Journal of the Korean Association of Geographic Information Studies, 27(1) 2024, pp.1-11 ISSN 1226-9719(Print)
https://doi.org/10.11108/kagis.2024.27.1.001 ISSN 2287-6952(Online)

3차원 표면에서의 시계열 분석결과 표출방법*

이봉준[1※]·박철희[2]

Method of Displaying Time Series Analysis Results on a Three-Dimensional Surface*

Bong-Jun LEE[1※]·Cheol-Hee PARK[2]

요 약

현재 측정되는 많은 데이터가 지표를 기반으로 측정되지만, 측정지점의 높이값이 기초자료로 활용되지 않기 때문에, 3차원 지리정보시스템에 활용하고자 할 때 어려움이 발생한다. 지표면에 많은 양을 표시하기 위해서는 지형정보를 이용하여 지표면에 점을 그리거나, 지표면의 각 측정지점의 높이값을 추출하여 다각형을 생성하는 방법이 다양하게 사용될 수 있다. 본 연구에서는 다양한 형태의 데이터 표현 방법 중 지표면에 표현되는 시계열 측정 데이터의 시각화 성능을 향상시키기 위한 데이터 구축 및 표시 방법을 시도하고, 그 절차와 장단점을 살펴본다.

주요어 : 공간정보 데이터, 3차원 공간정보 데이터, 지표면 데이터, 3차원 시각화

ABSTRACT

A lot of the data currently measured is measured based on indicators, but since the height value of the measurement point is not used as basic data, difficulties arise when trying to use it in a three-dimensional geographic information system. Various methods can be used to create polygons by drawing points on the surface using topographic information or extracting the height value of each measurement point on the surface in order to display large quantities on the surface. Among the various types of data expression methods, this study attempts a data construction and display method to improve visualization performance of time series measurement data expressed on the surface, and examines its procedures and strengths and weaknesses.

KEYWORDS: Spatial Data, 3D Spatial Data, Earth Surface Data, 3D Visualization

---

2023년 10월 13일 접수 Received on October 13, 2023 / 2024년 1월 18일 수정 Revised on January 18, 2024 / 2024년 1월 23일 심사완료 Accepted on January 23, 2024

* 본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호: RS-2020-KA158151)

1 ㈜씨엠월드, 대표 / CmWorld Corporation, CEO
2 ㈜씨엠월드, 부장 / CmWorld Corporation, General Manager
※ Corresponding Author E-mail: bjlee@cmworld.co.kr

서 론

2000년대 초 미국 구글사의 구글어스(Google Earth)가 출시되면서 본격적으로 대중적인 3차원 가시화 시스템이 활용되기 시작한 이래로 현재는 웹 환경에서도 별도의 설치 과정 없이도 활용되고 있을 정도로 일반화되었다. 초기에는 단순한 지형정보를 제공하는 형태로 이루어지던 3차원 지리정보 시스템은 건물을 포함하여 각종 시설의 모델 데이터를 함께 서비스하는 형태를 지나, 현재는 3차원 분석 및 행정 업무에 활용될 정도로 다양한 지리정보를 제공하는 시스템으로 발전하였다.

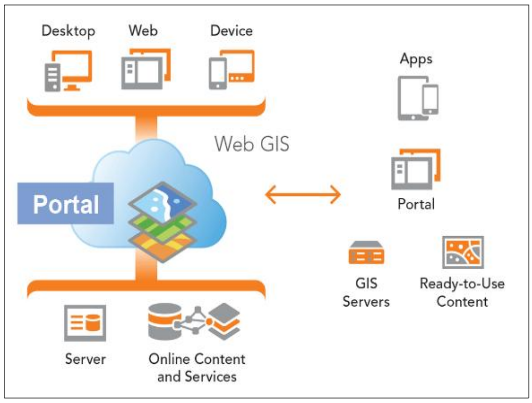

FIGURE 1. ESRI Web GIS concept diagram

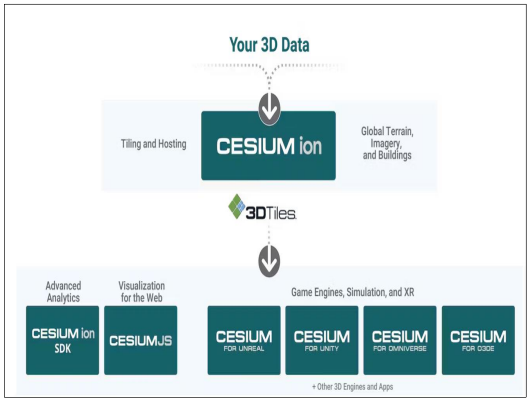

FIGURE 2. AGI cesium system architecture

현재 가장 많이 사용되고 있는 3차원 지리정보 웹 3D 엔진은 ESRI사의 ArcGIS Online 그림 1 혹은 AGI사의 Cesiumjs 그림 2, NASA의 WorldWind와 같은 프로그램들이 활용되고 있다. 웹 3D GIS 엔진들은 3차원 데이터를 그림 3과 같이 가시화하고 사용자 인터페이스를 제공하며, 사용자가 제작한 3차원 데이터를 표출하기 위한 API를 제공하고 있다.

본 연구에서는 오픈소스로 제공되고 있는 Cesiumjs를 대상으로 하여 기능을 테스트하고 비교하였다.

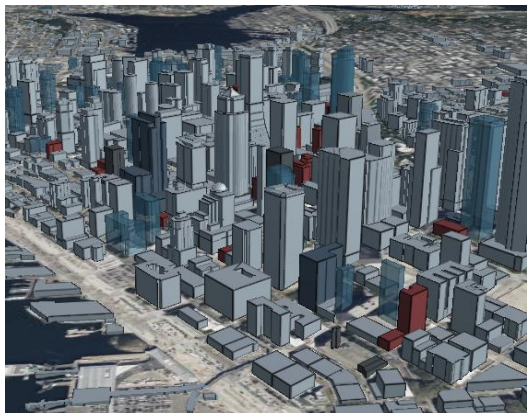

FIGURE 3. Cesiumjs execution screen

이번 연구에서는 3차원 지리정보 시스템에서 성능을 저하시키는 다양한 사례 중에서 다수의 분석 결과를 시계열로 데이터로 가지고 있는 경우에 대하여 그 성능을 향상 시키기 위한 방법을 시도해 보고 그 절차와 성능 향상 요소에 대하여 검토해 본다.

이론적 고찰

3차원 지리정보 데이터를 가시화 하는데 있어 데이터의 용량과 복잡성으로 인하여 현재도 성능 향상을 위해 다양한 방법이 시도되고 있다. 근본적으로는 그림 4의 그래픽 하드웨어 성능 향상을 통하여 자연스럽게 성능이 향상되고 있지만 빠르게 증가하는 3차원 데이터를 처리하기 위한 소프트웨어적인 방법에 대한 연구 또한 진행되고 있다.

FIGURE 4. Annual graphics card performance enhancement graph

하지만 여전히 미리 준비된 지형정보와 지표 영상데이터를 제외하면 기술적 한계로 인하여 분석결과 데이터를 가시화하는데 제약이 따르게 된다.

웹에서의 지형, 지표 이미지 등의 기본 데이터 표출은 TMS(Tile Map Service), 그림 5와 같은 3D Tiles 형태로 어느 정도 표준화 되었지만 대량의 분석 결과를 표출하는데 있어서는 제약이 존재한다. 이런 제약은 한 번에 표시해야 할 데이터가 많은 경우, 혹은 가시화 이전에 원본 데이터를 변형해야 하는 경우에 주로 발생한다, 한 번에 표시해야 할 데이터가 많은 경우는 다량의 측정 데이터를 동시에 표출하거나 혹은 지점 표출 시 지점마다 다른 형식으로 데이터를 표출하는 경우이고, 원본 데이터를 변형해야 하는 경우는 원본 데이터가 측량점(Point)으로 이를 지표에 히트맵(HeatMap)으로 표현하는 경우처럼 원본의 형태와 다른 형태로 결과를 표출해야 하는 경우이다.

또한, 웹을 기반으로 하는 경우에는 좀더 근본적으로 크게 세가지 제약이 따르게 된다. 첫 번째는, 항상 네트워크를 통하여 전달되는 데이터로 인하여 데이터의 크기에 따라 전송속도에 따른 제약이 따르게 되며, 이에 따라 항상 데이터 경량화 문제가 따르게 된다. 이를 해결하기 위한 방안으로 glTF와 같은 보다 향상된 압축 기술을 활용한 3차원 모델을 이용하거나, 사용자 위치와 데이터의 거리에 따라 해상도를 달리하는 LOD를 적용한 데이터를 이용하는 3D Tiles 그림 5와 같은 기술을 이용하게 된다.

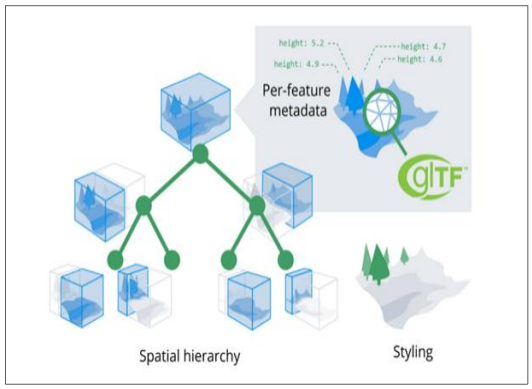

FIGURE 5. 3D tiles structure(LOD application structure)

두 번째는 웹에서 지원하는 3차원 아키텍쳐로 인한 제약이다. 현재 데스크탑의 3차원 시스템의 경우 DirectX 혹은 OpenGL과 같은 기본 라이브러리를 활용하고 있으며, 최신 3차원 기술을 적용하여 활용되고 있다. 그림 6은 WEBGL의 경우 OpenGL의 간략화된 버전인 OpenGL ES를 모델로 하기 때문에 지원되는 API에 제약이 있으며, 데이터를 처리하는 파이프라인에도 제약이 있어 완벽하게 GPU를 활용하기 어렵다. 이러한 제약은 웹 상에서 3차원을 지원하는데 제약을 만들게 되고 결국, 기능의 제약을 만들게 된다.

WebGL	VS	OpenGL
OpenGL ES2	Origin	Complex System
Fragment & vertex shaders	Features	Greater variety of shader types
Operates in browsers	Installation	Requires drives and installation
JavaScript	Programming language	C
Medium	Learning curve	More complex
Web services	Use cases	Video games

FIGURE 6. Comparison of WebGL and OpenGL

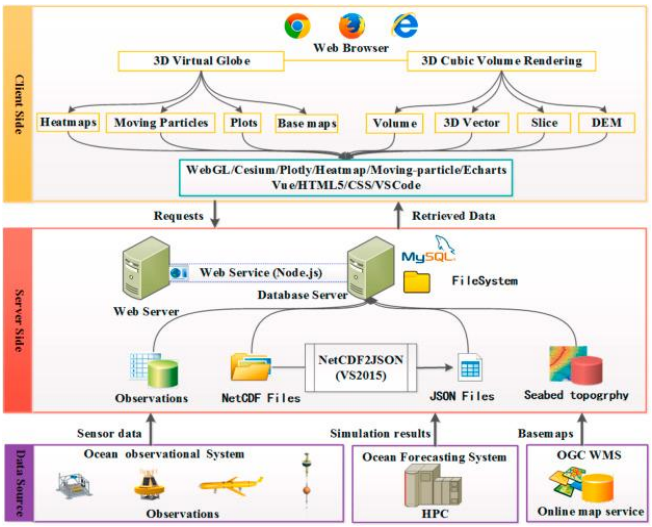

FIGURE 7. Marine data service framework proposed

세 번째는 응용 프로그램 운영 환경이다. 데스크탑 환경은 해당 데스크탑의 모든 기능을 이용할 수 있으며, 가장 빠르게 실행되는 고유의 명령어 집합으로 제공된다. 반면에 웹 환경은 웹 브라우저 내에서 동작하며, 웹 브라우저를 지원하는 가상기계 상에서 동작하는 이중 구조로 되어 있다. 이러한 구성은 프로그램 실행 시간에 해석되는 인터프리터 방식으로 실행되게 되어 성능에 제약을 가지게 되고, 가상기계가 제공하는 메모리의 양과 사용할 수 있는 장치의 기계적 사용 한계에 묶일 수 있다.

OGC에서 소개된 3DTiles 형식(OGC 2019)은 사전 정의된 비정형 격자를 이용하여 3차원 데이터에 가시화 조건을 부여하여 거리에 따른 객체의 선별적 가시화로 성능향상을 지원하고 있으며, Epic Games의 Unreal Engine의 Nanite(Epic Games. 2020)와 같은 기술은 지오메트리를 실행시간에 LOD 처리하여 거리에 따라 단순화된 객체로 시스템 부하를 줄이는 기법을 선보였다. 그림 7의 Qin(2021)의 연구에서는 공개된 오픈소스(Cesium)과 같은 3차원 지리공간 정보 소프트웨어에서의 대량의 해양

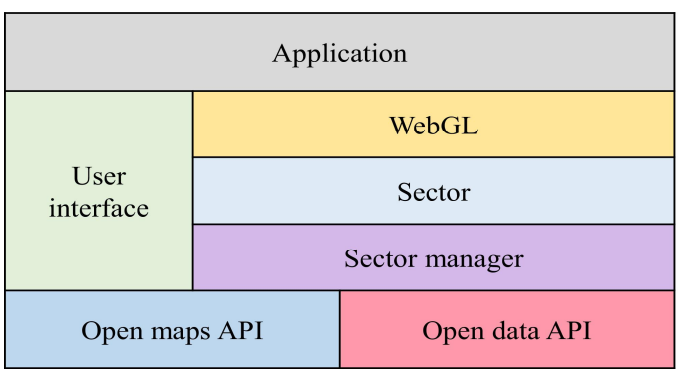

FIGURE 8. Structure of the vworld open platform for spatial information

데이터를 가시화 하기 위한 프레임웍을 제안하기도 했는데, 주로 전체적인 프레임웍에 대한 구성이 주를 이루고 서버 측에서 데이터를 사전에 처리하거나 혹은 실행 시간에 처리하여 웹 환경에 맞춘 Json 형태의 데이터로 변환하여 서비스 하는데 주안점이 있다.

Yang(2021)이나 그림 8의 Lee and Jang (2018)의 연구를 보면 국내 서비스 중인 VWorld 시스템 그림 8과 같이 범용적인 3차원 지리정보 서비스 아키텍처를 소개하며, 기초적인 지형정보와 고도 데이터, 건물 등의 데이터 가시화에 대한 프레임웍 및 방법을 소개하는데, 주로 데이터 관리와 아키텍처 에 대하여 소개하고 있고, 대부분의 경우 일반적인 3차원 운영기술을 지리정보에 적용할 경우 사용되는 기술에 대하여 설명하고 있다.

기초 데이터를 포함하여 3차원 공간 정보 분석 결과를 가시화 하는 방법에 대하여는 Lee(2011)의 연구와 같이 3차원 분석 결과를 모델화 하여 가시화 하거나 정명희의 원격탐사 영상 시각화 Jung(2008)에서 제시한 것과 같이 지표에 이미지를 대치하는 방법을 사용하는 경우도 있다.

Song et al.(2011)은 홍수위험지도를 3차원 지도위에 표출하기 위한 데이터 구성과 제작방법을 설명하고 있는데, 본 연구도 이와 유사한 구성의 아키텍처를 활용한다.

3차원 가시화 성능을 높이기 위한 방법은 위와 같이 하드웨어적인 측면, 기본 3차원 프레임워크 측면, 서비스 아키텍처 측면, 데이터 구성 측면 그리고 본 연구에서 제시하는 바와 같이 클라이언트 데이터 처리 방식 개선 측면에서 시도해 볼 수 있다.

연구방법

1. 데이터 구성 범위

연구를 위하여 사용된 데이터는 약 1 평방킬로미터의 영역에 분포하는 수위 센서를 통하여 측정된 홍수 시 수위 데이터를 사용하였다.

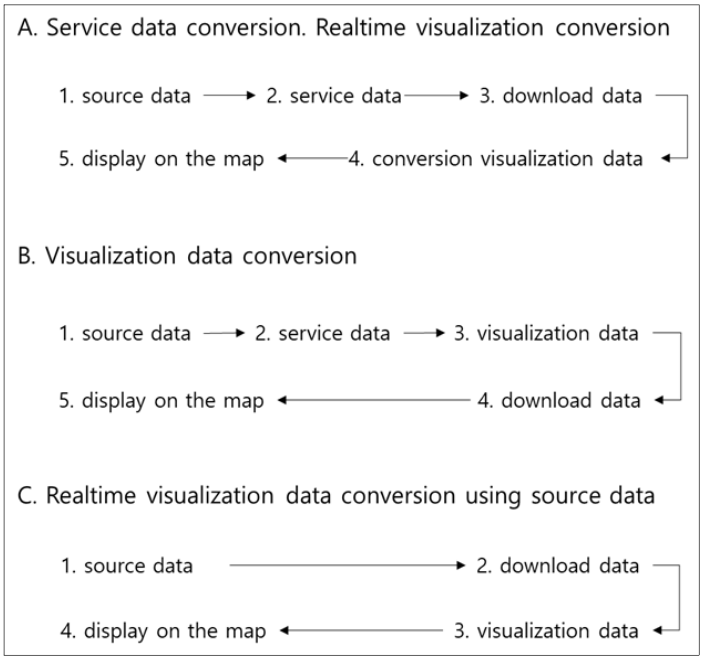

FIGURE 9. Data visualization procedures

천을 따라 분포하는 수위 센서로부터 측정된 데이터로 48시간 동안 10분 간격으로 측정된 데이터로 총 288개의 데이터 셋이다. 시스템에서 사용할 최종 데이터는 측정 시간별로 측정 데이터를 내삽을 통하여 격자 형태로 만들어졌으며 270x380개의 수심 데이터를 포함하고 있다.

2. 데이터 가시화 절차

시계열 데이터 가시화의 기본 절차는 데이터 형태나 목적에 따라 달라진다. 이를 구체적으로 살펴보면 그림 9와 같이 분류 할 수 있다.

A의 경우 원시데이터로부터 실행 시간에 3차원 지리정보 엔진에서 활용하기 좋은 형태로 미리 데이터를 변환해 놓는 경우로, 사전에 미리 작업을 함으로써 실행 시간에 데이터 변환 시간을 다소 줄일 수 있다. 서비스 데이터는 중간 데이터 형태로 플랫폼과 무관하게 데이터를 활용하기 위하여 정의하여 사용한다. 가장 일반적인 형태의 서비스 방식이다.

B의 경우는 가시화 가능한 데이터까지 미리 만들어서 다운로드한 데이터를 바로 가시화 할 수 있도록 하는 경우이다. 사전 처리를 미리 함으로써 데이터 처리시간을 줄일 수 있지만 실제 다운로드한 데이터를 바로 가시화에 사용할 수 있는 경우는 많지 않다. 이와 같은 경우는 가시화 데이터가 이미지와 같이 변환 없이 3차원 플랫폼에서 바로 사용 가능한 경우에만 해당한다.

C의 경우는 모든 데이터처리를 실행 시간에 하는 방식으로 가장 효율이 높지 않은 방식이지만 사전 데이터 처리가 어려운 외부로부터 데이터를 실행 시간에 공급받는 경우나 혹은 로컬 시스템에서 데이터를 공급받아 처리하는 경우에 사용한다. 데이터 변환 성능이 현저하게 느릴 경우 별도의 실시간 변환 프로세스를 운영하여 가시화 처리 기능과 분리하여 운영하는 방법도 고려해 볼 수 있지만 웹의 특성상 로컬마다 개별 데이터 처리 루틴을 설치하는 것은 보안 문제로 인하여 쉽지 않다. 그러므로 제한적인 경우에만 활용이 가능하다.

3. 평가 방법

위의 세 가지 경우가 분석 데이터를 처리하는 방식이다. 데이터 형태가 다르다면 이 들 중 선택하여 처리 방식을 정하게 된다. 가장 보편적인 방식은 A의 경우이다. 본 연구에서 준비한 데이터는 A의 형식을 기본으로 하여 처리하도록 구성하였다.

그림 9에서의 처리 절차에 따라 절차를 단순화하면, 데이터를 사전 처리하는 구간, 데이터를 다운로드하는 구간, 데이터를 가시화 하는 구간으로 나눌 수 있다. 이 중에 데이터를 가시화 하는 구간의 이전 절차는 동일하게 하고 다운로드 이후 가시화 시간을 측정하여 한다.

연구 결과 및 고찰

가시화 구간별로 성능을 높이기 위한 처리 방법을 적용해 보고 성능을 평가해 본다.

1. 가시화 방법

1) 가시화 방법1. 포인트로 표시하는 방법

측정 데이터를 포인트로 표시하는 방법은 준비한 데이터와 같이 내삽을 통하여 도출된 데이터를 사용하지 않고 측정 지점을 바로 사용하면 되지만 성능 개선을 위한 테스트를 진행하기 위해 준비한 최종 데이터를 이용하였다.

데이터 형태는 포인트 목록이므로 전체 좌표에 대하여 일반적인 asc(ASCII Grid File) 형식으로 그림 10과 같은 데이터를 준비하였다.

```
ncols 4
nrows 6
xllcorner 0.0
yllcorner 0.0
cellsize 50.0
NODATA_value -9999
-9999 -9999 5 2
-9999 20 100 36
3 8 35 10
32 42 50 6
88 75 27 9
13 5 1 -9999
```

FIGURE 10. ASC format data (sample)

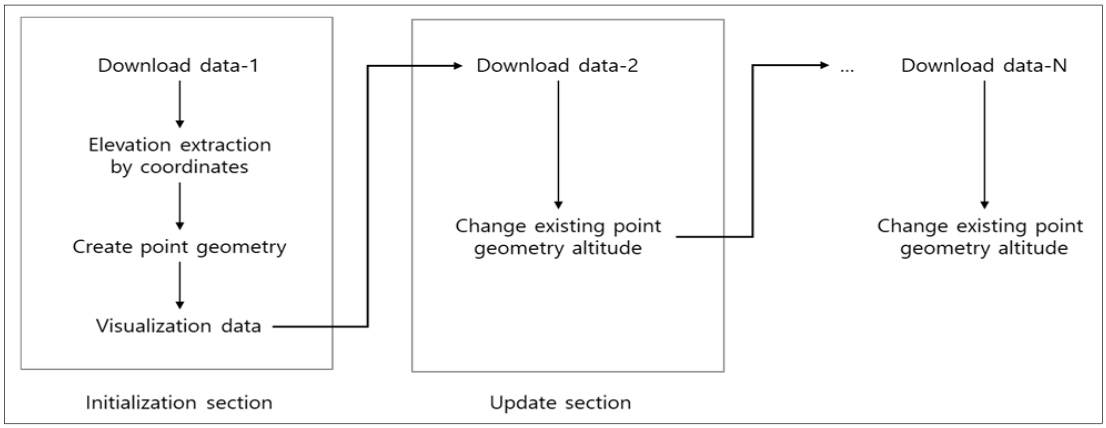

FIGURE 11. Analytical data visualization procedure using point

가시화를 위한 데이터 처리 절차로는 각 지점의 높이 값이 사전에 정의 되어 있지 않으므로 입수한 데이터에 대한 각 지점별 높이 값을 구해야 한다. 모든 데이터가 격자 형태로 되어 있으므로 같은 위치에 데이터가 존재하게 되므로 시작 지점 데이터에 대하여 한 번만 처리하고 이후에는 구해놓은 고도 값을 이용한다. Cesiumjs에서는 각 지점의 높이 값을 구해주는 API가 존재하는데 단일 지점의 경우 globe.getHeight 함수를 사용하는 것이 보편적이지만 다수 위치의 높이 값을 가져오려면 여러 번 반복할 수밖에 없어 함수 실행간 지연으로 인하여 다소 성능에 영향을 끼칠 수 있다. 대신, Cesium.sampleTerrain 함수는 지정한 지점에 대하여 일괄적으로 구하는 것이 효과적이다.

가시화를 위한 데이터 프로세스 초기화 구간은 그림 11과 같이 최초 데이터 로딩 시 한 번만 수행하게 되기 때문에 이후 가시화 구간에서는 단순 고도 갱신만으로 시간의 흐름에 따른 변화를 표출할 수 있게 된다. 데이터 고도 갱신 시 포인트의 색상을 함께 갱신하여 가시성을 높일 수 있다.

2) 가시화 방법2. 폴린곤으로 표시하는 방법

분석 데이터를 폴리곤으로 영역을 만드는 방식은 방법 1과 같은 방식으로 진행하나 Point

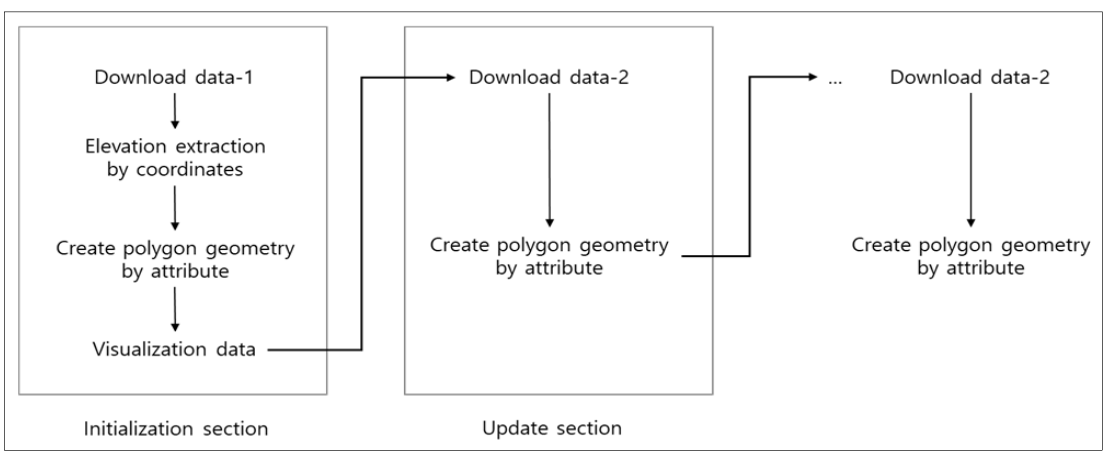

FIGURE 12. Analytical data visualization procedure using polygon

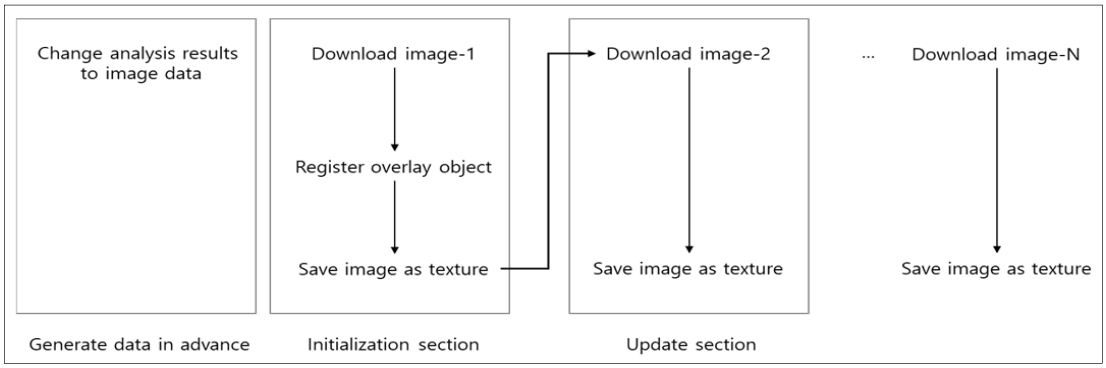

FIGURE 13. Visualization procedure for analysis data using images

Geometry로 가시화를 하는 대신 값에 따른 영역을 폴리곤으로 처리하는 구간으로 대체 된다. 그림 12의 이런 방식은 단일 속성 영역의 범위를 표시하는 데 이용될 수 있지만 다중 속성을 가진 경우 속성별 영역을 만드는 부분에서의 처리가 복잡하여 많은 처리 시간이 필요하게 된다.

3) 가시화 방법3. 이미지로 중첩 표시하는 방법

이미지 중첩으로 시계열 분석 데이터를 처리하는 방식은 사전에 분석 결과를 이미지화하여 실행시간에는 단순 이미지 중첩만으로 데이터를 처리하는 방식이다. 그림 13은 사전 처리가 이루어지므로 보다 데이터 처리 구간을 로컬의 웹 브라우저에 의존하지 않는다. 가시화 방법1과 가시화 방법2에 비하여 고도를 추출하는 부분을 없애고, 브라우저에서 가시화를 위한 처리 프로세스가 사라지므로 전반적인 처리 속도가 향상된다.

분석 데이터를 이미지화하기 위한 내삽 알고리즘은 Kriging 알고리즘을 이용하였으며, 이미지 형식은 PNG를 사용하였다. PNG를 사용한

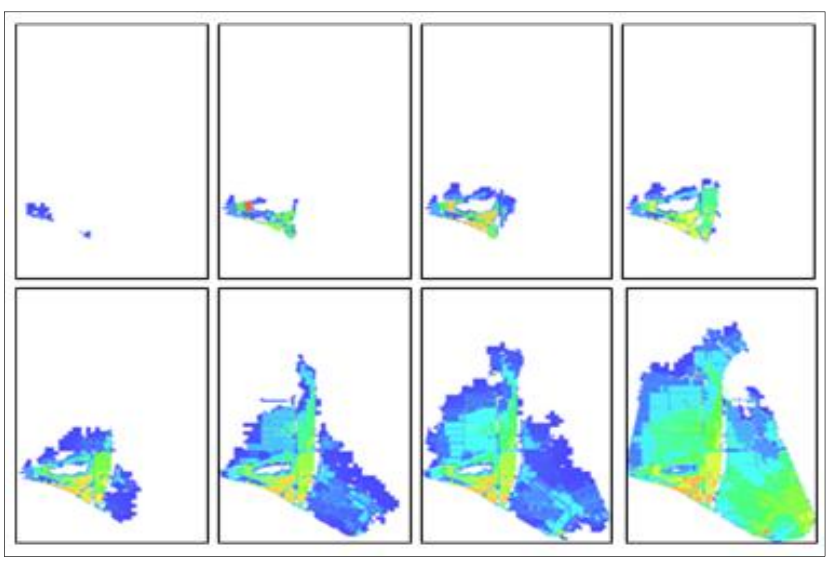

FIGURE 14. PNG image after imaging processing

이유는 Alpha 값을 이용하여 NODATA_Value를 처리하기 위함이며, NODATA_Value에 해당할 경우 Alpha 값을 0으로 처리하여 그림 14와 같은 완전 투명 처리를 하였다.

이미지 처리가 끝난 데이터는 시간의 흐름에 따라 순차적으로 웹 브라우저로 다운로드 되고, 첫 번째 이미지 다운로드 후 생성한 오버레이 객체에 지정되어 화면에 디스플레이된다. 두 번째 이미지 이후로는 첫 번째에서 생성한 오버레이 객체의 텍스쳐를 다운로드한 이미지로 교체함으로써 처리가 끝나게 된다.

위와 같은 처리 결과 평균 가시화 시간은 표 1과 같다.

TABLE 1. Average display time

Display Method	Average display time(ms)
Display Method 1 With Point	48.3
Display Method 2 With Polygon	123.5
Display Method 3 With Image	5.2

위의 결과로 보면 이미지로 처리하는 방식이 세 가지 방법 중 가장 좋은 성능을 보인다.

2. 시스템 처리

이미지를 중첩하는 기본 방식에 따라 시계열 데이터를 순차적으로 오버레이 객체에 적용 시켜서 표출하는 방식은 단순하지만, 시간에 따른 분석 결과의 표출을 효과적으로 시각화 할 수 있는 방법이다. 하지만 웹의 특성상 이미 한번 생성한 객체에 대한 참조 리소스를 지속적으로 변경하는 것은 느린 시스템의 경우 이미지 교체를 하는 순간이 3차원 전체의 Rendering 주기와 정확히 맞추어져 이루어지지 않기 때문에 화면에서 깜박임 현상이 나타날 수 있다. 이런 문제점으로 부드럽지 않은 전환으로 인하여 가시화 품질 저하가 나타날 수 있어 그림 15와 같은 몇 가지 다른 시도를 해보았다.

1) 시스템 처리 방법 1.

가장 일반적인 방식으로 하나의 이미지를 최초 설정할 때 문제가 없으나 내부적으로 이미지를 지정할 때마다 별도 Canvas Element를 생성하고 이를 객체에 적용하는 방식을 사용하여 단순 하지만 반복적으로 객체를 생성하는 부분이 성능에 영향을 미친다.

2) 시스템 처리 방법 2.

시스템 처리 방법 1을 개선하여 하나의 Canvas Element를 미리 설정하여 다운로드된 이미지를 Canvas Element에 Rendering 하는 방식으로 처리하게 되어 새로운 Canvas Element의 생성을 방지하게 되고 경우에 따라 Rendering 구간에서 약간의 추가적인 처리로 부

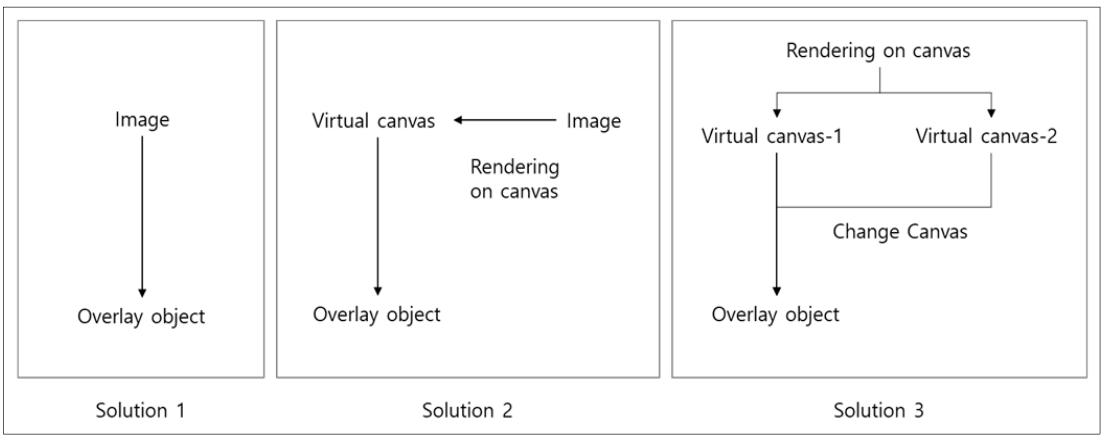

FIGURE 15. Image replacement method for overlay objects

TABLE 2. Average system processing time

Processing Method	Average Processing time(ms)	Blinking
Processing Solution 1	1.2	O
Processing Solution 2	0.71	O
Processing Solution 3	0.73	X

수적인 효과를 더할 수도 있다. 그런데 이 방법은 다운로드가 끝난 이후 Rendering하는 시간과 전체 Rendering Loop와의 차이로 인하여 화면이 깜박이는 Blinking 현상이 나타날 수 있다.

3) 시스템 처리 방법 3.

시스템 처리 방법 3은 Blinking 현상을 방지하기 위하여 가상 Canvas Element를 두 개 이용하여 이를 교체하는 방식으로 추가적인 Canvas Element가 하나 더 필요하기는 하지만 안정적으로 데이터 전환이 이루어진다.

표 2의 결과에서 보는 것처럼 시스템 처리 방법 1이 다소 시간이 더 걸리긴 했지만 어떤 방법이든 전체 시스템의 성능에 영향을 미칠 정도로 유의미한 시간 차이를 보이지 않는다. 오히려 화면에 깜빡이는 현상이 나타 나는지 여부가 성능 보다 더 중요한 요소가 된다. 즉, 시간 차이가 크지 않다면 Blinking 현상이 나타나지 않는 시스템 처리 방법 3을 선택하는 것이 합당해 보인다.

결 론

상기에서 검토한 바와 같이 단순히 데이터를 가시화 하는 경우에도 목적과 내용에 따라 여러 가지 가시화 방법이 있고 성능을 높이기 위한 다양한 방법을 도출 할 수 있다. 웹상에서 여러 형태의 3차원 데이터를 그림 16과 같이 처리하고 가시화하는 것은 여전히 많은 도전이 필요한 분야이다. 특히, 지리정보시스템의 경우 기본 데이터의 용량이 크고, 데이터가 복잡하여 자칫 일반적인 장치에서 운영하기 힘든 경우가 발생하기도 한다.

살펴본 바와 같이 측정 데이터를 가시화 하기

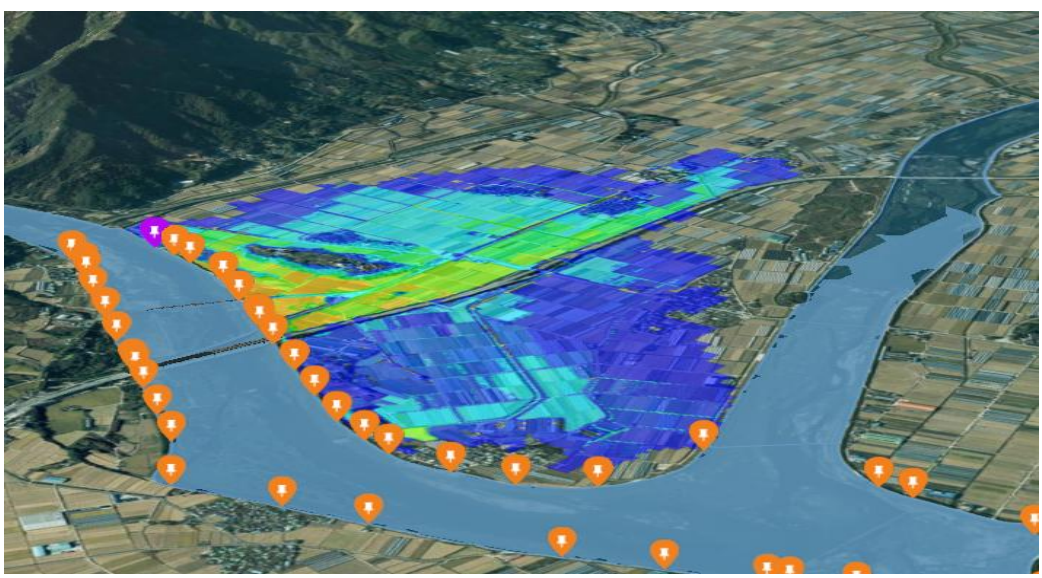

FIGURE 16. Visualizing the results of the analysis on the indicators

위해 데이터의 사전처리를 통한 형식 변경, 가시화 형태의 설정 및 적용, 시스템 지원 아키텍쳐 고려 등 여러 사항을 검토하여야 한다. 각 단계별로 적용 가능한 성능 향상 방안이 존재하며, 본 연구에서와 같이 단순하지만 수행할 수 있는 데이터 처리 절차와 기법을 적용하여 대량의 공간정보를 가시화 하는 방법에 대하여 결과를 도출할 수 있다. 여러 방법을 데이터별, 각 단계별로 적용하고, 기술적 누적을 통하면 전반적인 3차원 지리정보 프로그램의 성능에 확실한 영향을 미칠 수 있게 된다. 여기서 검토한 방식이 유일한 최상의 방법은 아니라도 3차원 지리정보 프로그램상에서 분석 데이터를 처리하는 경우 참고할 만한 기법으로 적용되기를 기대한다. KAGIS

REFERENCES

Lee, A. and I. Jang. 2018. Implementation of an open platform for 3D spatial information based on WebGL. ETRI Journal-Wiley 284-285.

Mujtaba, H. 2021. AMD & NVIDIA GPU Silicon Performance, Efficiency & Cost Progress From The Last 10 Years Visualized WCCFTECH Journal.

Yang, J., Wu, F., Lai, E., Liu, M. and B. Liu. 2021. Hebei Tongtai Surveying and Mapping Service Co.,Ltd, Analysis of Visualization Technology of 3D Spatial Geographic Information System. Hindawi Mobile Information Systems Volume 2021.

Lee, J.Y., Jang, Y.G. and I.J. Kang. 2011. Geotechnical Information Application System Using 3D Geo-Spatial Information Technique. Journal of the Korean Society of Civil Engineers 336-337 (이지영, 장용구, 강인준, 2011. 3차원 공간정보 기술을 활용한 지반정보 활용시스템 개발. 대한토목학회 논문집 336-337).

Jung, M. and E. Yun. 2008. Utilization of Dtabase in 3D Visulization of Remotely Sensed Data. Journal of the Institute of Electronics Engineers of Korea 181-183 (정명희, 윤의중. 2008. 원격탐사 영상의 3D 시각화와 데이터베이스의 활용. 전자공학학술논문 181-183).

NVidia. 2020 Nanite Virtualized Geometry. Unreal Engine 5.0 Documentation.

Qin, R., Feng, B., Xu, Z., Zhou, Y., Liu, L. and Y. Li. 2021. Web-based 3D visualization framework for time-varying and large-volume oceanic forecasting data using open-source technologies. Environmental.

Song, Y.S., Lee, P.S., Yeu, Y. and G.H. Hong. 2012. Flood Risk Mapping using 3D Virtual Reality Based on Geo-Spatial Information. Journal of the Korean Society for Geospatial Information System pp.99-102 (송영선, 이필석, 유연, 김기홍. 2012. 공간정보기반 3차원 가상현실을 이용한 홍수위험지도 제작. 한국지형공간정보학회지 99-102쪽).

3D Tiles Specification. 2019. OGC.

https://modis.ornl.gov/documentation/ascii_grid_format.html. (Accessed October 13, 2023) ASCII Grid Format Description.

https://registry.khronos.org/glTF/. (Accessed October 13, 2023) glTF Specification. KHRONOS Group. KAGIS