Streamlining 3D Data Integration with Standardized Web APIs

Ingo Simonis
Open Geospatial Consortium
Kriftel, Germany
isimonis@ogc.org

Melisa Harder
Open Geospatial Consortium
Denver, CO
mharder@ogc.org

ABSTRACT

The capability to analyze and visualize data in 3D is critical to any modeling and simulation exercise. Nowadays, a variety of solutions and standards co-exist to access and transfer 3D content over the internet (e.g., 3D Tiles, I3S, glTF, CDB, CityGML). These solutions use different distribution mechanisms, are optimized for particular user requirements and bandwidth situations, and deliver data in different formats (e.g., as an image stream, scenes, or raw vector data). All these differences make integration into analysis and visualization environments challenging. At the same time, we see a growing number of available real-world 3D resources with enormous potential for holistic and detailed 3D views of real-world and augmented situations.

What is lacking, however, is a resource model and corresponding API to access and query 3D content from various sources in their native format. The Open Geospatial Consortium (OGC) community is filling this gap by designing a new Web API with corresponding resource models for a variety of existing 3D standards. This is done from the ground up via the OGC Innovation Program (IP). The IP is a collaborative research and development program that advances innovative concepts from low technology readiness levels up to the highest levels of market integration and expansion.

This paper describes the OGC GeoVolume API, a new Web API that allows access to, and query of, 3D geospatial data, with a corresponding container format for streamed data delivery compatible with glTF. The GeoVolume API supports smooth transitions between 2D and 3D environments, allows applications to get 2D, 2.5D, and 3D resources, and enables 3D bounding volumes to support multiple types of 3D content.

ABOUT THE AUTHORS

Dr. Ingo Simonis is Chief Technology Innovation Officer at the Open Geospatial Consortium (OGC), an international consortium of more than 530 businesses, government agencies, research organizations, and universities, driven to advance geospatial information technology at all levels of data handling, processing, sharing, and analytics. As the head of research and innovation, he is responsible for the integration of the latest technology trends and research results into the organization and coordinates OGC's global research activities. Dr. Simonis develops OGC's research strategy, is responsible for planning, managing and execution of OGC's R&D initiatives, and supports member organizations in their adoption process of state-of-the-art, open, distributed, standards-based enterprise architectures. His own research concentrates on distributed software architectures for data sharing and collaborative knowledge generation. Before joining OGC staff, he created his own business in Germany and served as research group leader at the Council for Scientific and Industrial Research (CSIR) South Africa.

Melisa Harder is the Director of Member Development, joining the OGC from the private sector where she helped Fortune 500s innovate using technology to “take away the fear of a bad decision”. She has worked with both international organizations as well as startups in the geospatial industry. For OGC Melisa works to drive innovation, membership, and member engagement. She is trained in usability, research, and ecommerce practices and lives in Colorado.

Streamlining 3D Data Integration with Standardized Web APIs

Ingo Simonis
Open Geospatial Consortium
Kriftel, Germany
isimonis@ogc.org

Melisa Harder
Open Geospatial Consortium
Denver, CO
mharder@ogc.org

INTRODUCTION

The capability to analyze and visualize real-world 3D geospatial content is critical to any modeling and simulation exercise. The use case spectrum spans from flight training simulators to high fidelity mission rehearsal operations. Most of these scenarios need to integrate multiple data from different providers into a single application. These data originate from drones flying over the area of interest, satellite data collected from space, and field observations from sensors on the ground. With almost every satellite product offered in its native format, all sensors producing data at different scales and sampling frequencies, and the wide variety of drone and on-ground sensor data, data integration is a significant challenge. The variety of products derived from the raw data, e.g., 3D building data derived from radar missions or environment data derived from laser scanning, adds further complexity to the integration challenge. What is required is a user-centric approach that: first allows one to discover all 3D content for an area or volume of interest (see figure 1); second, a hierarchical model allows the user to filter out the required content and supported format; and third an API that allows accessing this content via a limited set of standards-based interfaces and formats. To further allow the integration with non-3D content, the native 3D content formats are aligned with APIs for sensor data access.

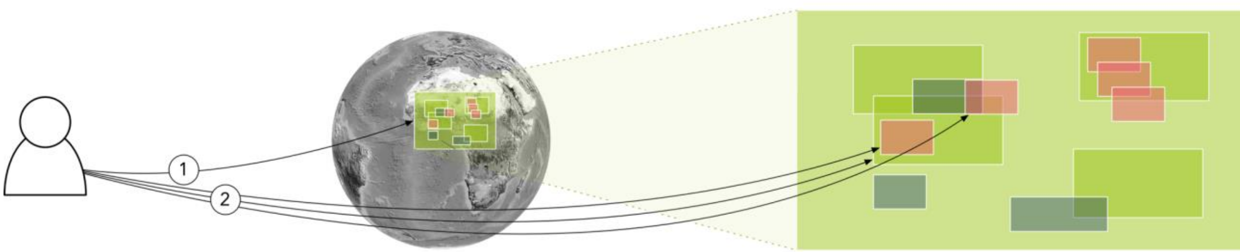

Figure 1: User (left) discovers all 3D data for area of interest (1), then binds selected subsets (2)

Client or application developers need to integrate all relevant data to represent a holistic view of the real world. Various solutions and standards co-exist to access and transfer 3D content over the internet. However, these solutions again use different distribution mechanisms, are optimized for particular user requirements and bandwidth situations, and deliver data in different formats. All these differences make integration into analysis and visualization environments challenging. At the same time, we see a growing number of available real-world 3D resources with enormous potential for holistic and detailed 3D representations of real-world situations and their augmentation.

Substantial progress is being made within single environment solutions, such as the Common Database Standard “OGC CDB” [Reed 2021, Saeedi et al. 2017]. Here, data content and representation of the synthetic environment are normalized, structure and organization of the synthetic environment are defined, and file formats for physical storage on media are standardized. As said before, the situation is different in cases where 3D content from various sources needs to be visualized, fused, or analyzed. What is lacking is a resource model and corresponding API to access and query 3D content from multiple sources in their native format. The Open Geospatial Consortium (OGC) community fills this gap by designing the new GeoVolumes Web API with corresponding 3D data container resource model (called the 3D GeoVolume) for a variety of existing 3D standards. Both are built from the ground up via the OGC Innovation Program, a collaborative research and development program that advances innovative concepts from low technology readiness levels up to the highest levels of market integration and expansion.

This paper first provides a short overview of space-centric 3D data and delivery formats. It then introduces a high-level distribution architecture for 3D content over the internet. The distribution architecture allows discovering, accessing, and retrieving relevant 3D content from a variety of sources. Access and delivery are optimized with respect to available bandwidth and processing capabilities at the client, which results in different distribution models and formats even for the same dataset. This is particularly important for mobile clients, where the available bandwidth may vary during operation. For example, the client may receive lots of detailed data while connected at high bandwidth and switch to a different distribution format when the available bandwidth degrades.

The paper then abstracts from the actual content and introduces the 3D GeoVolume. The 3D GeoVolume is a data container format for 3D content that can aggregate different 3D distribution and access formats and protocols in a single information resource. As such, a 3D GeoVolume can describe any 3D content, the access point to retrieve the data, and the distribution format for that data in a single resource. The 3D GeoVolume allows describing different formats for a single dataset and different datasets in a single format. The 3D GeoVolume is described by an enclosing 2D, or 3D bounding volume and represents 3D model datasets by reference to one or more distributions.

The 3D GeoVolume resources are served at the 3D GeoVolumes API, which is defined in the following section. Combined, the 3D GeoVolume and the GeoVolumes API demonstrate substantial progress towards a multi-source multi-format solution that integrates various data sources into a single operational picture. In addition, the approach further supports the integration of moving objects observed by a variety of distinct sensors into a single scene, enriching the scene with live sensor feeds. As an example, imagine a scenario of a post-disaster city. The scenario includes 3D buildings and road networks, cars as moving features observed in real-time, weather conditions as observed by sensors, and building and road surface conditions derived from multi-spectral satellite imagery combined with real-time drone data.

The paper concludes with experiences made and lessons learned during several multi-vendor integration pilots and design experiments, outlines the commercial potential of the described solutions, and defines a way forward towards widespread use and standardization.

SPACE-CENTRIC 3D DATA & DELIVERY FORMATS

A variety of technologies have been developed to store, tile, access, and deliver 3D content over the internet. All these technologies have a different focus. On the storage side, CDB, CityGML, Collada, OBJ, X3D, and OpenFlight are used widely to represent 3D content on media. Tiling, here in the sense of geospatial tiling, is the focus of 3D Tiles, I3S Scene Layer Package, and OGC CDB. While glTF, I3S and OpenFlight are focused on transmission.

Table 1: Overview of focus areas for 3D content standards

Focus On	**Storage**	**Tiling**	**Transmission**
Relevant Standards	CDB CityGML Collada OBJ X3D OpenFlight	3D Tiles I3S & I3S Scene Layer Package	glTF I3S OpenFlight

Not all these technologies are developed as open, consensus-driven standards, and all have their strengths and weaknesses. Whereas CDB provides sophisticated support for 2D, and 3D tiled raster, vector, and metadata combined with strong semantics, 3D Tiles is optimized for WebGL rendering and web transmission with adaptive quadtree tiling, replacement and additive refinement, Draco mesh compression and textures scaling. In saying this, however, various comparisons and performance tests have shown that best practices are often more important than the actual choice of standard.

From a geospatial perspective, storage and tiling formats such as 3D Tiles, Indexed 3D Scene Layer (I3S), CityGML as well as OGC CDB are particularly important here, complemented by glTF for efficient data transmission over the internet.

3D Tiles
The OGC 3D Tiles Community Standard [Cozzi et al 2019] specifies a model and encoding for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds. In addition, the community standard defines a hierarchical data structure and a set of tile formats that deliver content that can be rendered, with glTF being one option.

I3S
OGC Indexed 3D Scene Layer (I3S) and Scene Layer Package Format Specification [Reed & Belayneh 2020] is an OGC Community standard that specifies a model and encoding format for transmitting 3D content as well as a persistence model for scene layers. A scene layer is a container for arbitrarily large amounts of heterogeneously distributed 3D geographic content, supporting coordinate reference systems and height models in conjunction with a rich set of layer types.

CityGML
CityGML [Kutzner et al. 2020] is an open data model for 3D city data supporting various feature types such as buildings, water bodies, land use, transportation, vegetation, and others. CityGML allows modeling, storing, and exchanging semantic 3D city models and is governed as an international standard by the Open Geospatial Consortium (OGC) [Gröger et al. 2012]. Version 2.0 of the standard was adopted by OGC in March 2012. To increase the usability of CityGML for more user groups and areas of application, the OGC CityGML SWG and the Special Interest Group 3D (SIG 3D) of the initiative Geodata Infrastructure Germany (GDI-DE) started in 2013 to work on the next major version CityGML 3.0, which the OGC approved in June 2021.

CDB
The CDB standard [Reed 2021] defines a standardized model and structure for a single, "versionable", virtual representation of the earth. A CDB structured data store provides a geospatial content and model definition repository that is plug-and-play interoperable between database authoring workstations. Moreover, a CDB structured data store can be used as a common online (or runtime) repository from which various simulator client-devices can simultaneously retrieve and modify, in real-time, relevant information to perform their respective runtime simulation tasks.

DISTRIBUTION ARCHITECTURE

The high-level architecture implements the Point of Need (PoN) concept for simulation, mission command, execution and rehearsal, intelligence, and collective training. As illustrated in the figure below, the high-level architecture assumes three entities A, B, and C all being part of a distributed enterprise system. Communication between these entities is realized with varying bandwidths and connection reliability, illustrated in Figure 2 by the varying stroke thickness and dash pattern. The three entities further differ in the amount of data that can be stored, processed, analyzed, and visualized. Most to least powerful capacities are indicated by font size and object size.

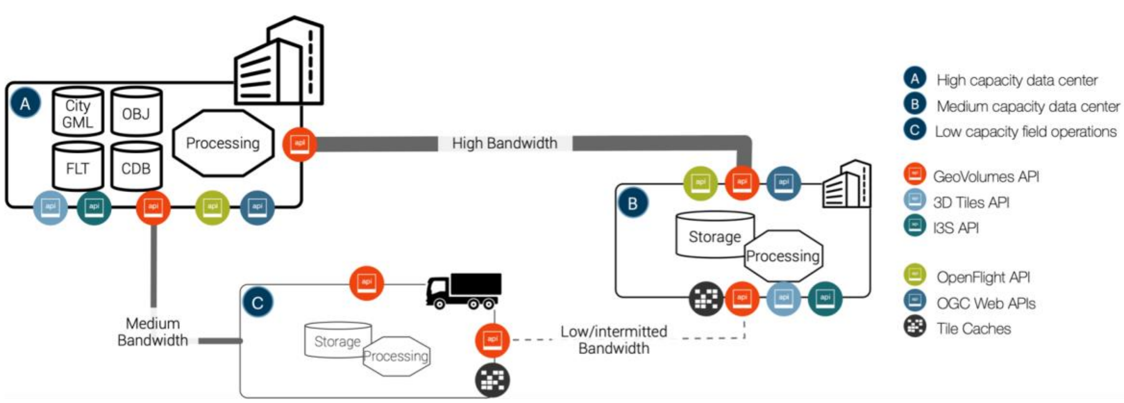

Figure 2: High level architecture with different data centers (A & B), field operations (C), available bandwidth (high, medium, low), and offered interfaces

As discussed before, holistic exploration of 3D content for an area or volume of interest requires integrating data offered in different native formats. The Point of Need concept requires relevant information available at a single point independently of the origin of the data. Given that data volumes and bandwidth constraints do not allow transmitting the original data, focus needs to be on 3D content being streamed with just the necessary level of attribution to support required analytics. To facilitate discovery and access to 3D content offered by various sources in different native formats, an API is needed to abstract from the specifics of the various distribution formats. Thus, the objective is to provide a single common API that supports a single model of 3D data in space but also acknowledges available 3D data formats and distribution standards.

In Figure 2 above, each entity A, B, and C supports the GeoVolume API as a single API to describe the entity's 3D content offerings, features data discovery, and mediates requests to single format APIs. These single format APIs include 3D Tiles, I3S, CityGML via OGC API - Features or - Tiles (both being part of OGC Web APIs), or CDB/OpenFlight and allow the customer to retrieve 3D data in the optimal format for their respective tasks. Tile caches are available for situations where insufficient bandwidth requires the usage of offline data containers.

3D GeoVolume / 3D Data container

The 3D GeoVolume is an information resource described by an enclosing bounding volume. The 3D GeoVolume has an enclosing bounding box and contains at most one 3D model dataset relevant to that volume. The 3D model dataset is represented by reference to one or more distributions.

A bounding volume is a closed volume containing the union of a set of geometric objects. The following figure illustrates typical bounding volumes (red) with enclosed objects (yellow, green, and blue cuboids). Typically, the bounding volume is a simple shape like a sphere, rectangular box, or convex hull that can be tested for intersection or overlap.

Figure 3: Bounding volumes: Box (left), region (convex hull, middle), sphere (right)

GeoVolumes follow a conceptual organization of space applied by humans, which is a nested collection of spaces where every space contains either a number of sub-spaces or a set of objects. As an example, the GeoVolume "Earth" contains a set of child GeoVolumes, one for each continent. Each continent then may have a set of child

GeoVolumes for the various countries, or, if countries are irrelevant in that scenario, a number of datasets that represent the topography, rivers, and human settlements.

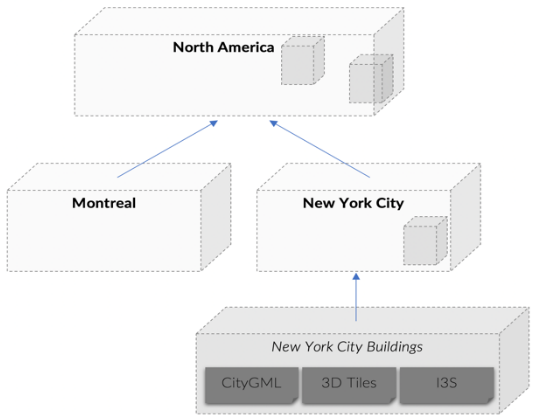

Figure 3. Hierarchical organization of 3DCs with varying distribution formats

Thus, 3DC may describe a collection of spatially disjoint GeoVolumes or a nested, hierarchical collection of GeoVolumes. This concept is illustrated in figure 3. Here, the GeoVolume "North America" contains the two child GeoVolumes "Montreal" and "New York City". Both are spatially disjoint. The GeoVolume "New York City" again contains a single 3D model dataset representing buildings. These buildings are available in distribution formats CityGML, 3D Tiles, and I3S.

The default representations of a 3D GeoVolume are json/html information documents that define the bounding box, link to an implicit tileset scheme if applicable, and provide links to the actual content. The figure below shows the formal definition of the 3D GeoVolume.

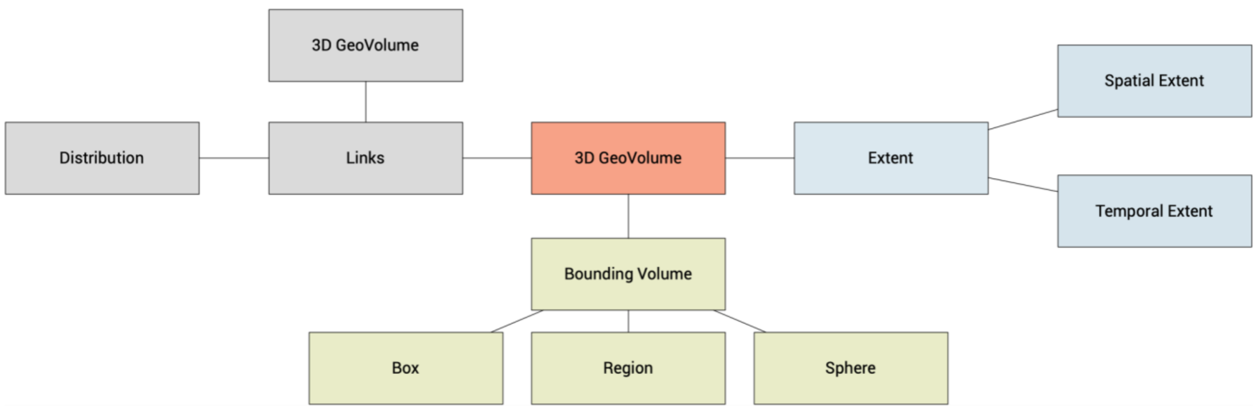

Figure 4: 3D GeoVolume model schematic view

Each 3D GeoVolume is a spatial information resource with a distinct *bounding volume* and a mandatory enclosing bounding box (*extent*). The bounding volume is a closed volume that completely contains the union of a set of objects enclosed in the volume. It is expressed as an oriented bounding box (*box*), a geographic region in WGS84 with the minimum and maximum latitudes, longitudes, and heights above or below the WGS84 ellipsoid (*region*), or a bounding *sphere*.

The 3D GeoVolume contains a 3D model dataset by reference (*links*) to one or more *distributions*. The reference targets for the various distribution formats depend on the format itself. They are defined as tilset.json file for 3D-Tiles, the 3dNodeIndexDocument for I3S, a feature collection information document or CityModel for CityGML, the root folder of a CDB data store, or a feature collection information for OGC API - Features.

Optionally, a 3D GeoVolume contains references to other 3D GeoVolume as illustrated in Figure 3 above, which are fully contained by the parent container's bounding volume. Additional properties provide the necessary information about the type of relationship, e.g., to link to an enclosing 3D GeoVolume or to the top-level 3D GeoVolume.

GEOVOLUME API

The GeoVolume API is a RESTful, HTTP-based API defined in OpenAPI version 3.0. The API provides access to 3D resources and allows the exchange of content compliant with the 3D GeoVolume model described above. At the root level, the API provides all essential information to the customer, including a landing page to describe itself, information about supported capabilities and features, and the OpenAPI definition itself. The only other offered resource is a collection of 3D GeoVolumes. The default representations of the GeoVolumes API are json/html information documents. The following table lists the offered resources, corresponding access paths, and required HTTP access method.

Table 2: GeoVolumes API Root Level

Resource	Path	HTTP	Description
Landing page	/	GET	Self-description of the API, mostly intended for human consumption, includes metadata about the offered 3D GeoVolume(s)
Conformance declaration	/conformance	GET	Description of the implemented features and capabilities of the API as defined in the API standard
API definition	/api	GET	The OpenAPI v.3.0 definition of the API
Collections	/collections	GET	Entry path to all 3D GeoVolumes
3D GeoVolume	/collections/{3DGeoVolume-ID}	GET	Specific 3D GeoVolume

In the simplest case, a GeoVolumes API server only provides the API definition, a minimum conformance declaration and some metadata about the offered 3D GeoVolumes. Servers offering many 3D GeoVolume products require additional features to facilitate efficient discovery and access to specific containers. These additional requirements are handled by optional conformance classes. As an example, the spatial query extension allows adding a spatial query element to the container collection path. Additional extensions will be developed in the future to allow organization and discovery of 3D GeoVolumes by other criteria, e.g., by theme, creator, or production date.

IMPLEMENTATION EXPERIENCES

The GeoVolumes API together with the 3D GeoVolume model has been tested in several OGC Innovation Program initiatives. These initiatives - mostly executed in the form of multi-day sprints or 2-6 months short, rapid development exercises - are open to anyone to participate. They support the OGC standardization process, which is an open, consensus process that creates royalty free, publicly available geospatial standards. The original 3D GeoVolume model and GeoVolumes API was developed by the OGC 3D Container and Tiles API Pilot, conducted between October 2019 and July 2020. The focus of the pilot was to create an open standard to support the exchange and visualization of 3D data offered and delivered in the form of other emerging 2D and 3D data standards such as 3D Tiles, I3S, glTF, CDB, and CityGML. The goal of the 3D Data Container and Tiles API Pilot was not to replace existing APIs and distribution models for 3D data, but to develop an integration concept for existing OGC 3D delivery standards to support smooth transitions between 2D and 3D environments; allow applications to get 2D, 2.5D, and 3D resources; and enable 3D bounding volumes to support multiple types of 3D content. This first pilot brought six organizations together and developed and tested the 3D GeoVolume model and GeoVolumes API with six server and five client implementations. All clients have been tested with all server implementations and used 3D-Tiles, I3S, and CityGML served at an OGC API - Features server for data exchange.

Figure 5: Five client implementations accessed six GeoVolume API servers serving 3D Tiles, I3S, and CityGML data

Experiences during this first initiative have proven the validity of the chosen hierarchical organization and multi-format delivery concept. For many participants, the implementation experience of the API was relatively linear. Once the participant's native API implementation was aligned to the specifications and schemas outlined in the 3D GeoVolume model, only minor modifications were required to achieve and support successful technical interoperability experiments with all other GeoVolume API servers. It has been emphasized that the concept could be applied to any data set and is not constrained to 3D data. The 3D GeoVolume solution could be extended to serve a single data set in the form of maps through a downstream mapping API, as vector features served by a downstream features API, or in the form of a coverage by a downstream coverages API. Thus, the 3D GeoVolume can be used for any type of multi-format content delivery even in completely different data types.

The OGC Interoperable Simulation and Gaming Sprint 1 [Daly & Serich 2021] further explored the 3D GeoVolume model and corresponding GeoVolumes API in a practical exercise with focus on handling and integration of glTF models coming from multiple sources. The Sprint did not uncover any major defects with the 3D GeoVolume model or API specification but confirmed the maturity and usability of the specification. Still, several inconsistencies were discovered that floated back into the emerging specification. These inconsistencies are mainly related to URL handling and transactions, HTTP request methods, and media type handling.

Servers either define links to the existing document as relative or absolute paths. Another pain point with URL handling is the different mappings of path characters such as slashes and colons across operating systems and their usage in HTTP. IETF RFC8820, Uniform Resource Identifier (URI), specifies that a colon is a reserved character that needs to be URL-encoded. This requirement may be sufficient for URI access but may cause issues in an environment that integrates static file-mode access, including Windows-based servers. So trivial to handle on the client-side, small inconsistencies like these can be easily avoided to further improve interoperability across client applications and server instances.

The handling of HTTP transactions is another starting point for debates. In principle, the HTTP specification provides sufficient guidance on using HTTP methods such as GET, PUT, POST, or DELETE. However, often it is API specific aspects that cause interoperability failures during 3D model updates.

On the positive side, the Sprint successfully experimented with GeoVolume API instances on the Amazon Web Service platform, integrating with other API specifications and existing standards to access sensor data, dynamic updates to both terrain and models in the datastores, and partial integration with Unity's game engine.

OUTLOOK / NEXT STEPS

The 3D GeoVolume model and GeoVolumes API have been tested during a number of open OGC Innovation Program initiatives. The basics of the GeoVolumes API specification with 3D-Content model are complete and well-specified. The integration of GeoVolumes with the game engine Unity provides a promising outlook that should be further explored with other engines such as Unreal. The support of the API by these game engines would enable engine-based applications to utilize the GeoVolumes API without the need for extensive graphics development.

ACKNOWLEDGEMENTS

Development of the 3D GeoVolumes open API and encoding was supported by the US Army Geospatial Center (AGC). Additionally, the sprint activities supported by the United States Special Operations Command (USSOCOM).

REFERENCES

[Cozzi et al. 2019] Cozzi, Patrick et al. (2019) 3D Tiles Specification 1.0. OGC 18-053r2. Retrieved June 18, 2021 from http://docs.opengeospatial.org/cs/18-053r2/18-053r2.html

[Daly & Serich 2021] Daly, Leonard and Scott Serich (2021) Interoperable Simulation and Gaming Sprint Engineering Report. OGC 20-087. Retrieved June 18, 2021 from http://docs.opengeospatial.org/per/20-087.html

[Gröger et al. 2012] Gröger et al. (2012) OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0. OGC 12-019. Retrieved June 20, 2021 from https://portal.ogc.org/files/?artifact_id=47842.

[Kutzner et al. 2020] Kutzner, Tatjana., Chaturvedi, K. & Kolbe, T.H. (2020) CityGML 3.0: New Functions Open Up New Applications. PFG 88, 43–61 (2020). https://doi.org/10.1007/s41064-020-00095-z

[Reed 2021] OGC (2021) Volume 1: OGC CDB Core Standard: Model and Physical Data Store Structure. OGC 15-113r6. Retrieved June 18, 2021 from http://docs.opengeospatial.org/is/15-113r6/15-113r6.html

[Reed & Belayneh 2020] Reed, Carl and Tamrat Belayneh (2020) OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package Format Specification. OGC 17-014r7. Retrieved June 18, 2021 from http://docs.opengeospatial.org/cs/17-014r7/17-014r7.html

[Saeedi et al 2017] Saeedi, Sarah et al. (2017) Overview of the OGC CDB Standard for 3D Synthetic Environment Modeling and Simulation. In: ISPRS Int. J. Geo-Inf. 2017, 6, 306; doi:10.3390/ijgi6100306