

Blender 기반 3D GIS 모델링 및 성능 분석

이 동 엽¹ · 이 주 경¹ · 박 미 르¹ · 최 재 명^{2*}

¹한국전력공사 전력연구원

^{2*}목원대학교 컴퓨터공학과 교수

3D GIS Modeling based on Blender and Performance Analysis

Dong-Yeop Lee¹ · Ju-Kyoung Lee¹ · Mir Park¹ · Jae Myeong Choi^{2*}

¹Korea Electric Power Research Institute, Korea Electric Power Corporation, Daejeon 34056, Korea

^{2*}Department of Computer Engineering, Mokwon University, Daejeon 35349, Korea

[요 약]

3D 콘텐츠는 다양한 기술의 발전과 융합 등으로 양질의 콘텐츠를 제작할 수 있으며 제작 시간도 많이 단축되었다. 그러나 양질의 콘텐츠의 제작을 위해서는 모델링, 렌더링, 애니메이션 과정을 거쳐 생성하는데 사실감을 부여하는 렌더링에는 많은 시간이 소요된다. 또한, 웹 환경에서는 많은 양의 3D 데이터를 전송할 때 전송 지연이 발생할 수 있다. 따라서, 본 연구에서는 웹 3D GIS 렌더링 성능 향상을 위하여 3D 지도를 구성하는 주요 요소인 물, 토지, 고도 데이터를 가공함으로써 로딩 속도를 개선하는 방법을 제안하였고, 3D 오브젝트를 갯수별로 FPS를 측정 및 Web Worker를 사용하여 성능을 분석하였다.

[Abstract]

3D content can produce high-quality content through the development and convergence of various technologies, and production time has been greatly reduced. However, in order to produce high-quality content, it takes a lot of time to create it through modeling, rendering, and animation processes that give it a sense of reality. In addition, in a web environment, a transmission delay may occur when a large amount of 3D data is transmitted. Therefore, this study proposed a method to improve loading speed by processing water, land, and altitude data, which are the main components of the 3D map, to improve web 3D GIS rendering performance. In addition, performance was analyzed using a Web Worker after measuring FPS for each number of 3D objects.

색인어 : 웹, 3차원, 지리정보체계, 웹 기반 그래픽 라이브러리, 지오티프

Keyword : Web, 3D, GIS, WebGL, GeoTIFF

<http://dx.doi.org/10.9728/dcs.2022.23.10.2085>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 03 October 2022; **Revised** 19 October 2022

Accepted 19 October 2022

***Corresponding Author; Jae Myeong Choi**

Tel: +82-42-829-7644

E-mail: jmchoi@mokwon.ac.kr

I. 서론

오늘날의 3D 콘텐츠는 다양한 기술의 발전과 융합 등으로 인하여 양질의 콘텐츠를 제작할 수 있고, 그 시간도 많이 단축되었다. 더욱이 코로나로 인하여 비대면 산업들이 활성화가 되고 4차 산업혁명의 다양한 산업들로 인하여 3D 콘텐츠 산업은 더욱 활발해지고 있다.

3D 콘텐츠란 3D로 제작된 모델을 3차원 공간에서 자유롭게 이동하고, 확대, 축소, 회전 변환이 가능한 디지털콘텐츠를 의미한다. 또한, 콘텐츠 표면에 재질과 색을 입혀서 이미지나 동영상 그리고 VR(Virtual Reality)로 제공된다. 3D 콘텐츠는 모델링(Modeling), 렌더링(Rendering) 그리고 애니메이션(Animation) 과정을 거쳐서 생성된다. 그 중 렌더링 기술은 모델링 된 콘텐츠에 사실감을 부여하는 기술로, 렌더링에는 많은 시간이 소요된다[1].

웹 3D GIS는 HTML5/WebGL의 등장으로 별도의 플러그인 설치 없이 모든 브라우저에서 공간정보 서비스를 제공한다. HTML5/WebGL을 활용하여 3D 공간정보에 대한 서비스를 원활하게 받기 위해서는 렌더링 속도가 개선되어야 한다.

3D 데이터 압축, 복원 기술은 데이터 용량은 최소화하면서 데이터 손실이 발생하지 않도록 하는 기술이며, 데이터 전송과 가시화의 경우 효율성을 최대화한다[2]. 3D 데이터의 압축, 복원 기술을 통해 대용량 데이터 관리와 처리의 효율성을 증대시킬 수 있다. 3D 데이터 전송기술은 웹 기반 3D GIS에서 사용자가 필요한 영역의 공간정보 데이터 실시간으로 수신하는 기술이며, 이는 3D GIS 데이터의 용량과 공간적 범위 제한의 한계를 극복할 수 있다. 또한, 3D GIS 데이터를 효율적이고 빠른 속도로 가시화하기 위하여 LOD(Level of Detail) 기술을 적용하는데, LOD 기술은 사용자 시점으로부터 거리에 따라 지형, 영상, 3차원 객체의 정밀도와 해상도를 단계적으로 표현하는 기술이다. 이로 인해 실제 데이터보다 저용량의 데이터를 가시화함으로써 데이터 가시화 속도를 향상시키고, 사용자의 하드웨어 사용을 최소화하도록 지원한다[2][3].

WebGL의 3D 렌더링은 화면의 갱신 속도가 중요한데 스크립트 언어는 바이너리 언어보다 처리 속도가 느리며, 모바일 환경이나 저사양의 시스템에서는 속도 차이가 발생한다. 또한, 웹 환경에서는 많은 양의 3D 데이터를 전송할 때 전송 지연이 발생할 수 있다. 따라서 본 연구에서는 웹 3D GIS 렌더링 성능 향상을 위하여 3D GIS 데이터를 직접적으로 가공하는 것은 한계가 있으므로 3D 지도를 구성하는 주요 요소인 물, 토지, 고도 데이터를 가공함으로써 로딩 속도를 개선하는 방법을 제안한다.

II. 관련 연구

렌더링은 모델링 된 객체에 위치, 광원, 농도 등의 외부 조건을 객체의 재질과 형상에 위치, 광원, 농도 등의 외부 조건

을 넣어 객체를 보다 사실적으로 보일 수 있도록 하는 기술이다[4]. 많은 양의 데이터는 렌더링을 통하여 WebGL에서 표현하려면 속도가 중요하다. 그래서 3D 공간 데이터의 렌더링 속도를 개선하기 위해 다양한 기법들이 존재하며, 이 기법들은 기술의 진화에 따라 발전을 거듭하고 있다.

본 연구에서는 WebGL에서 사용 중인 절두체 컬링, LOD, Web Worker, Emscripten과 3D 모델링을 위한 Blender GIS에 대하여 알아본다.

2-1 절두체 컬링(Frustum Culling)

절두체 컬링(Frustum Culling)은 보이는 부분만을 렌더링하고 보이지 않는 부분은 렌더링을 하지 않는 기법으로 대용량의 데이터 서비스를 위해 필요한 기법이며 3D 공간데이터를 효율적으로 서비스할 수 있어 현재 대부분의 서비스에서 사용하고 있다[5].

절두체는 입체를 절단하는 하나나 두 평행면 사이의 부분을 나타내며, 컴퓨터 그래픽스에서는 뷰잉 프러스텀(Viewing Frustum)이라하여 스크린에서 볼 수 있는 삼차원 영역을 나타낸다. 이것은 잘라낸 각뿔으로 만들어지며, 은면 검출의 방법이다.

가장 기본적인 컬링 기법은 6개의 평면을 이용한 방식이며, 카메라 시야 상, 하, 좌, 우 방향으로 경계선이 되는 평면, 카메라 시야에 수직하며 가장 가까운 곳과 먼 곳의 시야 범위를 나타내는 평면으로 총 6개의 평면을 이용하는 방식이다. 표 1은 절두체를 이루는 6개의 평면을 나타낸다.

절두체 컬링은 6개의 평면에 포함되는 영역만을 렌더링하므로 (1)의 평면 방정식을 사용하여 평면 내부와 외부를 판단한다.

$$ax + by + cz + d = 0 \tag{1}$$

여기서 a, b, c는 평면에 대한 법선벡터이며, d는 평면과 원점 간의 거리를 나타낸다. 또한 x, y, z는 각 축에 대한 평면을 이루는 점들이다.

표 1. 절두체를 이루는 6개의 평면

Table 1. The Six Planes forming the Frustum

Name	Definition(Location)
Near Plane	Plane perpendicular to the camera and representing the nearest viewing range
Far Plane	Plane perpendicular to the camera and representing the farthest viewing range
Left Plane	Plane representing the camera's left field of view
Right Plane	Plane representing the camera's right field of view
Top Plane	Plane representing the camera's top field of view
Bottom Plane	Plane representing the camera's bottom field of view

2-2 LOD(Level Of Detail)

LOD(Level Of Detail)은 멀리 있는 물체는 잘 보이지 않는 특성을 이용하여 객체를 표시하는 기술로 여러 단계의 해상도로 표현하여 렌더링 속도를 개선한다[5]. LOD는 저장하는 방식에 따라 정적 LOD와 동적 LOD로 구분하는데 3D 공간정보 서비스에서는 빠르게 렌더링이 가능한 정적 LOD를 사용한다.

LOD 알고리즘은 거리, 크기, 속도, 편식율, 깊이 등으로 분류할 수 있으며, 현재 거리기반 방식이 가장 많이 사용되고 있다. 거리기반 방식은 시점과 거리가 가까운 지형은 자세히 표현하고, 시점과 거리가 먼 지형은 간단히 표현한다.

2-3 Web Worker

Web Worker(웹 워커)는 자바스크립트의 Background Thread에서 스크립트를 실행하는 방법으로, UI Thread와 별개의 Thread를 실행시켜 사용자 인터페이스(UI)를 방해하지 않고 작업을 수행할 수 있다.

Web Worker는 매우 복잡한 수학적 계산 작업, 원격지에 있는 리소스에 대한 액세스 작업(로컬 스토리지를 액세스하는 경우), 백그라운드에서 오랜시간 작업해야 하는 경우, UI 스레드에 방해 없이 지속적으로 수행해야 하는 작업 등에 활용할 수 있다[6].

Web Worker는 3D 공간데이터(지형, 모델, 텍스처 렌더링 처리 등)를 렌더링 하는 경우 비동기 방식을 사용하여 렌더링 속도를 향상시킬 수 있다.

그림 1은 Web Worker의 동작 방식을 나타낸다.

2-4 Emscripten

Emscripten은 LLVM(Low Level Active Machine) 컴파일러에 대한 백엔드로서 실행되는 소스 대 소스 컴파일러로서, Asm.js라는 이름의 자바스크립트 하위 집합을 생성한다. 웹 어셈블리도 생성이 가능하다. Ams.js는 브라우저에 미리 컴파일되어 가능하여 컴파일된 프로그램들이 자바스크립트로 전통적으로 작성된 것들보다 훨씬 더 빠르게 실행될 수 있다[7].

Emscripten은 언리얼 엔진 3, SQLite, MeshLab, 불릿 물리 엔진, 오토캐드, Qt 애플리케이션 프레임워크 등을 이식하기 위해 사용되고 있다.

Emscripten은 속도 향상을 위하여 C언어로 개발하기 때문에 변환하는 과정이 필요하다. 그러나 성능 향상에 장점이 크기 때문에 대부분의 웹 브라우저들은 Asm.js를 지원한다.

2-5 Blender

Blender는 3D 컴퓨터 그래픽 제작 소프트웨어로서, 오픈 소스 GPL(General Public License) 라이선스이며 무료로 사용할 수 있다.

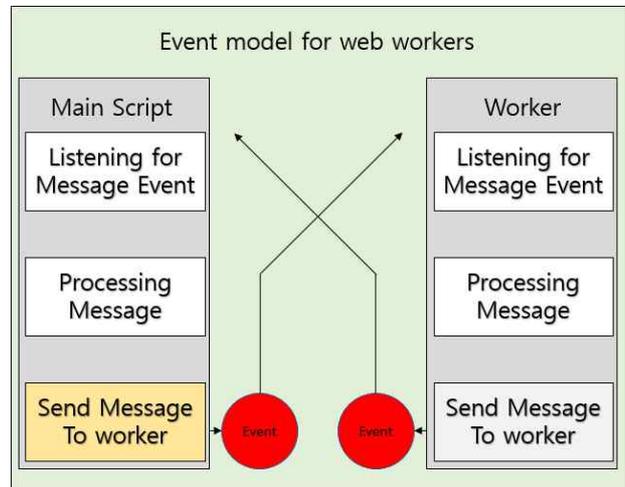


그림 1. Web Worker 동작 방식

Fig. 1. Web Worker Mechanism

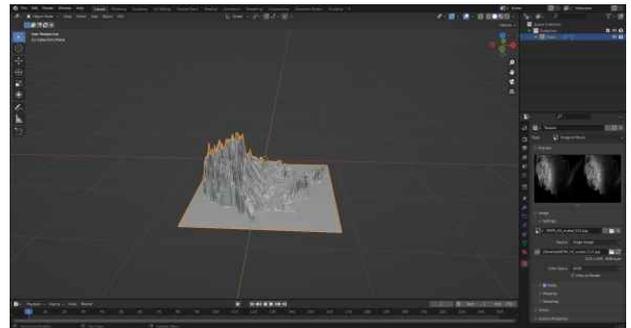


그림 2. Blender OSM

Fig. 2. Blender OSM

Blender는 Blender GIS 또는 Blender OSM(OpenStreetMap)을 통해 3D GIS를 제작할 수 있다. OSM 관련 애드온을 사용하면 지형과 건물을 바로 모델로 불러들일 수 있으며, 국가공간정보포털에서 제공하는 지형도 등고선만큼 정확하지는 않지만, 빠르게 그 지역의 지형과 건물을 세울 때 아주 유용하다. 그림 2는 Blender OSM 작업 화면이다.

III. 제안하는 렌더링 방법

본 연구에서 제안하는 렌더링 속도 개선 방법은 많은 양의 3D GIS 데이터를 직접적으로 가공하는 것은 한계가 있으므로, 3D 지도를 구성하는 주요 요소인 물, 토지, 고도 데이터를 가공함으로써 로딩 속도 등의 성능을 개선하는 방법을 제안한다. 또한, 확장자 마다 성능 차이가 있는데 Blender를 이용하여 x3d, fbx, glTF/Glb, obj 등의 다양한 확장자를 Web Browser 내에서 성능이 우수한 glTF/Glb 파일을 사용해 GeoTIFF로 제작된 지리 데이터와 Blender GIS로 작성된 지리 데이터를 사용하여 FPS를 측정한다.

3-1 GeoTIFF

GeoTIFF는 위성 영상 시스템, 항공 스캔 사진, 스캔 지도, 디지털 표고 모델 또는 지리 분석 결과로 생성된 TIFF 이미지와 관련된 모든 지도 정보를 설명하기 위해 제공되는 TIFF 태그 집합을 의미한다. GeoTIFF는 USGS SDTS 표준이나 FGDC 메타데이터 표준과 같은 기존의 지리적 데이터 교환 표준을 대체할 의도가 없으며, 기존의 대중적인 래스터 데이터 형식을 보강하여 GeoReference 및 GeoCoding 정보를 지원한다.

그림 3은 GeoTIFF 생성 흐름도를 나타낸다.

Step 1 : DEM(수치표고모델)은 OpenTopography 또는 DEM을 제공하는 기관으로부터 다운받을 수 있다. OpenTopography 내에서 제공하는 BBOX(Boundary Box) 좌표값을 입력하거나 그림 A와 같이 Select a region을 사용하여 특정 지역만큼 DEM 이미지를 추출할 수 있다. 추출된 DEM 이미지는 그림 B와 같이 Esa, Nasa, SRTM 등 다양한 기고나으로부터 다운로드를 받을 수 있다.

Step 2 : DEM 이미지를 GeoTIFF 파일로 변환하기 위해서는 QGIS와 같은 GIS 응용 프로그램으로부터 변환이 가능하다. DEM 파일을 QGIS에 Import한 후 저장할 때 Format 형식을 GeoTIFF로 변환한 상태로 저장해 DEM 이미지를 GeoTIFF 형식으로 변환한다.

Step 3 : Web Browser 내에서 GeoTIFF 파일을 사용해 고도를 표현하려면 3차원 공간을 표현하는 JavaScript 미들웨어인 WebGL을 기반으로 한 THREE.JS를 사용해 나타낸다.

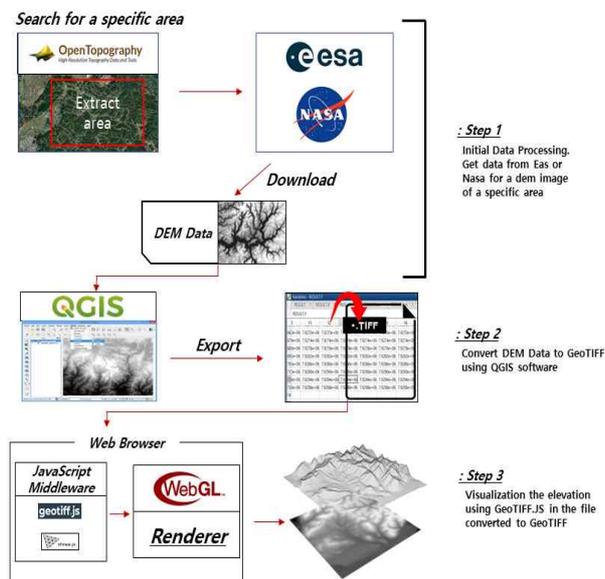


그림 3. GeoTIFF 생성 흐름도
Fig. 3. GeoTIFF Creation Flowchart

THREE.JS로 3차원 공간을 나타내려면 조명, 카메라, 3D 객체, 공간을 렌더링하여 WebBrowser에서 3차원을 표현한다. GeoTIFF 파일을 사용해 지형을 나타낼 경우 THREE.JS에 PlainGeometry를 이용하며 높이 값을 추가하는 형식으로 지형과 고도를 나타낸다.

GeoTIFF.JS의 메서드인 fromUrl() 파라미터로 현재 GeoTIFF 파일이 있는 경로를 적어내고 GeoTIFF 파일을 변수에 담는다. 변수로부터 getImage() 메서드를 사용하여 이미지 데이터를 추출하고 readRasters() 메서드로부터 래스터 데이터를 사용해 PlainGeometry Browser 내에서 3차원 지형과 고도를 표현할 수 있다.

3-2 Blender GIS Modeing

Blender 및 3D Modelling 소프트웨어를 사용해 지형을 제작했을 때 사실적인 효과나 세밀한 부분을 직접 수정할 수 있는 장점이 있지만 Modelling에 사용된 점, 선 면의 개수가 많아질수록 Web Browser 환경에서 좋지 못한 성능이 나타난다. 특정 지역이나 일부만 표현한다면 glTF/Glb로 작성된 파일이 다른 파일들에 비하여 성능이 좋지만 타일 지도 서비스와 같이 다양한 지역을 표현해야 한다면 GeoTIFF가 더 좋은 성능을 낼 수 있다. 그림 4는 Blender로 제작된 GIS Model을 나타낸다.

IV. 3D GIS 모델 성능 분석

4-1 GeoTIFF 성능 분석

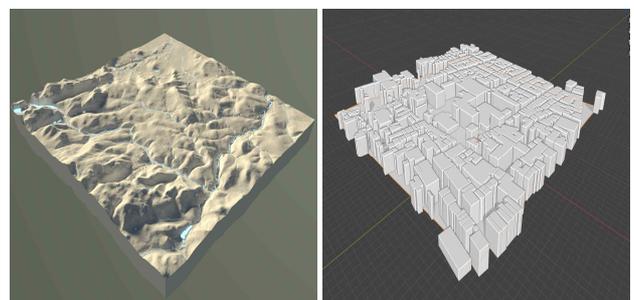


그림 4. Blender GIS 모델
Fig. 4. Blender GIS Model

WebGL 기반의 THREE.JS 라이브러리 내의 GLFLoader 메소드를 사용하여 Web 3D 공간 안의 GIS 데이터를 나타내었으며, Blender 모델의 경우 THREE.JS Clone 메서드를 사용해 복제 시킨 후 3D Object들을 갯수별로 나타내었을 때의 FPS를 측정하였다.

표 2와 3은 GeoTIFF와 glTF/Glb의 속도를 측정한 결과이다.

표 2. GeoTIFF 속도 측정 결과

Table 2. GeoTIFF Velocity Measurement Results

Count	Min FPS	Max FPS	MS	Loading Time (ms)
10	40	63	17	2441
20	45	63	17	2427
30	46	63	17	2486
40	44	62	17	2456
50	44	63	17	2627

표 3. gITF/GIb 속도 측정 결과

Table 3. gITF/GIb Velocity Measurement Results

Count	Min FPS	Max FPS	MS	Loading Time (ms)
10	51	60	17	2451
20	41	50	17	2672
30	27	48	33	2446
40	23	49	35	3021
50	19	47	50	3439

4-1 Web Worker 기반 성능 분석

Web Worker 기반의 성능 분석은 크롬 브라우저로 진행하였으며, Web Worker를 사용하여 지형 데이터를 XML 통신을 통해 지형 데이터를 전송하고 콜백 함수로 고도 값을 받는 Loader 함수를 작성하고, Loader를 통해 들어온 고도 값을 배열로 처리하였다. 부동 소수점 표고 값 0 ~ 2470을 16비트 부호 없는 정수(0 ~ 65535)로 조정해 더 많은 세부 사항을 유지했으며 다음 식을 통해 쉽게 고도 값을 찾을 수 있다.

$$var \text{높이} = \text{Math.} \delta \left(\frac{\text{값}}{65535 \times 2470} \right) \quad (2)$$

또한, Geometry.vertices 안의 Z값에 식을 대입해 TerrainObject를 생성할 수 있다. 그림 5는 GeoTIFF로 제작된 TerrainObject이다.

성능 분석을 위하여 WebWorker를 사용해 XML 통신과 계산하는 부분을 처리하고 기존 Main Thread에서는 장면의 렌더링하는 부분을 나누어서 작업했으며, Web Worker를 사용하지 않는 과정에서는 Main Thread 하나만을 사용해 측정하였다.

표 4는 Web Worker를 사용하였을 때의 측정 결과이고 표 5는 Web Worker를 사용하지 않았을 때의 측정 결과를 나타낸다.

표 4. Javascript Web Worker + GeoTIFF 속도 측정 결과

Table 4. Javascript Web Worker + GeoTIFF Velocity Measurement Results

Count	Min FPS	Max FPS	MS	MB	Loading Time (ms)
10	50	61	17	72	1645
20	48	61	17	71	1894
30	44	61	17	70	2224
40	47	62	17	71	2531
50	47	62	17	70	2528
60	47	60	17	70	2519
70	48	63	17	71	2544
80	46	61	17	71	2224
90	47	62	17	72	2560
100	47	62	17	71	2597

표 5. Javascript Non Web Worker + GeoTIFF 속도 측정 결과

Table 5. Javascript Non Web Worker + GeoTIFF Velocity Measurement Results

Count	Min FPS	Max FPS	MS	MB	Loading Time (ms)
10	40	63	17	68	2441
20	45	63	17	77	2427
30	46	63	17	73	2486
40	44	62	17	77	2456
50	44	63	17	75	2627
60	40	63	17	76	2644
70	45	63	17	78	2690
80	45	62	17	73	2551
90	46	63	17	78	2590
100	45	63	17	76	2481

Web Worker를 사용해 WCS(Web Coverage Service)와 WMS(Web Map Service) 데이터를 받아오는 과정을 새로운 Thread를 사용하였다. 기존 JS에서는 받아 온 데이터를 연산해 높이를 추출하고 장면을 추가하는 방식으로 진행했을 경우 Web Worker를 사용하는 방식이 조금 더 안정적인 상태와 Min FPS가 높은 것을 알 수 있다.

V. 결 론

웹 3D GIS의 렌더링 성능 향상을 위해서는 3D GIS 데이터를 3D 지도를 구성하는 주요 요소인 물, 토지, 고도 데이터

로 가공함으로써 로딩 속도를 개선할 수 있다. 따라서 본 논문에서는 확장자 마다 성능 차이를 보이는 3D GIS 데이터를 웹 브라우저에서 성능이 우수한 glTF/Glb 파일을 사용해 GeoTIFF로 제작된 지리 데이터와 Blender GIS로 작성된 지리 데이터를 사용해 성능을 분석하였다. 그 결과 웹 3D GIS로 특정 또는 일부 지역만 표현한다면 glTF/Glb로 작성된 파일이 성능이 좋으며, 다양한 지역을 표현해야 한다면 GeoTIFF가 성능이 좋은 것을 알 수 있다. 또한 Web Worker를 사용하면 사용하지 않은 경우보다 조금 더 안정적인 상태로 렌더링이 된다는 것을 알 수 있다.

향후 효율적으로 3D 콘텐츠 생성을 위해서는 웹 환경에서 많은 양의 3D 데이터 처리를 위한 알고리즘 연구가 필요할 것이다.

감사의 글

이 논문은 한국전력공사 전력연구원의 "위험성 평가기반 작업자 안전관리 및 감독 협조 기술 개발(R22IA05)" 과제의 지원을 받아 작성된 논문임.

참고문헌

[1] Jeongyeol Baek, Research Trend in 3D Contents Technology, *IITP*, no.1962, Weeklytrend, ITFIND, 2020.09.

[2] Kangwon Lee, Geo-Spatial Information System-GSIS, *Goomibook*, 2016.

[3] The Evolution of 3 Dimensional Technolgy [Internet]. Available: https://blog.naver.com/mltm_nsd/90044029608

[4] Hyungseok Lee, "Analysis of Digital Terrain Model Display by Comparison of GIS Shaded Relief and Rendering", *Journal of the KAGIS*, vol.9, no.1, pp.127-136, 2006.

[5] Seokchan Kang, Rendering speed improvement of 3D geospatial data based on HTML5/WebGL using tile-rendering, Master Degree, University of Seoul, Seoul, Feb. 2017.

[6] [Javascript] serviceWorker, PushManager, Web Worker [Internet]. Available: <https://codens.info/1808>

[7] Emscripten [Internet]. Available: <https://ko.wikipedia.org/wiki/Emscripten>



이동엽(Dongyeop-Yeop Lee)
 2009년 : 순천향대학교 대학원
 (공학박사-전기전자통신시스템)
 2001년~2005년: 청운대학교,
 나사렛대학교,
 순천향대학교,
 한양대학교 시간강사
 2005년~현 재: 한국전력공사
 전력연구원
 재난안전연구팀
 2022년~현 재: 한국안전학회
 전기안전전문회 간사

※ 관심분야 : 스마트안전, 스마트재난관리, Robotics, Arc-Flash, AIoT, Big-data, 지리정보시스템 (GIS), AR/VR, U-헬스케어, 디지털콘텐츠, 재난안전통신, 전력계통 설비진단 등



이주경(Ju-Kyoung Lee)
 2004년 : 충북대학교 (학사)
 2004년~현 재 : 한국전력공사
 전력연구원
 ※ 관심분야 : 인공지능, 디지털트윈, ICT 융복합 기술 등



박미르(Mir Park)
 2004년 : 순천대학교 (학사)
 2016년~현 재: 한국전력공사
 전력연구원
 ※ 관심분야 : 인공지능, AR, VR, 융복합 기술 등



최재명(Jae Myeong Choi)
 2014년 : 목원대학교 대학원 IT공학과 (공학박사)
 2014년~현 재 : 목원대학교
 컴퓨터공학과 교수,
 대학원
 사회안전학과 교수
 2014년~현 재 : 한국디지털콘텐츠학회
 정회원 및 부회장
 2016년~현 재: 행정안전부 재난대응
 안전한국훈련
 중앙평가단 평가위원
 2019년~2021년 : 과학기술정보통신부
 통신재난관리심의위
 원회 위원

※ 관심분야 : 무선통신시스템, 통신재난, 스마트재난시스템, IoT, 재난안전통신, 사회재난(통신)정책, 빅데이터, 재난예측, 재난관리, 디지털콘텐츠 등