary-scale 3D geospatial coordinates and support various uses and applications. The proposed UDT model visualization was implemented with the universal game engine Unity3D, the most common cross-platform development engine. Therefore, users can easily make their own UDT model by utilizing the development environment desired and external libraries.

5. Conclusions

Previous UDT or smart city research has focused on reconstructed the three-dimensional topography of urban spaces. These studies lacked ways to manage individual mobility data, such as those on vehicles and pedestrians. In this study, we designed a UDT platform specialized for individual mobility objects. Unlike previous work on UDT or smart cities, our proposed platform manages data objectification and analysis, and the compression of individual pedestrians or vehicles. The proposed platform constructs static and dynamic geospatial data using a tile-based system and includes data storage and management methods, detection of individual mobility data, and 3D visualization of UDT models. We built a small testbed and presented the results by implementing the proof-of-concept of the

platform we designed. Our result visualizes an urban digital twin model with individual mobility objects in 3D, based on the universal game engine Unity3D. There is a long way to go before large amounts of data on big cities can be managed; however, we believe that our approach can have a substantial global impact on an urban digital twin study.

Author Contributions: Conceptualization, A.L. and K.-W.L.; methodology, A.L.; software, A.L. and K.-W.L.; validation, A.L.; investigation, K.-H.K.; resources, A.L.; writing—original draft preparation, A.L.; writing—review and editing, A.L. and K.-W.L.; visualization, A.L.; project administration, S.-W.S.; funding acquisition, S.-W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by an Electronics and Telecommunications Research Institute Information regarding the funder and the funding number should be provided. Please check the accuracy of funding data and any other information carefully. (ETRI) grant funded by the Korean government. [21ZR1210, DNA based national intelligent core technology development].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boschert, S.; Rosen, R. Digital twin—The simulation aspect. *Mechatronics* **2016**, *59*, 59–74. [\[CrossRef\]](#)
2. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y. Digital twin in industry: State-of-the-art. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2405–2415. [\[CrossRef\]](#)
3. Chen, B.; Wan, J.; Shu, L.; Li, P.; Mukherjee, M.; Yin, B. Smart factory of Industry 4.0: Key technologies, application cases, and challenges. *IEEE Access* **2017**, *6*, 6505–6519. [\[CrossRef\]](#)
4. Wang, S.; Wan, J.; Li, D.; Zhang, C. Implementing smart factory of industry 4.0: An outlook. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 3159805. [\[CrossRef\]](#)
5. Han, S. A review of smart manufacturing reference models based on the skeleton meta-model. *J. Comput. Des. Eng.* **2020**, *7*, 323–336. [\[CrossRef\]](#)
6. Dembski, F.; Wössner, U.; Letzgus, M.; Ruddat, M.; Yamu, C. Urban digital twins for smart cities and citizens: The case study of Herrenberg. *Sustainability* **2020**, *12*, 2307. [\[CrossRef\]](#)
7. Xue, F.; Lu, W.; Chen, Z.; Webster, C.J. From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 418–431. [\[CrossRef\]](#)
8. Gobeawan, L.; Lin, E.S.; Tandon, A.; Yee, A.T.K.; Khoo, V.H.S.; Teo, S.N.; Poto, M.T. *Modeling Trees for Virtual Singapore: From Data Acquisition to Citygml Models*; International Archives of the Photogrammetry, Remote Sensing, and Spatial Information Sciences: Delft, The Netherlands, 2018.
9. Ignatius, M.; Wong, N.H.; Martin, M.; Chen, S. Virtual Singapore integration with energy simulation and canopy modelling for climate assessment. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Tokyo, Japan, 6–7 August 2019; pp. 12–18.
10. Showkatbakhsh, M.; Makki, M. Application of homeostatic principles within evolutionary design processes: Adaptive urban tissues. *J. Comput. Des. Eng.* **2020**, *7*, 1–17. [\[CrossRef\]](#)
11. Ahamed, A.; Vallam, P.; Iyer, N.S.; Veksha, A.; Bobacka, J.; Lisak, G. Life cycle assessment of plastic grocery bags and their alternatives in cities with confined waste management structure. A Singapore case study of Singapore. *J. Clean. Prod.* **2021**, *278*, 123956. [\[CrossRef\]](#)
12. Lai, S.; Loke, L.H.; Hilton, M.J.; Bouma, T.J.; Todd, P.A. The effects of urbanisation on coastal habitats and the potential for ecological engineering: A Singapore case study. *Ocean Coast. Manag.* **2015**, *103*, 78–85. [\[CrossRef\]](#)
13. Gorelick, N.; Hancher, M.; Dixon, M.; Ilyushchenko, S.; Thau, D.; Moore, R. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sens. Environ.* **2017**, *202*, 18–27. [\[CrossRef\]](#)
14. Chen, Y.; Shooraj, E.; Rajabifard, A.; Sabri, S. From IFC to 3D tiles: An integrated open-source solution for visualizing visualising BIMs on cesium. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 393. [\[CrossRef\]](#)
15. Krämer, M.; Gutbell, R. A case study on 3D geospatial applications in the web using state-of-the-art WebGL frameworks. In Proceedings of the 20th International Conference on 3d Web Technology, Heraklion, Greece, 18 June 2015; pp. 189–197.
16. Krämer, M. GeoRocket: A scalable and cloud-based data store for big geospatial files. *SoftwareX* **2020**, *11*, 100409. [\[CrossRef\]](#)
17. Kim, M.S.; Jang, I.S. Efficient in-memory processing for huge amounts of heterogeneous geo-sensor data. *Spat. Inf. Res.* **2016**, *24*, 313–322. [\[CrossRef\]](#)
18. Lee, A.; Jang, I. Implementation of an open platform for 3D spatial information based on WebGL. *ETRI J.* **2019**, *41*, 277–288. [\[CrossRef\]](#)
19. Lee, A.; Chang, Y.S.; Jang, I. Planetary-Scale Geospatial Open Platform Based on the Unity3D Environment. *Sensors* **2020**, *20*, 5967. [\[CrossRef\]](#)
20. Wilhelm, E.; Siby, S.; Zhou, Y.; Ashok, X.J.S.; Jayasuriya, M.; Foong, S.; Tippenhauer, N.O. Wearable environmental sensors and infrastructure for mobile large-scale urban deployment. *IEEE Sens. J.* **2016**, *16*, 8111–8123. [\[CrossRef\]](#)
21. Monnot, B.; Wilhelm, E.; Piliouras, G.; Zhou, Y.; Dahlmeier, D.; Lu, H.Y.; Jin, W. Inferring activities and optimal trips: Lessons from Singapore’s National Science Experiment. In *Complex Systems Design & Management Asia*; Springer: Singapore, 2016; pp. 247–264.
22. Kim, E.; Muennig, P.; Rosen, Z. Vision zero: A toolkit for road safety in the modern era. *Inj. Epidemiol.* **2017**, *4*, 1–9. [\[CrossRef\]](#)

23. Mammen, K.; Shim, H.S.; Weber, B.S. Vision Zero: Speed Limit Reduction and Traffic Injury Prevention in New York City. *East. Econ. J.* **2020**, *46*, 282–300. [[CrossRef](#)]
24. Caprotti, F.; Liu, D. Platform urbanism and the Chinese smart city: The co-production and territorialisation of Hangzhou City Brain. *GeoJournal* **2020**, 1–15. [[CrossRef](#)]
25. Yuan, Y. Application of Intelligent Technology in Urban Traffic Congestion. In Proceedings of the 2020 International Conference on Computer Engineering and Application (ICCEA), Guangzhou, China, 18–20 March 2020; pp. 721–725.
26. Kang, S.; Lee, J. Developing a tile-based rendering method to improve rendering speed of 3d geospatial data with html5 and webgl. *J. Sens.* **2017**, *2017*, 9781307. [[CrossRef](#)]
27. Shuai, B.; Berneshawi, A.; Li, X.; Modolo, D.; Tighe, J. SiamMOT: Siamese Multi-Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Event, 19–25 June 2021; pp. 12372–12382.
28. Singh, G.; Akrigg, S.; Di Maio, M.; Fontana, V.; Alitappeh, R.J.; Saha, S.; Cuzzolin, F. Road: The road event awareness dataset for autonomous driving. *arXiv* **2021**, arXiv:2102.11585.
29. Bishop, C.M. Pattern recognition. In *Machine Learning*; Elsevier: New York, NY, USA, 2006; pp. 1–31.