Performance Analysis of GLTF/GLB to Improve 3D Content Rendering Performance

[1]Jae Myeong Choi, [2]*Ki-Hong Park

Abstract

3D content rendering is one of the important factors that give a sense of realism when creating content, and this process takes a lot of time. In this paper, we proposed a method to improve rendering performance by reducing the vast amount of 3D data in the web environment, and conducted a performance test using DEM and 3D model elevation data. As a result of the experiment, the digital elevation model showed faster performance than the Blender-based 3D modeling, but when the screen was moved using OrbitControl, the fps dropped momentarily. In the case of Terrain, if the range is limited to a speed that maintains 24 to 60 fps using frustum culling and LOD techniques, it is considered that a higher quality map can be produced than GeoTIFF.

Keywords: *3D Content Rendering, GLTF/GLB, HTML5/WebGL, DEM, OSM*

I. Introduction

Today, the 3D content industry is becoming more active due to the development and convergence of v arious technologies in the new industries of the 4th industrial revolution. In particular, many studies are being conducted to improve the trade-off relationship between data capacity and processing time when p roducing 3D content and to produce high-quality content when producing 3D content[1][2]. 3D content creation generally goes through modeling, rendering, and animation processes, and especially in the rend ering process, a lot of time is required for rendering to give realism to content.

In the field of geospatial information, HTML5/WebGL, which can provide geospatial information ser vices in all browsers without installing a separate plug-in, is often used[3]. WebGL is a JavaScript API f or expressing 2D and 3D graphics that can be interacted with in a web browser, and provides an API that satisfies OpenGL ES 2.0 used in HTML <canvas> elements. HTML5/WebGL, a next-generation web st andard technology, is used in many studies for high-precision and diverse 3D spatial information service s in various cross-operating systems such as Windows, Linux, and Mac, and cross-browser environments such as Chrome, Edge, Firefox, and Safari[4][5].

Rendering speed is important to smoothly receive services for 3D spatial information using HTML5/ WebGL. In order to improve the rendering speed of 3D content, there are various methods using frustum culling, level of detail (LOD), Web Worker, and Emsctipten.

In this paper, we proposed a method for improving the rendering performance by reducing a huge am ount of 3D data in a web environment, and performed a performance test using DEM (Digital Elevation Model) images. To improve the 3D rendering speed in WebGL, the map was created using Blender's int ernal plug-in, and the altitude and OSM (Open Street Map) data were adjusted and transformed into GL TF/GLB files in the web browser to measure performance.

The composition of this paper is as follows. Section 2 describes related technologies and existing servi ce for rendering, and Section 3 proposes a method for displaying altitude data in 3D space. Section 4 co mpares and analyzes the rendering speed between the proposed method and the method using DEM. Fin ally, Section 5 discusses conclusions.

---

[1] *Mokwon University, Dept. of Computer Engineering, Professor (jmchoi@mokwon.ac.kr)*
[2* Corresponding Author] *Osan University, Dept. of Computer Information, Professor (kihong@osan.ac.kr)*

Received: Aug. 16, 2023, Revised: Aug. 25, 2023, Accepted: Aug. 25, 2023

II. Related technologies and services

2.1. Related technologies

Frustum culling is a technique for filtering out objects outside the camera's field of view in 3D space, and is a technique used in combination with a quad-tree or oct-tree technique. Therefore, 3D spatial data can be efficiently serviced, and it is applied to most 3D spatial information services today.

LOD refers to the overall state of the information model at a specific point in the design process, and is a technology that displays objects differently depending on the distance using the characteristics that distant objects are not easily visible. LOD is divided into static LOD and dynamic LOD according to the storage method. Static LOD, which can be rendered quickly, is used in 3D geospatial information service.

Web Worker is a technology that allows script operations to be executed on a background thread separate from the main execution thread of the web application. In particular, Web Worker can be used for very complex mathematical calculations, access to remote resources, and tasks that need to be continuously performed without interrupting the UI thread in the case of long-time work in the background.

Emscriptenm is a source-to-source compiler that runs as a backend to LLVM (Low Level Active Machine) and generates a subset of JavaScript 'Asm.js'. 'Ams.js' can be precompiled into the browser, which allows precompiled programs to run much faster than traditional methods.

2.2. Related 3D spatial information service

This section describes the services to which various web technologies are applied in relation to the 3D spatial information service. Hera provides City and Map rendering service with 3D terrain, buildings and photo-realistic street view information based on WebGL technology[6]. Google provides 3D geospatial i nformation services through Google Earth Desktop[7] and Google Earth Plug-in-based web services. CE SIUM provides a service centered on simple data to provide a No Plug-in 3D spatial information service using HTML5/WebGL technology[8].

III. Proposed rendering technique

There is a limit to directly processing a vast amount of 3D GIS (Geographic Information System) data. There is a limit to directly processing a vast amount of 3D GIS data. Therefore, in this paper, we propose a method for improving rendering performance by reducing the vast amount of 3D data in the web envir onment. In other words, we propose a method to improve performance such as loading speed by processi ng water, land, and altitude data, which are the main elements that make up a 3D map. 3D GIS production and processing used Blender GIS or Blender OSM. Using OSM-related add-ons, terrain and buildings can be directly imported into models, which is very useful in terms of speed.

The overall procedure of Blender-based 3D modeling presented in this paper consists of three steps and is shown in Figure 1. The first step is to create a map using Blender's internal plugin. After installing the Blender GIS add-on, a provider-specific default map (BaseMap) can be selected. Providers include Microsoft Bing, Google, ESRI, and OSM, and each BaseMap of the provider can adjust the map area through Zoom Level. In addition, OSM data and SRTM (Shuttle Radar Topography Mission) can be used for mesh formation, which is a 3D object. OSM data is divided into Ways, Nodes, and Relations, and the height (or height) is adjusted through various options such as default height, random height threshold, and level height. Here, Ways and Nodes are subdivided into Building, Highway, Landuse, Leisure, Natural, Railway, and Waterway.

In the second step, the user selectively adjusts the altitude and OSM data. In this process, the texture image can be mapped and applied through the UV coordinates of the building data. Also, in the selective adjustment step, an animation such as flowing water may use shading, and a more realistic building may be expressed after applying a shadow or shading effect. In the case of elevation data, normal vectors are used, and user-defined transformations can be performed.

In the last step, altitude was expressed using GLTF/GLB files in the web browser, and 'Three.js' based on WebGL, a JavaScript middleware, was used to express 3D space. To express a 3D space with

'Three.js', it can be expressed in a web browser by rendering lights, cameras, 3D objects, and other spaces. At this time, GLTFLoader can be used to display 3D objects in the web 3D environment.

Most companies that provide GIS services provide DEM images by size, making it easy to build 3D GIS for each user. In this paper, 3D modeling was performed using DEM and Blender to represent altitude data in 3D space.

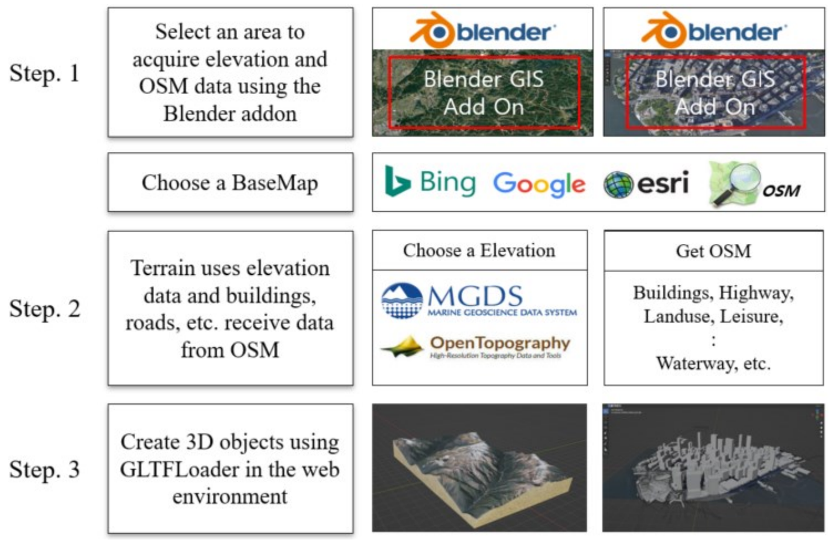

Figure 1. Blender-based 3D modeling process

IV. Experiments and Considerations

For the proposed rendering method, performance analysis was conducted using SRTM images, and the results were visualized when 10 to 40 layers were overlapped based on the size of 512 to build a 3D map. In addition, the use of DEM data measured the method of highly expressing vertex data using WebGL and the rendering time of 3D modeling. The measurement standard was based on constant camera screen movement, and the lowest and highest fps were measured and compared and analyzed.

Figure 2 is the result of a 3D model showing the altitude based on the numerical elevation after texturing the DEM image on the Blender-based PlaneGeometry in the method proposed in this paper. The obtained 3D model was exported in GLTF/GLB format and loaded with GLTFLoader of 'Three.js' for visualization. Figure 3 shows the result of measuring the fps using the GLTF/GLB 3D model.

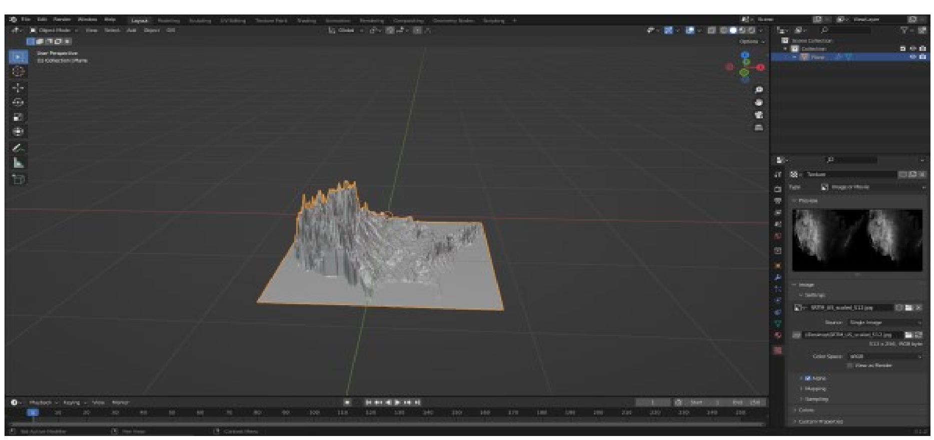

Figure 2. 3D modeling screen using Blender

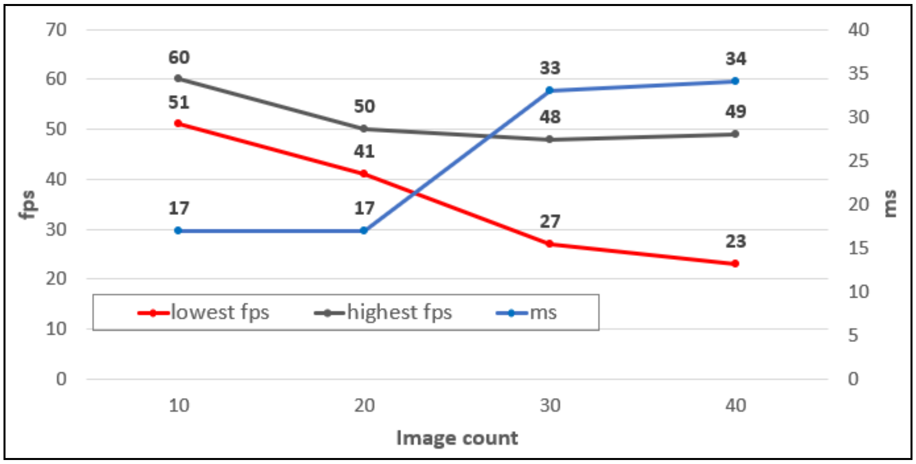

Figure 3. Velocity measurement results using GLTF/GLB format

As a comparison with the proposed method, DEM images were expressed in 3D space using WebGL, and visualization was performed by texturing DEM images on PlaneGeometry. Figure 4 is the expression of 3D GIS altitude data using DEM, and Figure 5 is the result of measuring fps for each number of DEM altitude images.

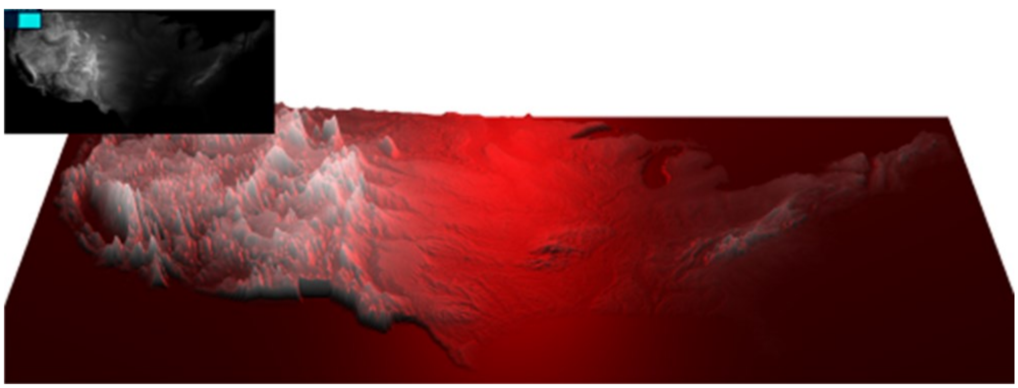

Figure 4. 3D GIS Elevation Data using DEM

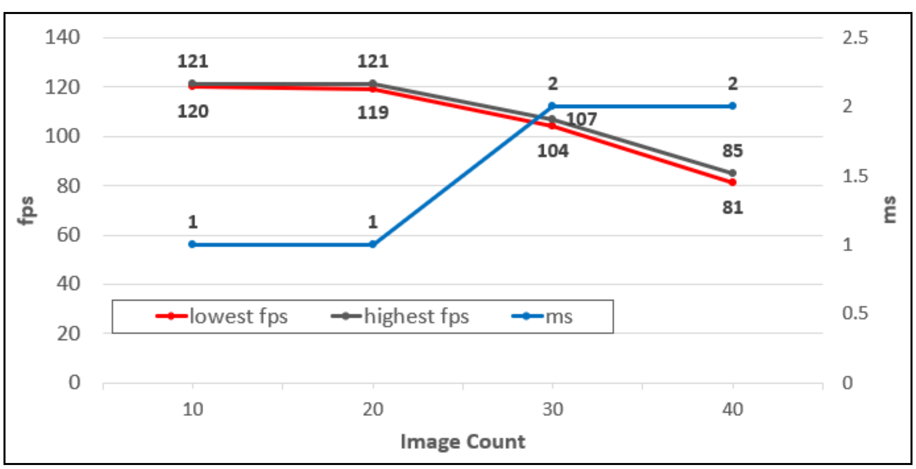

Figure 5. Velocity measurement results using DEM

In the case of modeling using the blender presented in this paper, it was possible to form a 3D model close to the real thing, but it was analyzed that the speed was affected by the constraints according to points, lines, and planes. When using LowPolygon, there is no major problem in rendering even in the web browser environment, but when using HighPolygon, it runs in the web browser, but there is a disconnection. However, it is believed that using Blender within a limited area can provide an environment similar to real life. Figure 6 and Figure 7 show maps and terrains created with GeoTIFF and Blender, respectively.

In order to increase the qualitative factor for building objects, vertex should not be reduced, but in the case of Terrain, there is not much difference even if many vertices are not used. Therefore, in this study, the range was limited to the extent of maintaining the speed of about 24~60 fps, and it is considered that higher quality maps can be produced than GeoTIFF if fps is maintained using frustum culling and LOD techniques.

Figure 6. Produced and processed maps; (left) using GeoTIFF, (right) using Blender

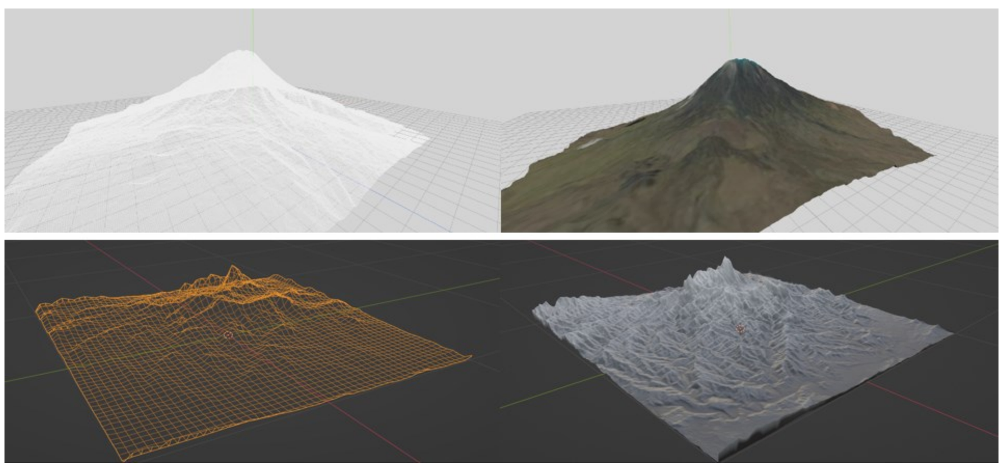

Figure 7. Generated terrains; (up) using GeoTIFF, (bottom) using Blender

V. Conclusions

In this paper, we analyzed the rendering performance of the proposed method using elevation data and the method using DEM information. As a result of the simulation, DEM showed faster performance tha n Blender-based 3D modeling, but when the screen was moved using 'OrbitControl.js', a phenomenon i n which the fps dropped momentarily occurred. Blender-based modeling can form 3D models that are cl ose to real life, but the speed is affected by the constraints of the basic elements, points, lines, and planes. In the case of LowPolygon, there was no problem with rendering in the web browser environment, but i n the case of the HighPolygon option, there was a disconnection while running in the web browser. Ther efore, it is considered that using Blender within a limited area can provide an environment similar to real life. In addition, although the speed of 24~60 fps was maintained when expressing the terrain in the pro posed method if the FPS is maintained using frustum culling and LOD techniques, it is considered that a higher quality map can be produced than GeoTIFF.

VI. References

[1] D. Y. Lee, J. K. Lee, Mir Park, and J. M. Choi, "3D GIS Modeling based on Blender and Performance Analysis," Journal of Digital Contents Society, Vol. 23, No. 10, pp. 2085-2090, Oct. 2022.
[2] S. C. Kang, "Rendering speed improvement of 3D geospatial data based on HTML5/WebGL using tile-rendering," Master's Degree dissertation, University of Seoul, Seoul, Feb. 2017.
[3] D. G. Lee, J. W. Go, and H. J. Lee, " Using Geospatial Information Open Platform for Design and Planning of Route Unused Land," Journal of Korean Society for Geospatial Information Science, Vol. 23, No. 3, pp. 95-106, Sep. 2015.
[4] M. S. Kim, and I. S. Jang, "Design and Implementation of 3D Geospatial Open Platform Based on HTML5/WebGL Technology," Journal of Korea Spatial Information Society, Vol. 23, No. 6, pp. 57-66, Dec. 2015.
[5] G. Qiu, and J. Chen, "Web-based 3D map visualization using WebGL," 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 759-763, 2018.
[6] HERE, HERE Map Rendering [Internet]. Available: https://www.here.com/platform/map-rendering/.
[7] Google, Google Earth [Internet]. Available: https://www.google.com/earth/.
[8] CESIUM, Cesium OSM Buildings [Internet]. Available: https://cesiumjs.org/2015/04/27/3D-Buildings-in-Cesium/.

Author

Jae Myeong Choi

2014.08 : Department of IT Engineering, Graduate School of Mokwon University (Ph.D Degree)
2014.03 ~ Present : Assistant Professor in Dept. Computer Engineering, Mokwon University

Research Interests : Wireless Communication System, Intelligence Disaster System, Societal Safety, Communication Disaster, Multimedia Communication, IoT, Digital Content etc

Ki-Hong Park

2012.03 ~ 2022.02 : Assistant Professor in Dept. Computer Engineering, Mokwon University
2023.04 ~ Present : Korea Institute of Civil Engineering and Building Technology (KICT), Visiting Researcher
2022.03 ~ Present : Assistant Professor in Dept. of Computer Information, Osan University

Research Interests : Digital Content, Computer Vision, Pattern Recognition, Disaster Prevention Application, etc.