

# 공간정보 오픈플랫폼 3차원 건물데이터 포맷 개선방안 연구

## A Study on the Improvement of 3D Building Data Format for Spatial Information Open Platform

김현덕\* · 강지훈\*\* · 김학준\*\*\*

Kim, Hyeon Deok · Kang, Ji Hun · Kim, Hak Joon

### 要 旨

공간정보 오픈플랫폼에서는 국가공간정보를 공개하여 국민이 자유롭게 활용할 수 있는 서비스를 제공하고 있다. 최근 고품질의 3차원 공간정보와 실내공간정보에 대한 수요가 증가하고 있으나, 현재 오픈플랫폼은 실내와 실외의 공간정보 데이터가 상이한 형식과 저장구조로 구성되어 매끄럽게 연결된 서비스가 불가능한 상황이다. 또한 현재 서비스에 사용되는 3차원 데이터 포맷이 최근 서비스 환경의 변화와 신기술을 반영하지 못하고 있는 실정이다. 따라서 본 연구에서는 오픈플랫폼 3차원 데이터의 서비스 향상 및 상호운영성 확보를 위하여 서비스에 사용되는 3차원 데이터 포맷의 개선방안을 제시하였다. 제안한 포맷은 기존 포맷과 비교하여 경량화되었고 렌더링 속도가 개선되었다.

핵심용어 : 공간정보 오픈플랫폼, 3차원 데이터, 건물데이터, 포맷

### Abstract

On the spatial information open platform, the national spatial data are released to provide services that the people can use freely. Recently, the demand for high quality 3D geospatial information and indoor spatial information is increasing. However, open platform is not able to provide seamless service because spatial data of indoor and outdoor are composed of different formats and storage structures. In addition, the 3D data format used in the current service does not reflect the recent changes in service environment and new technology. Therefore, in this study, we proposed new format of 3D data used in service to improve interoperability and service of open platform 3D data. The proposed format is lighter than the existing format and the rendering speed is improved.

Keywords : Spatial Information Open Platform, 3D Data, Building Data, Format

## 1. 서 론

공간정보 오픈플랫폼 서비스(이하 오픈플랫폼)는 국가가 보유하고 있는 국가공간정보를 다양한 서비스 체계로 공개하고 국민들이 자유롭게 활용할 수 있는 기반을 제공하는 서비스로서 국토교통부의 주도하에 출시되었다(Lee, et al., 2014; Spatial information open platform portal service, 2017a).

오픈플랫폼에서는 25~50cm 해상도의 항공사진을 비롯하여 전세계(육지) 15m 해상도의 위성영상, 전국 50m 해상도의 지형자료, 3D 건물 데이터 등의 기반공

간정보와 용도지역도, 연속지적도, 토지이용현황 등의 행정 공간정보 등 다양한 국가공간정보를 제공하고 있다(Jang, et al., 2016; SpaceN, 2017a; SpaceN, 2017b).

최근 실내공간정보가 대두되면서 오픈플랫폼에서도 이에 대한 수요가 증가하고 있는데, 향후 제공받고 싶은 서비스로 실내공간정보에 대한 의견이 제기되었다(Kang and Hwang, 2014). 그러나 현재 오픈플랫폼에서는 실내와 실외의 공간정보 데이터가 서로 상이한 형식과 저장구조로 구성되어 있다. 따라서 실외에서 실내까지 매끄럽게 연결된 서비스가 불가능한 상황이다(Kim and Jang, 2015).

Received: 2017.03.13, revised: 2017.03.02, accepted: 2017.03.21

\* 정희원 · 한국국토정보공사 공간정보연구원 연구원(Member, Researcher, Spatial Information Research Institute of Korea Land and Geospatial Informatix Corp., hdkim@lx.or.kr)

\*\* 교신저자 · 정희원 · 한국국토정보공사 공간정보연구원 선임연구원(Corresponding Author, Member, Senior Researcher, Spatial Information Research Institute of Korea Land and Geospatial Informatix Corp., kangdaejang@lx.or.kr)

\*\*\* 가이아쓰리디(주) 수석개발자(Chief Developer, Gaia3D, Inc., hjkim@gaia3d.com)

또한 현재 서비스에 사용되고 있는 3차원 데이터 포맷은 2007년 오픈플랫폼 초기 구축 당시에 정의된 포맷으로 최근 서비스 환경의 변화와 3차원 데이터 관련 신기술을 반영하지 못하고 있는 실정이다. 뿐만 아니라 3차원 데이터에 대한 수요가 증가함에 따라 3차원 서비스 도입시 메모리 부족 현상으로 인한 하드웨어 증설이 요구되고 있다(Jang, et al., 2015). 하지만 이러한 해결방안은 관리를 위한 인적, 물적 비용 부담이 수반되므로 원천(source) 데이터의 포맷 개선을 통해 메모리 효율을 근본적으로 높일 수 있는 방안이 요구된다.

따라서 본 연구에서는 오픈플랫폼 3차원 데이터의 서비스를 향상시키고 상호운영성을 확보하기 위한 방안으로서 서비스에 사용되는 3차원 건물데이터 포맷을 개선하고자 한다. 이를 위해 우선 대표적인 3차원 데이터 포맷들과 오픈플랫폼의 3차원 건물데이터 포맷에 대한 분석을 수행하고 개선 방안을 도출한다. 이후 분석 결과를 토대로 오픈플랫폼에서 사용할 새로운 3차원 건물데이터 포맷을 제안하고자 한다. 구체적으로 현재 연구되고 있는 HTML5 및 WebGL 기반의 차세대 공간정보 오픈플랫폼 환경을 고려하여 포맷을 설계한다. 또한 실내공간정보로의 확장성과 적용성을 고려하여 건물 내부의 객체를 담을 수 있도록 개발하되 최대한 경량화하여 설계하고자 한다.

## 2. 공간정보 오픈플랫폼 3차원 건물데이터 포맷 분석

공간정보 오픈플랫폼에서 3차원 건물데이터는 인덱스(index) 파일, 객체(object) 파일, 객체의 외곽 이미지인 텍스처(texture) 파일로 구성되어 있다. 인덱스 파일은 3차원 데이터 서비스 방식인 타일링(tiling)에 관한 규약과 타일 상에 3차원 건물의 위치정보 데이터 등 다수의 객체 파일들과 연동하기 위한 정보들로 구성되어 있고, 바이너리(binary) 형태로 기술된 DAT 포맷으로 제공된다. 객체 파일은 하나의 3차원 건물 객체에 대한 형상정보를 저장하는 데이터로 지오메트리 정보와 텍스처에 대한 정보들로 구성된다. 객체 파일은 XDworld Object(XDO) 포맷으로 제공되는데, XDO는 3차원 모델링 분야에서 범용적으로 사용되는 AutoDesk사의 3DS 포맷을 참고하여 만든 포맷으로 바이너리 형태로 기술되어 있다. 텍스처 파일은 범용적으로 사용되는 JPG 포맷으로 제공된다. 인덱스 파일과 객체 파일의 상세 내용은 Table 1 and 2와 같다(SpaceN, 2016).

포맷 분석결과 XDO 포맷은 3차원 객체의 외곽 정보만을 정의하고, 확장성을 고려하지 않아 향후 실내공간 정보나 텍스처 서비스 방식의 변화 등에 대응하기 어려운 구조를 지니고 있었다. 그리고 데이터 구조상 중복점을 발생시킬 수 있어 구조 개선을 통해 데이터 사이즈를 축소할 여지가 있는 것으로 나타났다. 또한 데이터가 바이너리 형태로 작성되어 있으므로 정확한 명세서와 다양한 레퍼런스가 공개가 요구된다.

3. 범용 3차원 데이터 포맷 분석

### 3.1 CityGML

CityGML은 독일의 special interest group 3D

Table 1. Components of index file

Name	Size (byte)	Type	Detail
Level	4	u32	Level (0 base)
IDX	4	u32	Latitude ID
IDY	4	u32	Longitude ID
Object Count	4	u32	Count of object
Version	4	u32	Registration version of object
Type	1	u8	Type of object
KeyLen	1	u8	Key value length of object
Key	KeyLen	char*	Key value of object
CenterPos	16	vector2dd	Center coordinate of object
altitude	4	f32	Altitude of object
box	48	aabox3dd	Boundary of object
ImgLevel	1	u8	Image level
dataFileLen	1	u8	File name length of object
dataFile	dataFileLen	char*	File name of object
imgFileNameLen	1	u8	File name length of image
imgFileName	imgFileNameLen	char*	File name of image

Table 2. Components of object file

Name	Size (byte)	Type	Detail
Type	1	u8	Type of Object
Object ID	4	u32	Identify number of object
KeyLen	1	u8	Key value length of object
Key	KeyLen	char*	Key value of object
ObjBox	48	aabbbox3dd	Boundary of object
Altitude	4	f32	Height of object
FaceNum	1	u8	Count of face
Vertex Count	4	u32	Count of vertex
Vertex	Vertex Count	S3DVertex*	Vertex value
Indexed Count	4	u32	Count of index
Indexed	Indexed Count	u16*	Index value
Color	4	u32	Object color
ImageLevel	1	byte	Depth of Image LOD
ImageNameLen	1	byte	Texture name length
imageName	ImageNameLen	char*	Texture name

(SIG3D)에서 개발한 포맷으로 표준화 단체인 Open Geospatial Consortium(OGC)에 의해 3차원 GIS 데이터의 표준으로 채택되었다(Li, et al., 2015). CityGML은 가상 3차원 도시 모델의 저장과 교환을 위해 개발된 XML 기반의 개방형 데이터 모델로, 공간데이터의 원활한 교환을 위해 국제 표준으로 채택된 Geography Markup Language 3(GML3)의 용용스키마로 표현된다. CityGML의 관계도는 Fig. 1과 같다.

가상 3차원 도시 모델에 대한 3차원 기하(geometry), 위상(topology), 의미(semantics), 표현(appearance)의 4가지 측면에 대한 정보를 포함하고 다른 응용 분야들 간에 공유할 수 있는 공통의 기본 항목(entity), 속성(attribute), 관계(relation)들을 정의하는 것이 목적이다. 객체단위 세밀도(Level of Details, LOD) 모델을 지원하여 LOD0 ~ LOD4의 5단계 세밀도를 기반으로 건물

모델들의 공간적 특성과 의미를 표현한다. 3차원 객체는 다른 데이터베이스 또는 데이터 집합의 객체로부터 제공되거나 관계를 맺을 수 있고 사용자의 기준에 따라서 임의의 도시 객체의 집합을 허용하여 해당 집합을 도시 모델의 부분으로서 표현하거나 이동시킬 수 있다. 나무와 식물, 신호등, 교통표지판과 같은 동일한 모양의 객체들은 다른 위치에 중복으로 표현할 수 있는 데이터 구조를 가진다.

### 3.2 KML

KML은 마크업 언어(markup language)로서 XML 표준을 기반으로 한 구글어스(Google Earth), 구글맵(Google Map)과 같은 지도 플랫폼에서 공간자료를 표현하는데 사용되는 파일 형식이다(Kang, et al., 2014). KML 자체에는 3차원 지오메트리(geometry)를 담당하

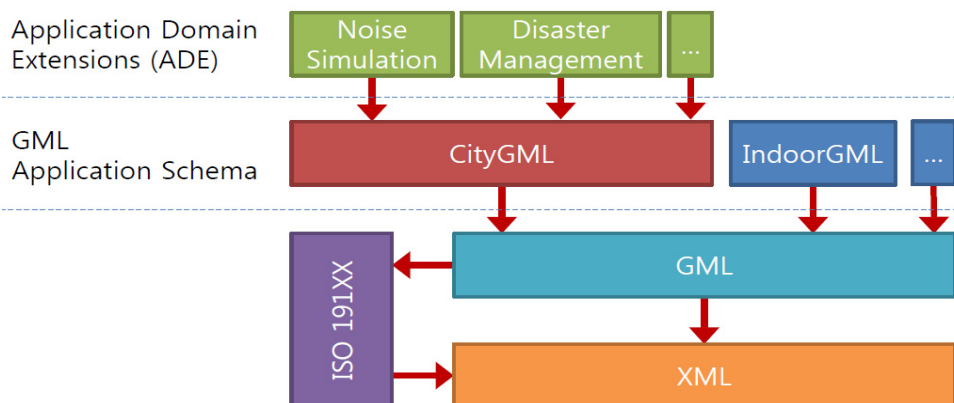


Figure 1. Concept of CityGML

는 태그(tag)가 존재하지 않고, 지오메트리에서 파생된 클래스 들 중 ‘Model’에서 외부 3차원 공간자료를 리소스로 연결해주는 역할을 한다. 즉, KML에서는 직접적인 3차원 지오메트리 기술로 3차원 공간자료를 다루는 것이 아니라, 외부 3차원 공간자료 정보를 연결하고, 연결된 3차원 공간자료를 구글 좌표계로 매핑하여 가시화 하는 방식에 대한 정보만을 가지고 있다.

KML이 수용 가능한 3차원 공간자료는 collaborative design activity(COLLADA)포맷이다. COLLADA는 서로 호환되지 않는 3차원 어플리케이션들의 자체 포맷들 간 호환을 목적으로 고안된 교환 포맷으로 크로노스 그룹(Khronos Group)에서 관리되고 있으며 ISO/PAS 17506 사양으로 ISO에 의해 공식적으로 채택되었다. COLLADA 파일의 확장자는 digital asset exchange(DAE)이고 XML 기반으로 작성된다. 따라서 KML에서 3차원 공간자료를 취급하기 위해서는 COLLADA 자료와 링크하여 구글 어스와 같이 해당 파일 포맷을 지원하는 소프트웨어에서 가시화하게 된다(Park and Lee, 2009).

### 3.3 GL transmission format(gITF)

gITF는 그래픽스 분야의 대표적인 오픈소스 그룹인 크로노스 그룹에서 발표한 포맷으로 OpenGL, OpenGL ES, WebGL 등의 그래픽 라이브러리들에서 쉽게 import/export가 가능하도록 표준화된 툴과 함수를 제공하는 것을 목표로 한 표준 명세이다. javascript object annotation(JSON) 형식으로 파일 사이즈가 컴팩트하고 빠르게 파싱이 가능하며, 용량이 큰 자료는 바이너리 형식으로 링크하여 사용하고 있다. OpenGL 그래픽 라이브러리의 API 자료와 최대한 근접하게 설계하여 로딩시간이 빠르고 독립적인 런타임 환경을 가져 특정 3D 엔진이나 응용프로그램에 종속적인 런타임은 실행하지는 않는다. 기본 사양을 충분히 제공하지만 향후 발전이나 개선을 위한 범용적인 정보와 벤더사 고유의 정보를 추가할 수 있는 확장성을 가지고 있다.

Table 3. Components of gITF format

Type	Extention	Contents
gITF	gltf	node hierarchy, materials, cameras, animation etc.
binary	bin	geometry, animation, Etc buffer information.
image	jpg, png, etc.	texture image
shader	glsl	GLSL(OpenGL Shader Language) shader source code

gITF는 Table 3과 같이 4개의 파일로 구성되어 3차원 데이터를 표현한다(GitHub, 2016).

### 3.4 시사점

CityGML은 3차원 포맷으로서 기본적인 geometry, topology, semantics, appearance 등의 3차원 객체 묘사를 위한 요소들을 가지고 있다. 그리고 3차원 객체 렌더링에 가장 중요한 geometry는 국제표준인 GML3를 사용하고 있다. 또한 교환 포맷으로서 서비스나 렌더링과 같은 성능보다는 객체정보의 기술(description)을 위한 목적이 더 큰 포맷이라 할 수 있다. 다양한 속성(attribute)과 관계(relation)를 기술할 수 있도록 구성되어 있는데, 이러한 특징은 3차원 객체들에 속성과 계층관계를 부여할 수 있게 해준다. 따라서 이 부분은 향후 브이월드 3차원 데이터의 서비스를 확장할 때 고려할 수 있다. 또한 LOD가 정의되어 있는데, 3차원 객체의 크기나 버텍스(vertex) 수로 LOD를 구축하는 것이 아니라 어떤 레벨에서는 박스로 표현한다거나 어떤 레벨에서는 지붕과 벽을 표현하는 방식으로 LOD가 정의되어 있다. 즉, 데이터 스케일에 따라 레벨을 정의하지 않고 주제 별로 레벨을 정의하였는데, 성능이 중요한 렌더링에서는 이러한 특징이 크게 도움이 되지 않는다. 따라서 CityGML은 교환 목적의 포맷으로 성능이 좋지 않은 XML 기반이고, 논리적으로도 속성과 관계 부분을 제외하면 포맷의 개선방향 설정시 참고할 부분은 많지 않다.

KML은 엄밀히 말해서 3차원 포맷이 아니라 3차원 리소스를 구글어스에서 취급할 수 있게 매핑(mapping) 기능을 제공하는 것이라 할 수 있다. 따라서 3차원 포맷으로서 KML을 분석하고자 한다면 KML이 연결할 수 있는 유일한 3차원 포맷인 COLLADA를 분석해야 할 것이다. COLLADA는 과거 서로 호환되지 않는 3차원 어플리케이션(application)들의 자체 포맷들 간 호환을 목적으로 한 중간 포맷(intermediate format)으로 만들어졌다. COLLADA는 WebGL/gITF를 관리하는 크로노스 그룹에서 관리하는 포맷으로 크로노스 그룹에서 gITF를 설계할 때 COLLADA에서 완벽히 변환이 되도록 설계하였다. 따라서 모델링 포맷에 가까운 COLLADA 보다는 개선 방향에 맞게 webGL에 최적화 된 포맷인 gITF를 분석하는 것이 더 타당하다.

gITF 포맷 분석 결과, 포맷 개선방안에 반영할 수 있는 부분은 다음과 같다. 첫째, gITF에서 3차원 객체 묘사에 필요한 리소스 들(geometry, texture, normal vectors, shader 등)은 바이너리로 기술되어 있고, 이들을 연결하여 계층구조로 엮는 매개체는 JSON을 사용

하였다. 이러한 구조는 자바스크립트(javascript)로 구현될 웹 클라이언트에서 접근하여 다루기 쉽고 확장성이 좋다. 따라서 신규로 개발될 포맷을 바이너리로 개발하고 최종적으로 서비스 할 때에는 JSON으로 래핑(wrapping)해서 넘겨주는 방안을 고려할 수 있다.

둘째, glTF는 WebGL에 넘겨주게 될 리소스나 정보들을 WebGL의 API가 받아들일 수 있는 형식 그대로 기록하여 네트워크를 통해 도착한 데이터를 재가공할 필요 없이 바로 WebGL의 API에 넘겨줄 수 있도록 구성되어 있다. 따라서 신규 포맷 개발시 3차원 리소스 데이터를 WebGL이 바로 읽어들이 수 있도록 포맷에 직접 기록하는 방법으로 설계할 수 있다.

셋째, glTF는 리소스를 glTF에 기록할 때 직접 base64-encoding으로 기록하지만 데이터 사이즈가 매우 클 경우 URI로 대체할 수 있다. 즉, glTF와 리소스 데이터를 분리하여 glTF에 해당 리소스의 위치 정보만 기록하는 것으로 이 방법 역시 신규 포맷 개발시 고려할 수 있다.

## 4. 개선방안

### 4.1 개선방향

3차원 포맷을 개선하기 위하여 먼저 서비스 사양 개선 방향을 제시하고, 리소스 데이터 구조 개선방안을 제안한다. 이후 개선된 포맷의 구성방안을 제안하고자

한다.

서비스 사양 개선 방향은 첫째, 3차원 리소스 데이터는 WebGL이 바로 읽어 들일 수 있도록 포맷에 직접 기록한다. 서버에서 도착한 데이터를 추가적인 작업 없이 WebGL이 인자(argument)로 사용하게 할 수 있다.

둘째, 3차원 리소스 데이터를 JSON 형식으로 래핑하여 서비스한다. 3차원 지도서비스의 클라이언트로 사용되는 웹 브라우저는 자바스크립트를 이용해 기능을 구현하고 있다. JSON은 자바스크립트에서 취급할 수 있는 가장 효율적인 컨테이너이고 유연성이 좋아 컨테이너에 새로운 콘텐츠를 확장하기 용이하므로 JSON 형식으로 래핑하고자 한다.

셋째, JSON 안에 리소스 데이터를 삽입할 때는 base64-encoding 기법을 사용한다. JSON은 텍스트 기반의 언어이고 리소스 데이터는 바이너리 기반이다. 따라서 바이너리 데이터를 텍스트 구문 안에 삽입할 때 일반적으로 사용하는 base64-encoding을 사용하고자 한다. JSON 구문의 "uri"라는 키(key)에 base64-encoding된 바이너리를 값(value)으로 할당하면 Fig. 2와 같다.

기존 XDO 포맷 분석 결과 리소스 데이터 구조에서 사이즈를 축소할 수 있는 여지가 있었으므로 이에 초점을 맞추어 리소스 데이터의 구조를 개선하기 위해 세 가지 방법을 사용하고자 한다.

첫째, index array를 제거한다. WebGL이 렌더링 하

```

"buffers": {
  "Duck": {
    "byteLength": 102040,
    "type": "arraybuffer",
    "uri": "data:application/octet-stream;base64,AAABAAIAAgABAAMAAQAAAAQABAAAAAUABgAHAUABQAHAAQABwAGAAgAC
  },
},
"cameras": {
  "cameraShape1": {
    "name": "cameraShape1",
    "perspective": {
      "aspectRatio": 1.5,
      "yfov": 0.660593,
      "zfar": 100,
      "znear": 0.01
    },
    "type": "perspective"
  }
},
},
"images": {
  "file2": {
    "name": "file2",
    "uri": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAgAAAAIACAIAAAB7GkOtAAAAAXNSR0IArs4c6QAAAA1wSF1
  },
},

```

Figure 2. Example of base64-encoding(GitHub, 2016)

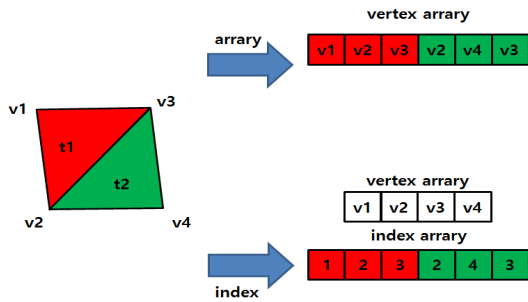


Figure 3. A comparison between index method and array method

는 방식에는 Fig. 3과 같이 array 방식과 index 방식이 존재하는데, array 방식은 버텍스가 중복되더라도 삼각형을 이루는 버텍스들을 연속으로 붙여서 그리는 방식으로 속도가 빠르지만 중복된 버텍스로 인해 메모리 사용량이 많은 것이 단점이다. 반면, index 방식은 버텍스 중복이 없는 대신 삼각형을 이루는 버텍스들의 index를 사용하는 방식이다. 따라서 메모리 사용량이 작다는 장점이 있으나, 속도가 느리다는 단점이 있는 것이 특징이다. 공간정보 오픈플랫폼의 3차원 건물데이터는 항공영상을 이용하여 도화한 데이터로 버텍스가 많지 않기 때문에 여러 삼각형에 중복되어 사용되는 버텍스 수도 적다. 따라서 index 방식의 장점이 극대화되지 않고 array 방식의 단점이 드러나지 않으므로 index 방식의 index array 제거하고자 한다.

둘째, normal vector array type을 float에서 byte로 변경한다. 이는 노말(normal) 벡터의 데이터 타입을 float에서 byte로 변경하는 것을 의미하는데, 3차원 지도 서비스의 특성상 원거리에 작은 크기로 가시화 되는 건물들이 많다. 이러한 건물들은 리소스를 소모할 뿐만 아니라 텍스처의 효과도 미비하다. 따라서 이와 같은

원거리 건물들은 텍스처를 입히지 않고 노말만을 사용하여 입체효과를 나타내는 방식이 효과적일 수 있다. 현재 오픈플랫폼은 텍스처를 사용해 건물을 가시화하므로 노말을 사용할 필요는 없다. 따라서 향후 성능 개선을 위해서는 텍스처 제거를 고려해야 하지만, 기존 포맷에서 이 정보에 대해 정의하여 사용하고 있으므로 byte로 변경하여 정교함을 낮추는 수준으로 데이터 사이즈를 축소하고자 한다.

셋째, UV(텍스처 좌표) array 타입을 float에서 short로 변경한다. 지도 클라이언트에서 많은 건물들을 가시화하기 위해서는 고해상도 텍스처를 사용하기는 어렵다. 또한 오픈플랫폼의 3차원 건물 데이터의 텍스처는 강남구와 여의도 데이터를 기준으로 최대 해상도가 '2,048 × 2,048'로 이루어진다. 따라서 unsigned short(0~65535) 수준에서도 텍스처 좌표 사이즈로 충분하기 때문에 float에서 short로 변경하여 UV의 정교함을 낮추어 데이터 사이즈를 축소하고자 한다.

포맷을 개선하기 위하여 제안할 포맷은 JSON 기반의 서비스 명세와 바이너리 기반의 건물 데이터 명세로 나누어 구성한다. 서비스 명세는 'Tile meta, Tile data, Texture data, Indoor data, Attribute'와 같이 5가지 서비스 응답으로 정의되어 클라이언트가 요구하는 타일에 대한 정보와 타일에 속한 건물 정보, 그리고 건물과 관련된 텍스처, 실내정보, 속성정보들에 대한 정보들을 제공한다. 건물 데이터 명세는 건물을 구성하는 리소스 정보로서 건물의 외관을 가시화하기 위한 필수정보들과 향후 실내정보나 지하정보로의 확장을 고려한 정보들로 구성된다. 하나의 객체를 다수의 객체로 복사하기 위해 필요한 데이터 명세를 추가로 정의함으로써 확장성을 고려한 정보들을 저장할 수 있는 방안을 마련하였다. 서비스 명세와 건물 데이터 명세를 정리하면 Table 4와 같다.

Table 4. Components of proposed format

Type	Contents	Detail
Service specification	Tile meta	Response to API what tiling method applied to the 3D data service.
	Tile data	Response to API what 3D building data contained in the particular tile.
	Texture data	Response to API requesting additional texture of a building.
	Indoor data	Response to API requesting indoor 3D model data of a specific building.
	Attribute	Response to API requesting specific building property information
Building data specification	Meta data(mandatory)	Meta-information of buildings.
	Simple model(mandatory)	Geometry of building face.
	Model mesh(option)	Original model data contained in the building.
	Compound reference (option)	Information needed to place the original model in space and make it a real instance.

#### 4.2 제안 포맷 렌더링 속도 테스트

제안한 3차원 건물데이터 포맷의 개선된 성능을 확인하기 위해 기존 오픈플랫폼 포맷과의 렌더링 속도를 비교하였다.

기존에 구축된 여의도 데이터로 테스트를 진행하였다. 테스트 데이터는 876개의 여의도 건물 데이터 복사본 100세트에 이루어졌고, 전체 삼각형 개수는 236,428개, 전체 버텍스 개수는 709,284개(중복 허용), 457,708개(중복 제거)이다.

테스트 데이터는 기존 데이터 포맷(index 방식), 버텍스 중복을 제거한 기존 데이터 포맷, 기존 포맷을 array 방식으로 변경한 포맷, 제안한 데이터 포맷(array 방식)의 총 4가지 형태로 구성하여 테스트를 진행하였다.

테스트는 1회 실험 시 대상 데이터를 500번 렌더링하고 측정된 시간을 500으로 나눈 수치를 기록하는 방식으로 수행하였고, 각 데이터별 렌더링 시간 테스트 결과는 Table 5와 같다.

테스트 결과, 기존 포맷에서 버텍스 중복을 제거한 것만으로 미비하지만 렌더링 속도가 개선된 것을 확인할 수 있었다. 또한 기존 포맷을 array 방식으로 변경한 데이터와 제안한 포맷 데이터의 렌더링 속도 개선 효과를 통해 현재 오픈플랫폼 포맷의 경우 index 방식보다 array 방식의 효율이 좋음을 확인할 수 있었다. 이는 오픈플랫폼 데이터의 경우 텍스처를 사용하는 객체로 버텍스 재사용이 적어 중복 버텍스 상쇄가 충분히 발생하지 못하기 때문인 것으로 판단된다.

기존 포맷을 array 방식으로 변경한 데이터가 제안한

포맷에 비해 미세하게 빠른 렌더링 시간을 보였다. 그러나 이는 데이터 사이즈 축소 효과를 고려했을 때 충분히 상쇄될만한 효과라 판단된다. 기존 데이터 포맷은 노말과 UV를 float형으로 정의하였고 제안한 포맷은 unsigned short로 정의하여 기존 20byte의 크기를 8byte로 축소하였다.

기존 포맷과 제안 포맷의 미세한 렌더링 속도 차이는 바인딩 하는 방식의 차이로 인한 결과로 생각되고 해당 부분은 기존 포맷 방식을 따르는 것이 타당하다.

## 5. 결론

공간정보 오픈플랫폼의 3차원 건물데이터 포맷은 2007년 초기 구축 당시에 정의된 포맷으로서 향후 서비스 변화에 대비하고 상호운영성을 향상시키기 위해 개선이 요구되고 있다. 또한 고사양이 요구되는 3차원 서비스를 효과적으로 운영하기 위해서는 원천 데이터의 경량화를 통한 메모리 관리가 필수적이다. 이에 따라 오픈플랫폼의 기존 데이터 포맷과 CityGML, KML, glTF 포맷의 특징들을 분석하였고, 최근 3차원 지도 서비스 환경에 적합한 새로운 포맷을 제안하였다.

제안하는 포맷은 5가지 서비스 명세와 4가지 건물 리소스 정보 관련 명세로 구성하여 웹 서비스 환경에서 효율적인 운영이 가능하고 건물 정보는 최대한 경량화하는 방향으로 설계하였다. 이후 기존 포맷과 기존 포맷에서 중요한 특징들을 변경한 포맷, 그리고 제안한 포맷으로 동일한 지역의 샘플 데이터를 구성해 렌더링 테스트를 진행하였고, 제안한 포맷의 렌더링 속도 개선 효과를 확인하였다.

실제 서비스 환경에서는 다양한 요인들이 성능에 영향을 줄 수 있으므로 향후 추가적인 테스트를 통해 성능 개선 요구들을 도출해 내고 서비스 환경 검토를 통해 불필요한 요소를 제거하는 연구가 필요할 것이다.

## 감사의 글

본 연구는 국토교통부 도시건축개발사업 연구비지원(13도시건축A02)에 의해 수행되었음.

## References

1. GitHub, 2016, KhronosGroup/glTF, GitHub, Inc., <https://github.com/KhronosGroup/glTF>
2. Jang, H. S., Hong, S. H., Kim, M. S. and Jang, I. S., 2015, Service status analysis about the spatial

Table 5. Results of rendering test

Unit: mili-seconds				
Count of test	Original format	Original format (Remove duplicate vertex)	Original format (Translate to array method)	proposed format
1	1034	994	818	820
2	1038	1040	804	815
3	1051	1028	807	829
4	1039	1025	810	822
5	1046	1031	815	816
6	1062	1041	808	824
7	1049	1014	820	811
8	1059	1036	816	815
9	1061	1039	800	815
10	1036	1024	825	813
Mean	1047.4	1029.6	812.3	817.5

- information open platform based on the analysis of web server log and system log, Journal of Korea Spatial Information Society, Vol. 23, No. 3, pp. 45-54.
3. jang, H. S., Jang, J. S., Go, J. H. and Jang, I. S., 2016, Performance comparative analysis of open source software for the new generation of V-World architecture configuration, Journal of the Korean Society for Geospatial Information Science, Vol. 24, No. 3, pp. 19-27.
  4. Kang, J. A. and Hwang, J. R., 2014, A study on the user satisfaction surveys and analysis for improvement of the V-World service, Journal of the Korea Spatial Information Society, Vol. 22, No. 6, pp. 23-32.
  5. Kang, J. H., Kim, J. O. and Lee, J. S., 2014, Specification validation method for improving KML interoperability, Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, Vol. 32, No. 4-1, pp. 353-361.
  6. Kim, M. S. and Jang, I. S., 2015, Design and implementation of 3D geospatial open platform based on HTML5/WebGL technology, Journal of the Korea Spatial Information Society, Vol. 23, No. 6, pp. 57-66.
  7. Lee, S. H., Cho, T. H. and Kim, M. S., 2014, A study on the service status of the spatial open platform based on the analysis of web server user log: 2014.5.20.~2014.6.2. log data, Journal of the Korea Spatial Information Society, Vol. 22, No. 4, pp. 67-76.
  8. Li, K. J., Kim, T. H., Ryu, H. G. and Kang, H. K., 2015, Comparison of cityGML and indoorGML-a use-case study on indoor spatial information construction at real sites-, Journal of Korea Spatial Information Society, Vol. 23, No. 4, pp.91-101.
  9. Park, S. H. and Lee, J. Y., 2009, Comparative analysis of 3D spatial data models, The Journal of GIS Association of Korea, Vol. 17, No. 3, pp. 277-285.
  10. SpaceN, 2016, VWorld 3D Map Binary Data File Format, SpaceN, Co., Ltd., [http://www.vworld.kr/notice/VWrold\\_Data\\_Format\\_D.pdf](http://www.vworld.kr/notice/VWrold_Data_Format_D.pdf)
  11. SpaceN, 2017, Spatial information open platform portal service, SpaceN, Co., Ltd., [http://www.vworld.kr/v4po\\_main.do](http://www.vworld.kr/v4po_main.do)
  12. SpaceN, 2017, Vworld Brief 2017. 2., pp. 1-8.