A Method for Constructing and Managing Level of Detail for Non-Closed Boundary Models of Buildings

Ahyun Lee
Department of Metaverse & Game, Soonchunhyang University, South Korea

***Abstract*—An urban digital twin (UDT) involves creating a virtual three-dimensional (3D) digital replica of a real-world city. To build a UDT model, it needs a comprehensive 3D representation of the city's terrain, buildings, and infrastructure. In order to effectively visualize and manage large-scale spatial data in 3D, it is essential to establish and maintain an appropriate level of detail (LoD) for the 3D model. This study proposes to construct and manage LoDs for VWorld building data. However, since buildings are often composed of non-closed boundary models, applying a quadric mesh-based simplification algorithm may result in the deletion of meshes containing important contour information that defines the shape of the building. To overcome this problem, this paper proposes to use a geometric filtering algorithm to preserve the building outline shape.**

***Keywords*—GIS; digital twin; 3D map; level of detail**

I. INTRODUCTION

Urban digital twinning (UDT) is the study of building a virtual digital twin of a city [1-3]. UDT requires the digital construction of three-dimensional (3D) geometry of terrain, buildings, and facilities. Digital twins digitize parts or machines by attaching sensors to them and synchronize them in real time [4-5]. Digital twins can simulate real physical products or processes by digitally modelling them, and are being used in many fields beyond manufacturing to increase productivity and improve product design and operations. Urban planning or road management operations require UDT models that model realistic 3D cities and work in conjunction with multimodal sensor data.

Google Earth [6] is a web-based 3D map. It is built with HTML5 and WebGL standards, making it easy to visualize 3D cities using only a web browser. Cesium allows users to upload their models to a cloud server and visualize customized 3D buildings or models on a 3D map [7]. All of the above services are global and require large-scale data, so they do not store data locally, but provide real-time streaming-based services.

In order to stream large-scale UDT models in real-time, a data level of detail (LoD) is required [8, 9]. The LoD sets a different amount of data for different levels of the model. For example, if a camera is located outside of space and looking at the Earth, it would be impossible to request, download, and render all the data contained in that region in real time. In the case of VWorld, which has built the most accurate urban 3D spatial information data in South Korea, it provides terrain data of the global Earth's surface and has a data scale of about 30 TB or more, including 3D buildings and facilities in South Korea [10].

VWorld data provides digital elevation models (DEMs) for terrain models and building models independently. Google Earth use digital surface models (DSMs) [11]. It does not distinguish between buildings and terrain as shown in Fig. 1(a). Even trees and cars which are placed on the streets are included in the DSMs. However, for some major buildings in a city, independent models are sometimes provided. On the other side, VWorlds provides all terrain and building models in separately. The terrain is provided by a DEM [12] and an aerial image, so only the terrain excluding buildings and facilities can be visualized, as shown in Fig. 1(b) and (c). This way has the advantage of being able to visualize buildings with higher precision than DSMs. However, because DEM provides additional building data compared to DSM, it requires a relatively large amount of data to be streamed.

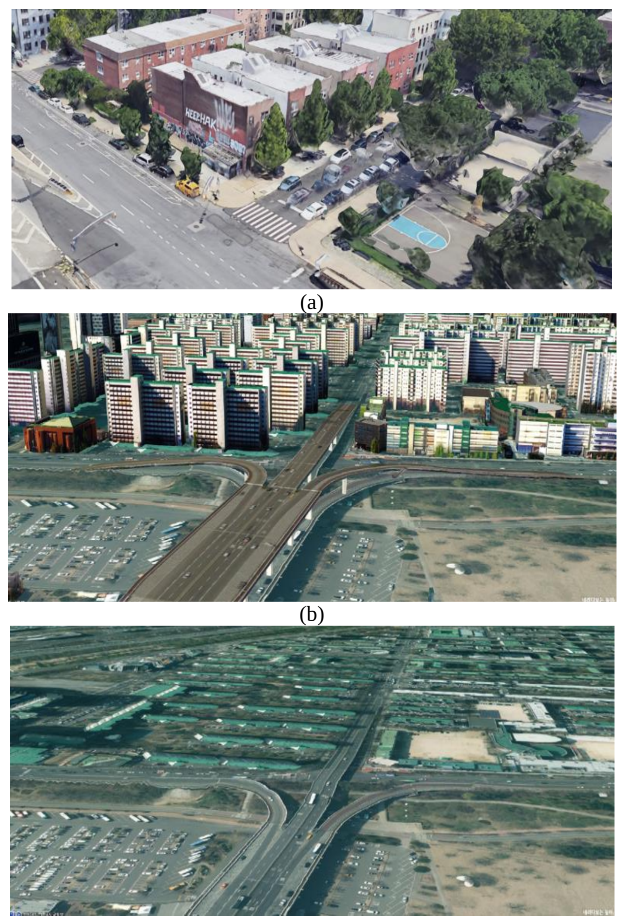


Fig. 1. Differences between DSM and DEM: (a) Google earth DSM, (b) VWorld DEM, and (c) Disabling 3D buildings in (b).

VWorld is built with a total of 16 levels of terrain LoDs [10]. All terrain LoDs have a DEM resolution of 64x64 and a texture image resolution of 256x256. By determining the rendering LoD based on the distance of the camera from the terrain, the texture image resolution and the size of the render data buffer can be maintained regardless of the camera position. However, for building data, only the texture image LoDs are built, which requires mesh simplification for model data consisting of many meshes.

This paper proposes a method for constructing and managing LoDs for non-closed boundary mesh models of buildings. The method utilizes VWorld building data in the form of floorless buildings, which can lead to mesh simplification that compresses or distorts crucial mesh information comprising the building in the outer regions of the non-closed boundary model [13]. Our proposed approach employs geometric filtering for non-closed boundary meshes to retain the primary elements constituting the building facade during simplification. It constructed three levels of LoD in total. Additionally, users are provided the flexibility to adjust the compression rate and a step of LoD according to their preferences.

II. Spatial Information Data Properties

The spatial information data used in this paper is VWorld data [10]. VWorld provides a planetary-scale global three-dimensional terrain model. It consists of a total of 16 levels of terrain LoDs. LoD 0 tiles are separated at intervals of 36 degrees latitude and 36 degrees longitude. There are five latitude levels and 10 longitude levels, for a total of 50 levels that separate the Earth's surface. The 3D surface outlined in orange in Fig. 2 is {Level: 0, IDX: 8, IDY: 3}. At LoD 0, IDY is the latitude level and ranges from 0 to 4, and IDX is the longitude level and ranges from 0 to 9.

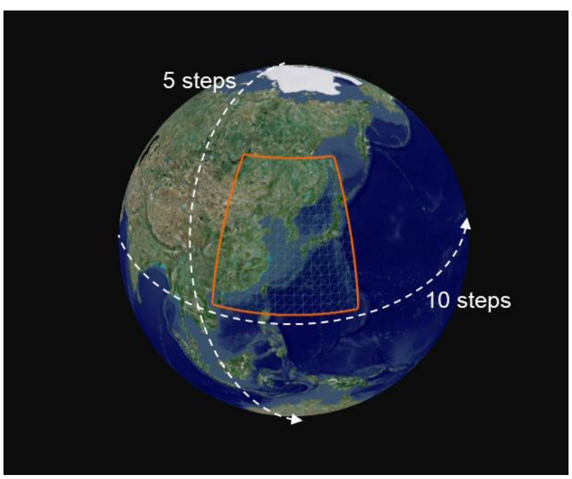


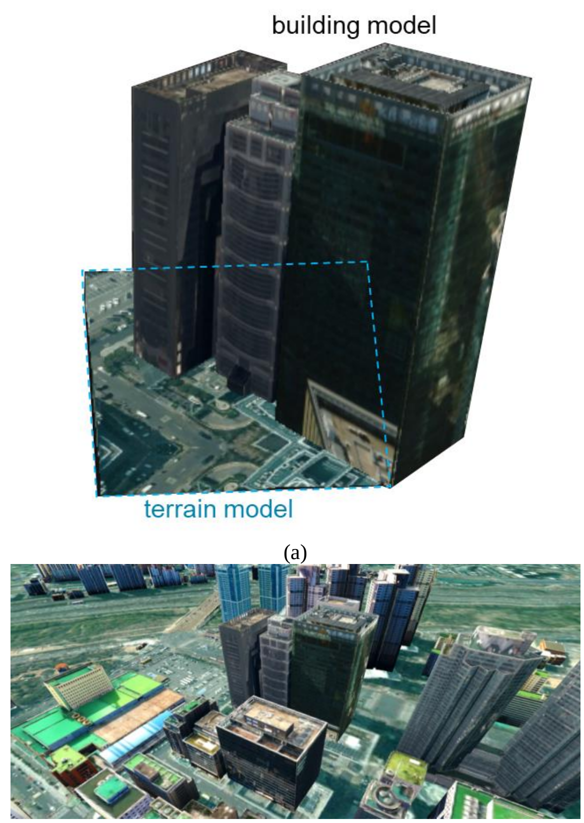
Fig. 2. VWorld tile structure, where the orange surface is a tile in LoD 0 (level: 0, IDX: 8, IDY: 3).

When a single LoD 0 tile is quartered, it increments level 1 and becomes LoD 1. All tiles have the same texture image resolution. As a result, the rendering resolution increases when representing the same-sized area with multiple high LoD tiles. The distance of the camera from the ground can determine the level of terrain tiles rendered, which in turn determines the number of tiles rendered per frame [14]. If the number of tiles to be rendered is similar, the resulting resolution and data buffer can be kept constant across different stages of LoD construction.

A tile is the smallest unit of geospatial data. All geospatial data is attributed to a tile ID based on level, IDX, and IDY. The criteria for inclusion in a tile are the level and location of the tile according to its data attributes. When a tile is visualized, the data contained in the tile is requested based on the active layer. For example, if a terrain tile is being rendered and the building's layer is active, the building data contained in the tile is requested, fetched, and rendered.

Tiles containing buildings use the center point coordinate values from the building data. In this case, the center point coordinates are defined as the center point by averaging the maximum and minimum values of the coordinates. The level of the terrain tile containing the building can be defined according to the situation. VWorld's geospatial data includes buildings at tile level 15 and large facilities such as bridges at level 14.

As shown in Fig. 3(a), when the corresponding tile is rendered, the terrain model of the tile is visualized to manage the building data to be rendered. Once the visualization system determines which tiles and layers to render, each layer contained in each tile is rendered. Once the cluster of tiles that make up the Earth's surface is determined, as shown in Fig. 3(b), the buildings that each tile contains are rendered.



(b)

Fig. 3. Examples of building rendering results in tiles and the implemented visualization SW: (a) One tile and the building model located on that tile and (b) The building rendering result with the surrounding terrain.

Once the building LoD is built, it can be mapped with terrain tiles based on the LoD step. This paper built a building LoD with three levels of steps. It can be mapped in the form of terrain tile level 13 and building LoD 2, terrain tile level 14 and building LoD 1, terrain tile level 13 and building LoD 0, etc. These mapping relationships can change depending on the usage scenario of the geospatial data. In actual implementation, the amount of data may decrease as the granularity of the LoD increases. However, the number of requests or the number of data to be managed may increase with the granularity. Therefore, it is not always efficient to increase the LoD level in the application phase.

In planetary-scale geospatial applications, the target data is built on a large scale. In the case of VWorld geospatial data, it is about 30TB or more, so it is very limited to utilize it by storing it locally. Therefore, all applications that provide planetary-scale geospatial information provide data in real-time streaming. The way to build the tile-based LoD system and manage the data is effective.

TABLE I. EXAMPLE OF THE REQUEST OF VWORLD DATAAPI

Type	Layer	Level	IDX, IDY	Request URL
Aerial Image	Type	0	0, 0	Request=GetLayer&Layer=tile&Level=0&IDX=0&IDY=0&Key=*
DEM	Type	1	1, 1	Request=GetLayer&Layer=dem&Level=1&IDX=1&IDY=1&Key=*
Building	Type	15	139689, 58037	Request=GetLayer&Layer= acility_build&Level=15&IDX=139689&IDY=58037&Key=*

The data request depends on the tile index and layer. Table I shows an example of using the VWorld Data API. Enter the data type in the Layer name and the tile index, {level, IDX, IDY}. You can get an authentication key from the VWorld server service site, and enter your personalized authentication key instead of "*" in the request URL tab in Table I. The request URL is categorized by tiles. A tile is the basic unit of VWorld geospatial data and the basis for management.

Data requests are prioritized. The priority by type is aerial image and DEM. Since the latitude/longitude range of a tile is determined by the tile index based on the elevation value, a regular grid mesh model can be created using DEM. By mapping aerial imagery to the generated terrain mesh model, you can create a 3D terrain model. Terrain models are prioritized in data requests over buildings. In particular, in the case of VWorld, building models are requested after all terrain models have been requested, as some models can be over 10MB per building. In general, web-based development limits the number of simultaneous requests that can be made from the web.

Fig. 4 shows the coordinates system of a building. Typical game engine-based three-dimensional applications use the y-axis as an elevation axis in a plane relative to the x and z axes [15]. However, in a spherical model of the Earth, no particular axis can be used as an elevation axis. On planetary-scale maps, if the origin of the coordinate system is the center of the Earth, the elevation axis of any building model must be the coordinate vector of the building's center point in order for the building to stand upright on the surface of the Earth. The orientation vector in blue becomes the reference vector on which the building is built. All building models in VWorld are pre-rotated according to the center vector.

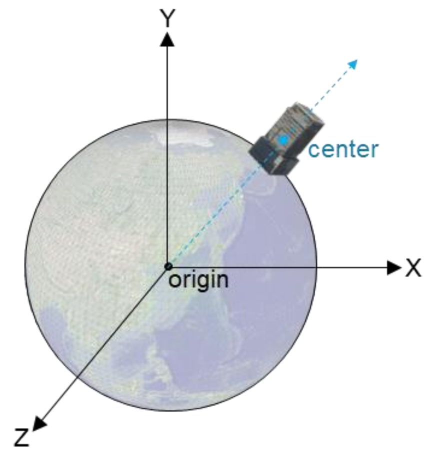


Fig. 4. The coordinates system: A building model is built with the building's center coordinate vector as the reference axis.

For the three-dimensional coordinate transformation used in the proposed method, it needs to convert to longitude-latitude coordinates. Each spatial information coordinate system has a transformation relationship based on longitude/latitude/ elevation. In this paper, it presents a three-dimensional transformation relationship with {*lat*, *lon*, *elev*}, which can be used to establish transformation relationships with other spatial coordinate systems.

$$h = UR + elevl \cdot UR/R \tag{1}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} h \cdot cos\, lat \cdot sin\, lon \\ h \cdot sin\, lat \\ -h \cdot cos\, lat \cdot cos\, lon \end{bmatrix} \tag{2}$$

The geospatial maps implemented in this paper are in the left-handed coordinate system. *UR* is the radius of the Earth set for Unity3D with limited map size, and *R* is the actual radius of the Earth. This assumes that the Earth is a perfect sphere. In order for a building to be on the surface of the Earth, the building model must be oriented around the y-axis. Rotation and translation are necessary for the rotation of a building that can be located on the Earth's surface, as shown in Fig. 4.

The point in Fig. 5(a) is the coordinates of the center of the model where the building should be located. In the coordinate system origin, there is a model of a building standing on the y-axis. At the center coordinate where the building is to be located, rotate it about the y-axis by the angle between the vector with y = 0 and the - z-axis, as shown in Fig. 5(a). Compute the cross vector between the y-axis and the center coordinate, as shown in Fig. 5(b), and rotate it about the cross vector by the angle between the y-axis and the center coordinate vector of the building. Finally, translate the building by the position of the central coordinate vector to position it on the Earth's surface, as shown in Fig. 5(c).

VWorld's individually provided building models are constructed by extracting and connecting building corner points or feature points from aerial photos. The commonality of these models is that they are all non-closed boundary mesh models [13]. As shown in Fig. 6, there are no building faces that touch the ground surface. Since the model was built to be visualized in combination with the terrain model, it does not have the mesh information of the floor surface. It is rendered on a 3D map with parts of the building buried in the terrain, as shown in Fig. 7.

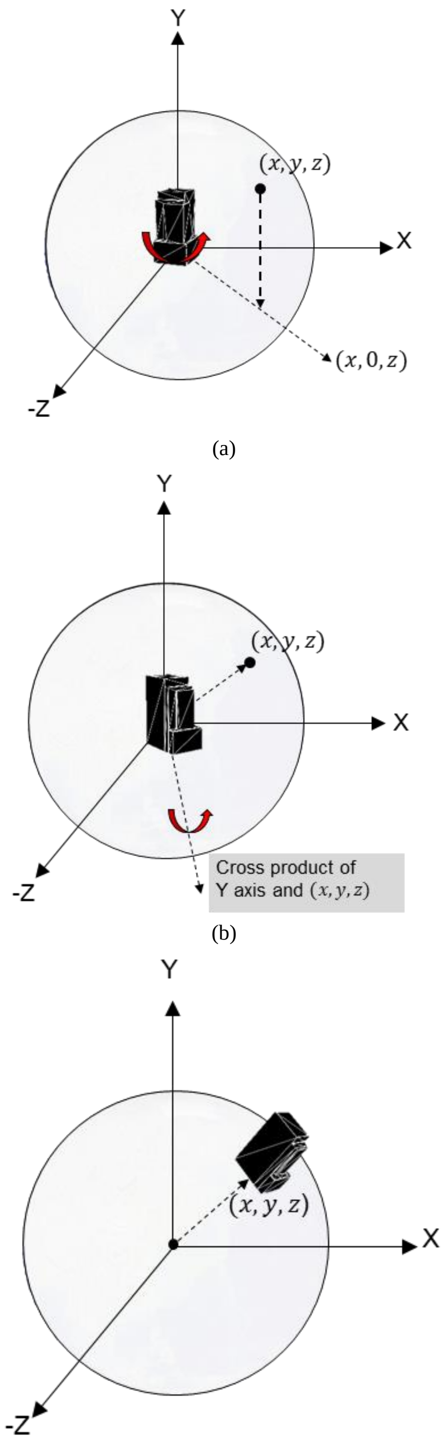


Fig. 5. Rotate and translate to position a building on a spherical surface.

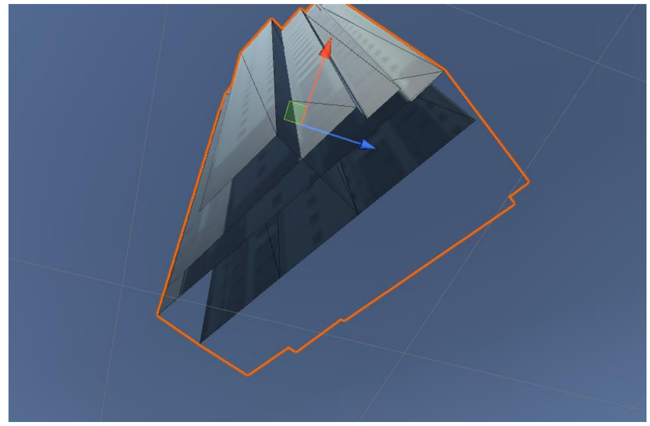


Fig. 6. A Non-closed boundary mesh model with an empty bottom surface.

Fig. 7. Rendering example of buildings as a non-closed boundary mesh model which are partially embedded in the terrain.

III. The Proposed Method

This paper proposes a method for building and managing the LoDs of a building in the form of a non-closed boundary mesh model. To build the LoD, it performs two main simplifications. The first is to reduce the resolution of the texture image of the building. The texture image of a building is stored as a texture atlas [16]. A texture atlas is an image that contains multiple small images, usually stitched together to reduce the overall dimensions. The atlas can be composed of uniformly sized images or images of varying dimensions. In the proposed method, the resolution of the texture atlas is used to construct up to three levels of LoDs.

The image in Fig. 8 has a resolution of 1024x1024. LoD 1 is reduced to 512x512, a quarter of the size, and LoD 2 is reduced to 256x256, 1/16 of the size. The texture image resolution varies depending on the size of the building. It built the LoDs by reducing the original resolution size by a quarter in a step.

The second step is to simplify the mesh model of the building. Compared to the texture image, which can be used to build the LoD easily, simplifying the building mesh model requires a lot of steps. Basically, the model is simplified by reducing the number of vertices that make up the mesh model. The focus is to ensure that the edge information of the building does not change significantly. This paper constructs the building LoD in three steps for the building image texture and three steps for the mesh model.

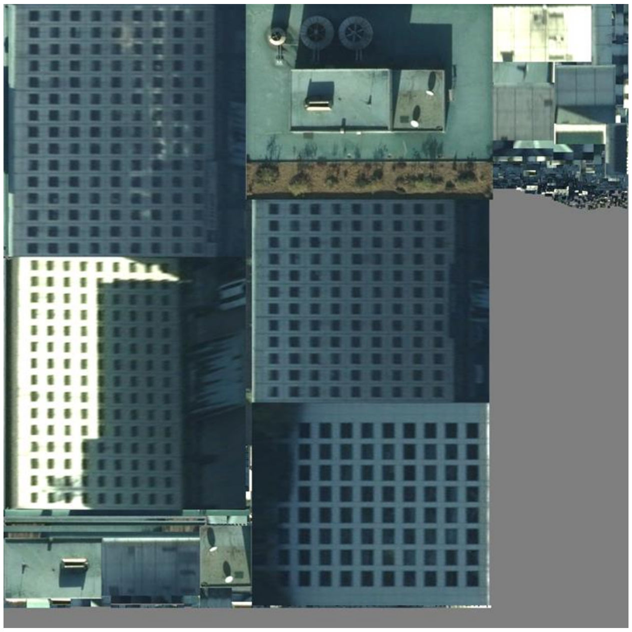


Fig. 8. A texture atlas image of VWorld data.

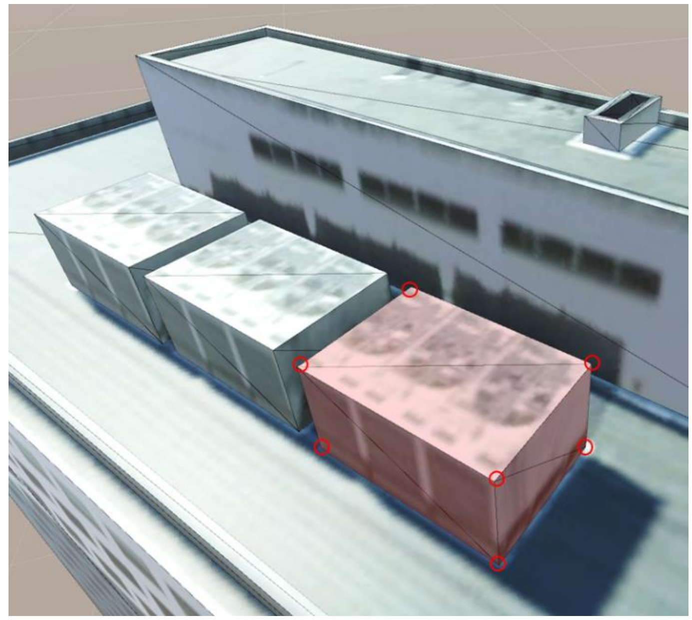
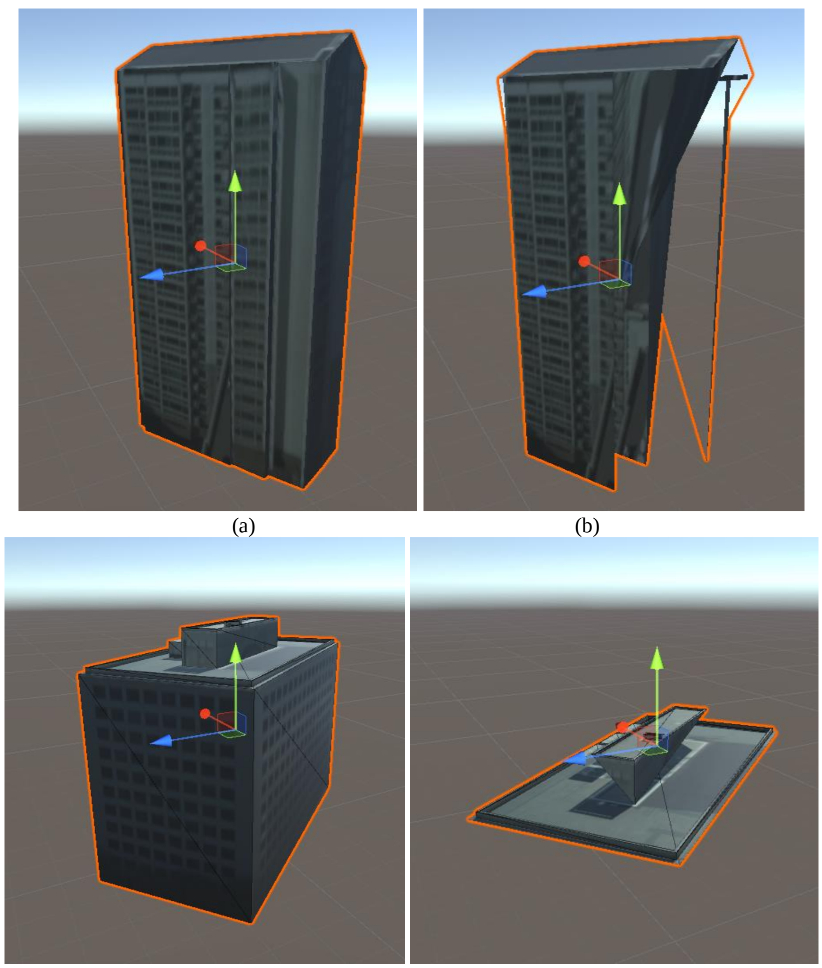


Fig. 9. Example of preserve border edges.



(a) (b)

(c) (d)

Fig. 10. Examples of failed simplification results for a non-closed boundary mesh model.

For the building LoD construction, it uses the fast quadric mesh simplification algorithm [17] to maintain the shape of the mesh model to fit the building characteristics. If it needs to preserve border edges as shown in Fig. 9, it preserves edges that do not share two triangles by default [18]. Otherwise, it removes the target vertices through simplification. This property prevents holes and strange border artifacts from appearing.

However, buildings with non-closed boundary mesh models, such as those shown in Fig. 10, sometimes disappear to the sides or lose key mesh information about the building. If the mesh lost by the simplification method is a large percentage of the building, the outline of the building can be deformed and look like a rendering error. So, it proposes a building simplification and LoD construction method that can be applied to buildings with non-closed boundary mesh models.

The building mesh model simplification method proposed in this paper is shown in Fig. 11. The building LoD 2 is the original model data, and the model data gets lighter as go down to LoD 0. The proposed method utilizes geometric properties such as the orientation vector and size of the building model to distinguish important mesh information in the building. For example, a simple rectangular shape, the side of a building, is a simple shape that uses two meshes, but is important to maintain the shape of the building. The meshes that make up this building's outline information are filtered out of the simplification target so that the outline shape of the building can be maintained.

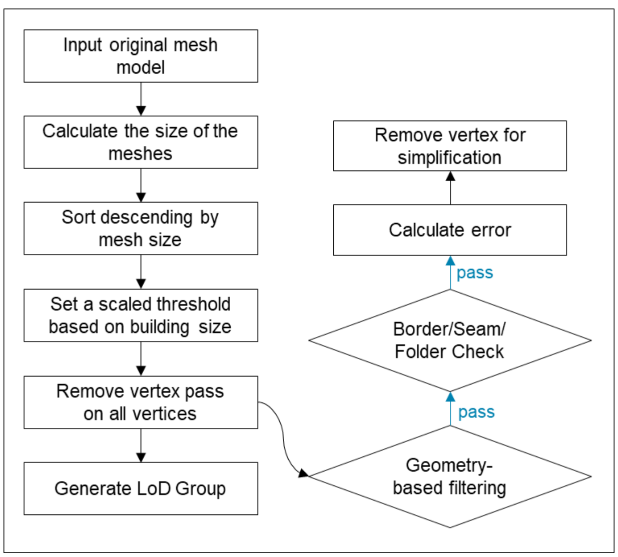


Fig. 11. The proposed simplification methods for non-closed boundary mesh models.

The proposed method first measures the size of all mesh polygons that make up the building model. Sort the measured sizes in descending order and set a threshold value based on the building size. This threshold value is the criterion for excluding meshes from the removal process, depending on the mesh simplification method. Meshes that are larger than the threshold is not eligible for simplification.

Due to the nature of V World building data, large faces such as building sides and roofs are relatively simple shapes. It tested 1248 building models and set the threshold to 30%, which means that the largest 30% of the mesh models or vertexes that make up the building are excluded from simplification. The border/seam/folder checks determine whether the target meshes and vertices are simplified or not. The relevant parameters are shown in Table II.

The proposed simplification algorithm results in the final LoD group shown in Fig. 12. LoD groups are used to manage data that may be displayed depending on the distance of the game object from the camera [19]. Unity3D provides a LoD group feature that allows you to manage LoDs within the game engine. As the LoD increases, the model's vertex, mesh, and texture image resolution decreases, resulting in less detail. However, as the camera gets farther away, the difference in detail is not noticeable to the user. When rendering large amounts of building data simultaneously, this approach can maintain frame per speed by reducing the size of the render target. You can determine the level of LoD visibility based on the camera distance in the properties.

TABLE II. BORDER/SEAM/FOLDER CHECK PARAMETER

Option	Description
Preserve Border Edges	Border edges that need to be preserved are edges that do not share two triangles.
Preserve UV Seam Edges	UV seam edges that need to be preserved are essentially connected edges that contain a difference in UV coordinates.
Preserve UV Foldover Edges	UV fold edges that need to be preserved are connected edges that contain the same UV coordinates.
Preserve Surface Curvature	When you need to preserve surface curvature
Vertex Link Distance	Distance between vertex links
Max Iteration Count	The maximum number of iterations to simplify the mesh. Reducing this value can speed up the simplification, but may reduce the quality of the result. This value is used to prevent infinite simplification from occurring.

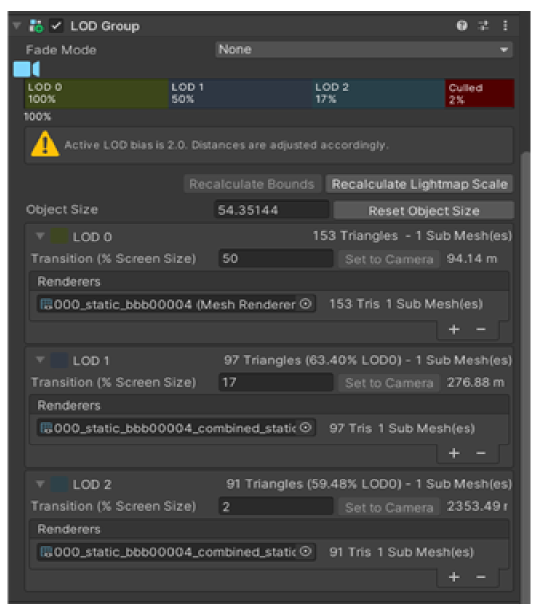


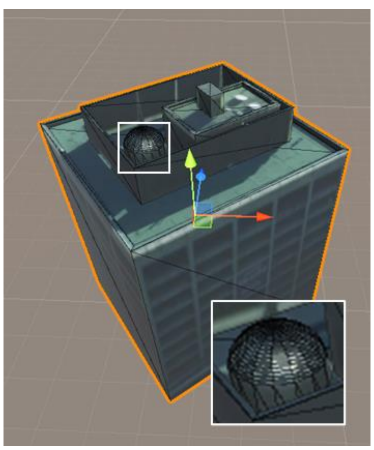
Fig. 12. The result of the created a building LoD group in the unity 3D editor.

It conducted experiments on 1248 random buildings provided by VWorld. Table III shows the simplification results generated by the proposed algorithm. LoD 0 is the number of meshes in the original model, and LoD 1 and LoD 2 are the number of meshes in the simplified result. Percentage means the ratio of the number of meshes in the original LoD 0 to the number of meshes in the simplified result. The reduction through simplification is not significant due to the low number of original meshes. For buildings with complex shaped meshes, size reductions of up to 80% or more have been observed.

Fig. 13 shows the simplified results. Compared to Fig. 13(a), the number of meshes for the semi-spherical shape of the sculpture in Fig. 13(b) and Fig. 13(c) is reduced. On the other hand, you can see that the important outline and corner information that make up the shape of the building is retained. This also avoids the problem of excluding information such as the sides of the building, which can occur when simplifying on non-closed mesh buildings. The building consists of 559 meshes at LoD 0, and the simplification results in 363 meshes at LoD 1 and 236 at LoD 2, a reduction of 64.94% and 42.22%, respectively.

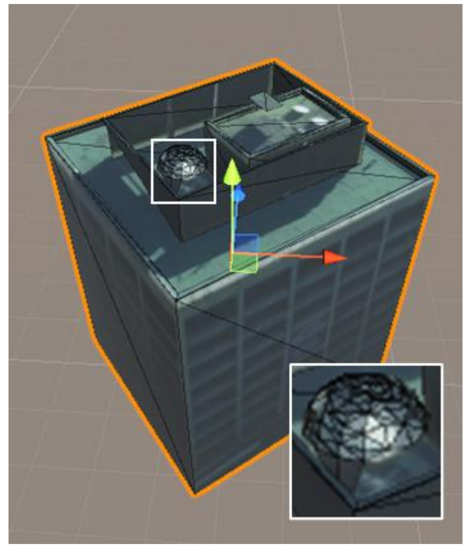
TABLE III. SIMPLIFICATION RESULTS FOR 1248 BUILDING DATA FORM VWOLD GEOSPATIAL DATA (MESH COUNTS, AVERAGE VALUES)

Mesh Count		
LoD 0	**LoD 1**	**LoD 2**
326.29 (100%)	220.14 (67.46%)	156.86 (48.07%)



(a)

(b)



(c)

Fig. 13. Building model simplification results: (a) LoD 0, 559 meshes, (b) LoD 1, 363 (64.94%) meshes, and (c) LoD 3, 236 (42.22%) meshes.

IV. CONCLUSION

This paper proposes a LoD construction and management method for a non-closed boundary model of buildings. VWorld geospatial data provides terrain and buildings as independent models. LoD construction experiments are built for each building. As a result of the experiments, it constructed three levels of LoDs for 1248 buildings in downtown Seoul and found that the number of meshes could be reduced by 67.46% in LoD 1 and 48.07% in LoD 2 through two steps of simplification. The problem of removing meshes at non-closed boundary locations in the simplified building model was also solved by filtering based on the geometric properties of the building model. In future research, I plan to study a simplification accuracy measurement method based on geospatial data to measure the quality of simplified building models. I also plan to apply it to the VWorld spatial information model to provide a planetary-scale real-time streaming-based service for building LoD data built from 3D maps.

ACKNOWLEDGMENT

This work was supported by the Soonchunhyang University Research Fund.

REFERENCES

[1] C. Weil, S. E. Bibri, R. Longchamp, F. Golay, and A. Alahi, “Urban digital twin challenges: A systematic review and perspectives for sustainable smart cities”, Sustainable Cities and Society, vol. 99, 104862, 2023.

[2] A. Lee, K.-W. Lee, K.-H. Kim, and S.-W. Shin, “A geospatial platform to manage large-scale individual mobility for an urban digital twin platform,” Remote Sensing, vol. 14, no. 3, p. 723, 2022.

[3] S. Ivanov, K. Nikolskaya, G. Radchenko, L. Sokolinsky, and M. Zymbler, “Digital twin of city: Concept overview,” in 2020 Global Smart Industry Conference (GloSIC), IEEE, pp. 178–186, 2020.

[4] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, “Characterising the Digital Twin: A systematic literature review,” CIRP journal of manufacturing science and technology, vol. 29, pp. 36–52, 2020.

[5] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. de J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review", Remote Sensing, vol. 14, no. 6, 1335, 2022.

[6] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google Earth Engine: Planetary-scale geospatial analysis for everyone," Remote sensing of Environment, vol. 202, pp. 18–27, 2017.

[7] F. Tsai, J.-S. Lai, and Y.-C. Liu, "An alternative open source web-based 3D GIS: cesium engine environment," in Asian Conference on Remote Sensing: Fostering Resilient Growth in Asia, Citeseer, pp. 1–4, 2016.

[8] A. Lee, Y.-S. Chang, and I. Jang, "Planetary-scale geospatial open platform based on the Unity3D environment," Sensors, vol. 20, no. 20, p. 5967, 2020.

[9] M. Breunig et al., "Geospatial data management research: Progress and future directions," ISPRS International Journal of Geo-Information, vol. 9, no. 2, p. 95, 2020.

[10] A. Lee and I. Jang, "Implementation of an open platform for 3D spatial information based on WebGL," ETRI Journal, vol. 41, no. 3, pp. 277–288, Jun. 2019.

[11] L. Zhang, Automatic digital surface model (DSM) generation from linear array images. ETH Zurich, 2005.

[12] L. Polidori and M. El Hage, "Digital elevation model quality assessment methods: A critical review," Remote sensing, vol. 12, no. 21, p. 3522, 2020.

[13] L. Yu, Q. Han, and X. Niu, "An improved contraction-based method for mesh skeleton extraction," Multimedia tools and applications, vol. 73, pp. 1709–1722, 2014.

[14] R. Kooima, J. Leigh, A. Johnson, D. Roberts, M. SubbaRao, and T. A. DeFanti, "Planetary-scale terrain composition," IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 5, pp. 719–733, 2009.

[15] A. Kamaludin, P. H. Rusmin, and A. Harsoyo, "Design and implementation educational game of coordinate systems and least common multiple using educational games design model," in 2015 4th International Conference on Interactive Digital Media (ICIDM), IEEE, pp. 1–6. 2015.

[16] C. Allene, J.-P. Pons, and R. Keriven, "Seamless image-based texture atlases using multi-band blending," in 2008 19th international conference on pattern recognition, IEEE, pp. 1–4. 2008.

[17] C. Li, Z. Zhao, W. Sun, and Z. Liu, "A fast quadtree - based terrain crack locating method that accounts for adjacency relationships," Transactions in GIS, vol. 23, no. 6, pp. 1374‑1392, Dec. 2019.

[18] K. Buchin, W. Meulemans, A. V. Renssen, and B. Speckmann, "Area-Preserving Simplification and Schematization of Polygonal Subdivisions," ACM Trans. Spatial Algorithms Syst., vol. 2, no. 1, pp. 1–36, Apr. 2016.

[19] V. Gerasimov, Building Levels in Unity. Packt Publishing Ltd, 2015.