International Journal of
Geo-Information

MDPI

Article

Representing Time-Dynamic Geospatial Objects on Virtual Globes Using CZML—Part II: Impact, Comparison, and Future Developments

Liangfeng Zhu [1,2,3,*], Zhiwen Li [1,2] and Zhongliang Wang [1,2]

[1] Key Laboratory of Geographic Information Science (Ministry of Education), East China Normal University, Shanghai 200241, China; hi@zhiwenli.com (Z.L.); wzlh100@126.com (Z.W.)
[2] School of Geography Science, East China Normal University, Shanghai 200241, China
[3] Shanghai Key Lab for Urban Ecological Processes and Eco-Restoration, East China Normal University, Shanghai 200241, China
* Correspondence: lfzhu@geo.ecnu.edu.cn; Tel.: +86-136-7172-1009

Received: 24 January 2018; Accepted: 12 March 2018; Published: 14 March 2018

Abstract: This is the second and final part of our Cesium Markup Language (CZML) study. Here, we describe the relevant applications, academic influence, and future developments of CZML. Since its emergence in 2011, CZML has become widely used in the geoscientific environment. It is also having a positive impact on geoscience. Numerous applications use CZML for generating time-dynamic geovisualization, facilitating data interoperability, and promoting spatial data infrastructures. In this paper, we give an overview of the available tools and services, representative applications, as well as the role that CZML plays for geoscientific research. Furthermore, we also discuss key similarities and differences between CZML and KML (Keyhole Markup Language), and outline some of the future improvements for CZML's research and development.

Keywords: CZML; KML; time-dynamic visualization; geospatial object; interoperability

1. Introduction

In recent years, the importance of time-dynamic geospatial objects and their virtual globe-based applications has been rapidly increasing [1–15]. At present, the most favorite schema for describing custom geospatial objects is the Keyhole Markup Language (KML) [16–18]. However, KML is designed for the purpose of expressing geographic annotation and visualization [18]. It has several serious limitations because it lacks advanced capability and the necessary flexibility in representing time-dynamic geospatial objects [2,3,19–21]. Its shortcomings have become more obvious as scientific users struggle to visualize and analyze time-dynamic geospatial data on virtual globes. The need for providing a more modern, universal, and web-friendly schema to describe time-dynamic geospatial objects on virtual globes becomes paramount. Cesium Markup Language (CZML), which is a JSON (JavaScript Object Notation)-based markup language, is designed as an alternative schema that allows users to easily describe time-varying objects [22,23]. As an emerging specification for the representation and exchange of time-dynamic geospatial objects on virtual globes, CZML has substantial advantages. In the accompanying first paper of our series, we described the overall structure of CZML, and explored two key issues and the associated solutions for representing time-dynamic geospatial objects using CZML: the first is how to use CZML properties to describe time-varying characteristics of geospatial objects, and the second is how to use CZML to support streaming data. In this final part, we concentrate on the implementation and applications of CZML in the geospatial industry.

In subsequent sections, we present the relevant applications, academic influence, and future developments of CZML. We give an overview of the available tools and services, representative applications, as well as the constructive role that CZML plays for geoscientific research. Then, we discuss key similarities and differences between CZML and KML, and outline some of the future improvements for CZML's research and development.

2. Impact: Tools, Applications, and the Role of CZML

As a general purposed scene description language for data-driven visualization, CZML has produced a positive effect on the representation, analysis, and exchange of time-dynamic geospatial objects since its emergence in 2011. By integrating with a variety of applications that focus on the dynamic, interactive representation of geospatial objects, CZML has facilitated the development of geovisualization. It also advances the progress of spatial data infrastructures and data interoperability, since it can be conveniently transformed into other data formats [24,25].

2.1. Tools and Services

In general, the users of CZML can process CZML data in the same way that they deal with JSON data. All the current major JSON tools, such as JSONUtil [26], simplejson [27], and fastJSON [28], are also available to read and write CZML. In addition to those general purposed JSON tools, there are some tools that claim to support the functionality of CZML especially. All of those tools are open-source products that can be obtained from GitHub [29]. Most of those tools are libraries that are designed for the purpose of reading, writing, and converting CZML data using one or more programming languages. Moreover, there are several plugins and services that support CZML by providing import and/or export interfaces. The most important of them are listed in Table 1. Among these tools, Systems Tool Kit (STK), which is a sophisticated modeling and visualization system for aerospace and defense applications, claims to support the functionality of CZML completely, and provides an option to export three-dimensional (3D)/four-dimensional (4D) simulation scenarios to CZML-formatted files. The created CZML files can be drag-and-dropped directly into other CZML clients to be shared with the world [25]. These tools and services prove that CZML has the potentiality for data interoperability.

At present, CZML-formatted geospatial objects can only be viewed in web browsers running the Cesium virtual globe, but they also have the potential to be visualized in other online or offline two-dimensional (2D)/3D clients [22]. If they so choose, users ought to create their own clients that are compatible with CZML. In the official website of Cesium [30], there is an online application, termed Cesium Viewer [31], for loading and displaying CZML files. Users can simply drag-and-drop a CZML file from their desktops into Cesium Viewer, and the viewer will automatically load and visualize geospatial objects described in the file. Moreover, Cesium Sandcastle [32], a live code editor and example gallery for the Cesium virtual globe, provides dozens of demos and examples of CZML. It allows for users to write and run CZML code without leaving the web browser, and easily see how changes in CZML affect the display on the virtual globe. Beginners of CZML can learn and experiment with CZML using this interactive prototyping tool freely.

With the increased attention paid to the time-dynamic geospatial objects, as well as the wide spread of the CZML specification and the rapid growth of the ecosystem for open-source CZML tools, it is likely that more and more virtual globe clients would support CZML in the foreseeable future.

Table 1. Tools and services supporting Cesium Markup Language (CZML).

Tool and Service	Description	Developer	URLs
		Libraries for CZML	
czml-writer	A library for writing CZML content using .NET and Java.	Analytical Graphics, Inc.	https://github.com/AnalyticalGraphicsInc/czml-writer
czml-Python	An open source python library to read and write CZML files.	Christian Ledermann	https://github.com/cleder/czml
json2czml	A data conversion tool for converting GeoJSON into CZML.	Umetsu Hidehiro	https://github.com/hideume/json2czml
carthe_czml	A python program to produce CZML files from CARTHE GLAD drifter data.	mjturtora	https://github.com/mjturtora/carthe_czml
gpx2czml	A javascript module that converts gpx data to CZML data.	KernYoo	https://github.com/trustyoo86/gpx2czml
kml2czml	A Java library for converting KML geometry models to CZML models.	workingDog	https://github.com/workingDog/kml2czml
TLE2CZML	A library for converting TLE (Two-Line Element) lists to CZML.	Michael Bowman	https://github.com/bowmanmc/tle2czml
ScalaCZML	A library for reading and writing CZML JSON entities and presenting them as Scala objects.	workingDog	https://github.com/workingDog/scalaczml
		Plugins for CZML	
czml_generator	A QGIS plugin for creating CZML files.	Mátyás Gede	https://github.com/samanbey/czml_generator
Systems Tool Kit (STK)	STK offers an option for exporting 3D/4D simulation scenarios to CZML-formatted files.	Analytical Graphics, Inc.	https://www.agi.com/products/stk
		Services providing CZML	
Eclipse Scraper	A Python module for scraping tabular data for the tracks of eclipse events from NASA's eclipse website and converting such events into usable CZML documents.	Christopher Clark	https://github.com/Frencil/eclipsescraper
		3D viewers for CZML	
Cesium Viewer	A Cesium reference application supporting drag-and-drop a CZML file from the desktop into the viewer.	Analytical Graphics, Inc.	https://cesiumjs.org/Cesium/Build/Apps/CesiumViewer/index.html
		Examples of CZML	
Cesium Sandcastle	A code editor and example gallery for the Cesium virtual globe, involving a large number of CZML examples.	Analytical Graphics, Inc.	http://cesiumjs.org/Cesium/Apps/Sandcastle/index.html
Offline-satelite-tracking-czmll	A script for the Tracing of the specifc satellite positions based on the specific time using the CZML and sgp4 libraries and finding the position.	Muhammad Shafay Amjad	https://github.com/shafaypro/Offline-satelite-tracking-czmll

2.2. Applications

As a new encoding schema originally designed for the Cesium virtual globe specifically, the development of CZML has kept pace with the growth of Cesium. Cesium was founded by Analytical Graphics, Inc. (AGI) in 2011 as an open-source JavaScript library for creating 3D virtual globes and 2D maps in a web browser without a plugin. It can be run cross-platform, cross-browser, and cross-device because it utilizes the Web Graphics Library (WebGL) to provide hardware acceleration and plugin independence. Users of Cesium can integrate, visualize, and analyze a variety of geospatial information, such as global high-resolution terrain and imagery, many standard vector data, and 3D models [33,34]. For a complete presentation of the major benefits of Cesium, the reader is referred to the Cesium website (http://cesiumjs.org) [30].

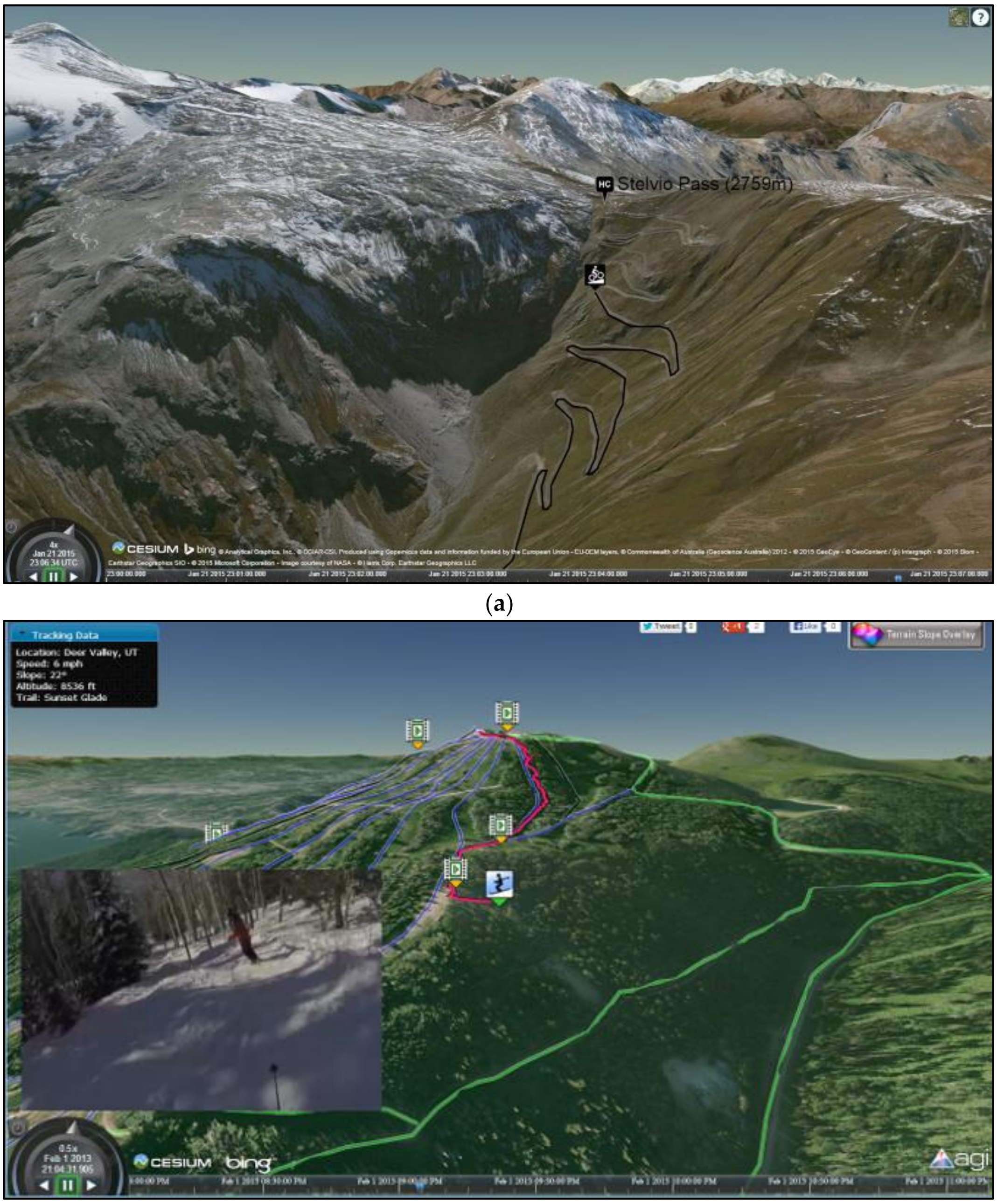
Up to now, Google Earth has been regarded as the most popular virtual globe in both public and scientific communities. One of the reasons for Google Earth's success is that it offers users the capability to customize and interact with their own geospatial data through KML and the Google Earth application programming interface (API) [35,36]. At present, there are plenty of Google Earth-based applications that utilize KML to create dynamic, interactive displays of geospatial information. However, due to the increasing incompatibility of the Google Earth API with modern web browsers, Google decided to retire the Google Earth API [1,37], and it did not provide any clear alternatives for the established web-based 3D visual globe applications. Therefore, many Google Earth-based applications are facing the problem of transition. Nowadays, an increasing number of Google Earth API developers are migrating to Cesium from Google Earth [38–40]. This technological trend also provides a great opportunity for CZML to expand its power rapidly, thus helping to shape the future development of the CZML ecosystem.

CZML has first-class treatment of time-dynamic data and is usually kept in step with Cesium capabilities. In recent years, more and more virtual globe developers began using CZML to represent time-dynamic geospatial objects. Accompanied by the Cesium virtual globe, CZML has facilitated and enabled the building of numerous applications in many fields, such as geospatial, aerospace, sports, environmental science, defense industry, and more. These applications primarily use CZML to generate dynamic data visualizations on virtual globes. Here, we bring together a cross-selection of the most representative examples of how CZML is being used for representing time-dynamic geospatial objects explicitly. The focus is on applications in which three specific features of CZML are essential: representing time-dynamic trajectories of moving objects, recording and rendering complicated geometry objects, as well as expressing 3D/4D thematic information.

2.2.1. Representing Time-Dynamic Trajectories of Moving Objects

Representing moving object trajectories is an important application field of CZML, supporting specific tasks, such as capturing and recording the real-time position of objects that are moving in space over time, as well as interpolating and visualizing the path of individual moving objects. There are a wide variety of CZML applications to reflect the spatio-temporal variation process of different kinds of moving objects, such as human beings, animals, aircraft, and satellites.

Among these applications, some are targeted to represent human trajectory. For example, Cycling the Alps, which is an online application that has been migrated from Google Earth to Cesium [41], allows for users to explore the routes of all cycling climbs in the Alps mountain range with many different visual features through a bike tour (Figure 1a). In this application, CZML is employed together with Cesium to create the start and finish points, and dynamic routes when users travel in between. Powder Tacks [42] is an application to integrate GPS ski track location data with videos taken on a ski trip to Salt Lake City, UT. In this application, the skier's trajectories derived from the raw GPS track data are automatically converted into CZML *path* properties. A series of CZML *billboard* properties are used to indicate the skier's current position and the location of available videos. Videos taken during the trip are synchronized with the skier's position and play alongside the animating visualization (Figure 1b). Camera flights provide smooth transitions to the next location when a user selects a new video to watch. Similarly, Relive the Day 10 Battle [43] is an online application that is designed for the purpose of tracking the Red Bull X-Alps 2015 (an intense, multi-day hiking and paragliding race across the Alps), as well as displaying the athletes' progress, evaluating their stats and monitoring their rankings (Figure 1c). In this application, CZML is used to represent racer's locations, and Cesium is used for showing them on a 3D virtual globe.

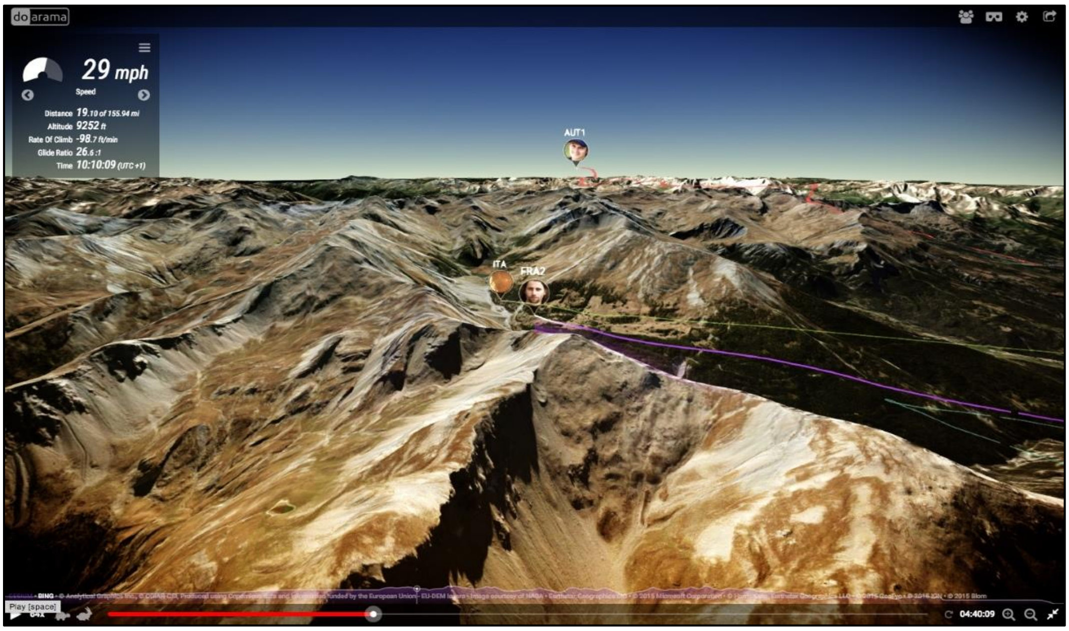


(a)

Tracking Data
Location: Deer Valley, UT
Speed: 6 mph
Slope: 22°
Altitude: 8536 ft
Trail: Sunset Glade

(b)

Figure 1. *Cont.*

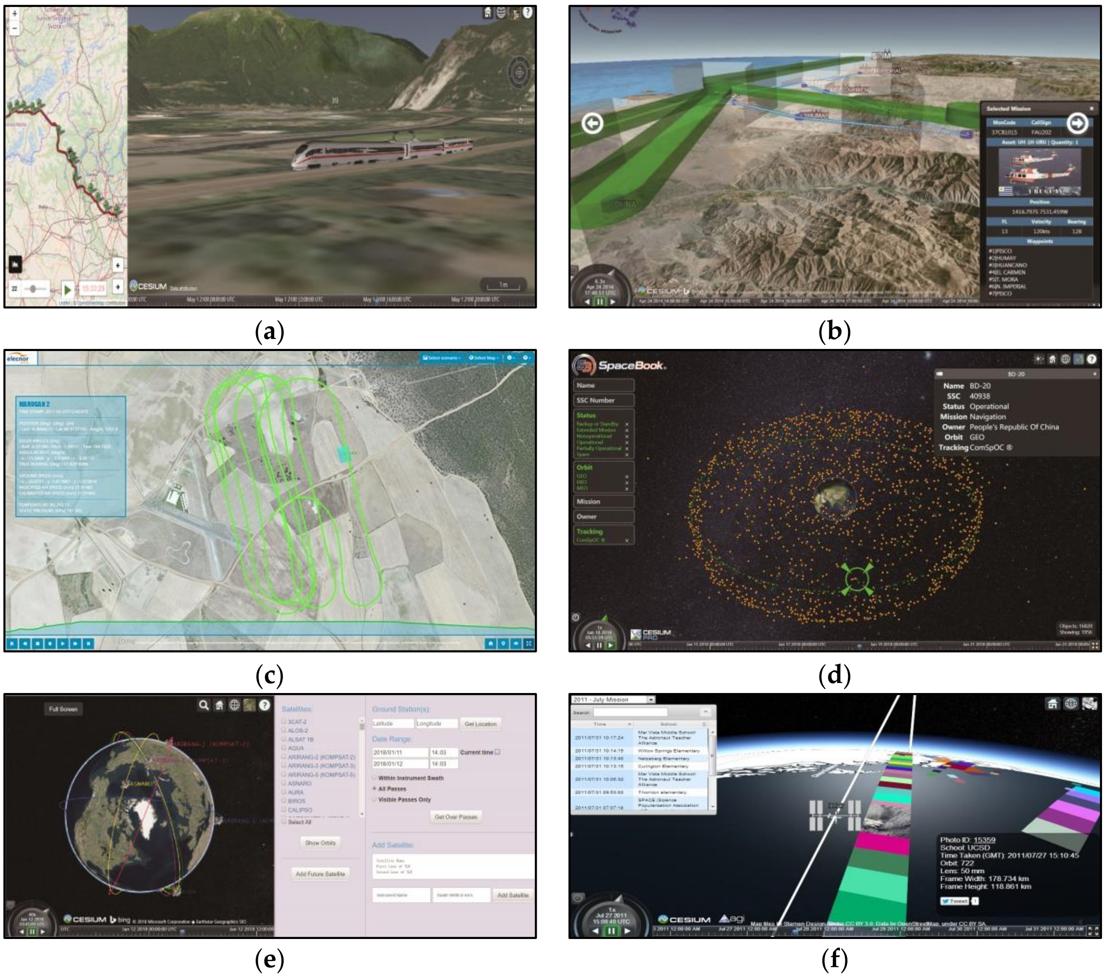


29 mph
Speed
Distance 19.10 of 155.94 mi
Altitude 9252 ft
Rate Of Climb -98.7 ft/min
Glide Ratio 26.6 :1
Time 10:10:09 (UTC +1)

(**c**)

Figure 1. CZML can be used to represent human trajectory, e.g., (**a**) exploring the routes of cycling climbs through a bike tour (image adapted from Cycling the Alps [41]); (**b**) integrating GPS ski track location data with videos taken on a ski trip (image adapted from Powder Tacks [42]); or, (**c**) tracking the athletes' progress, evaluating their stats, and monitoring their rankings (image adapted from Relive the Day 10 Battle [43]).

In addition to its use for representing human beings, CZML can be used to provide relevant information on trains, aircraft, satellites, and other moving objects. For example, Geoglyph Rail [44], which is an online train simulator with over 300 train lines and more than 100 countries' railways, uses CZML to write the data for each train trip (Figure 2a). ATO Visualizer [45], an online application for interactive visualization of Air Tasking Orders (ATOs), utilizes CZML to list air sorties for a fixed 24-h period, containing positions and mission information about a large number of aircraft (Figure 2b). ADAM [46] is a project to represent 3D virtual drone flights, and to visualize the images and the flight data captured by the onboard instruments in an interactive way (Figure 2c). In this application, CZML is used to record the positions, orientations, and time data for Unmanned Aerial Vehicles (UAV). Real-time CZML data are dynamically updated by a geographic data server (GeoServer) to improve the visualization performance on clients. SpaceBook [47], which is a Cesium-based service, provides an interactive, easy-to-use interface to monitor the performance and safety of satellites in a timely manner while tracking the activity of neighboring resident space objects (RSOs). More than 16,000 space objects being tracked by the United States (US) government, including their positions, status, and trending information are recorded in CZML format, incrementally loaded, and displayed into the Cesium virtual globe in real time, taking advantage of the streamability of CZML (Figure 2d). Orbital Predictor [48] is an online application that offers users the capability to visualize the orbits of groups of satellites, and to plan imaging campaigns. The satellites' orbits are generated from the satellites' two-line element sets and are encoded in the CZML file format (Figure 2e). Similarly, EarthKAM Explorer [49] adopts CZML to represent the International Space Station (ISS) EarthKAM and its orbit information on a global scene (Figure 2f). It also uses CZML generated from the EarthKAM data to visualize polygons where satellite images were taken. In EclipseTracks [50], an interactive solar eclipse track modeling system, NASA eclipse tables are transmuted into valid CZML documents to predict and render the track of solar eclipses occurring about once every 18 months.



(**a**) (**b**)

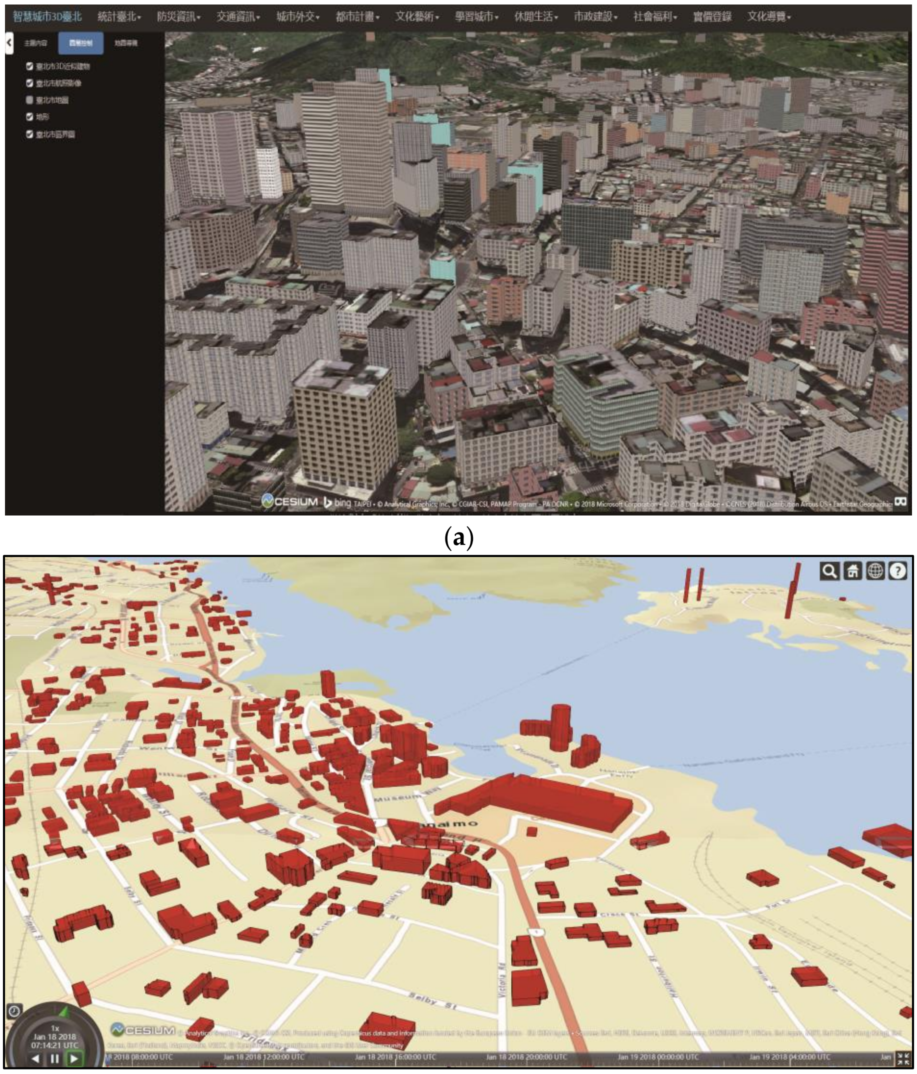
(**c**) (**d**)

(**e**) (**f**)

Figure 2. CZML can be used to provide relevant information on trains, aircraft, satellites and other moving objects. (**a**) A train trip (image adapted from Geoglyph Rail [44]); (**b**) Air sorties (image adapted from Air Tasking Orders (ATO) Visualizer [45]); (**c**) three-dimensional (3D) virtual drone flights (image adapted from ADAM [46]); (**d**) A large number of resident space objects (RSOs) (image adapted from SpaceBook [47]); (**e**) The orbits of groups of satellites (image adapted from Orbital Predictor [48]); (**f**) The International Space Station (ISS) EarthKAM and its orbit information (image adapted from EarthKAM Explorer [49]).

2.2.2. Recording and Rendering Complicated Geometry Objects

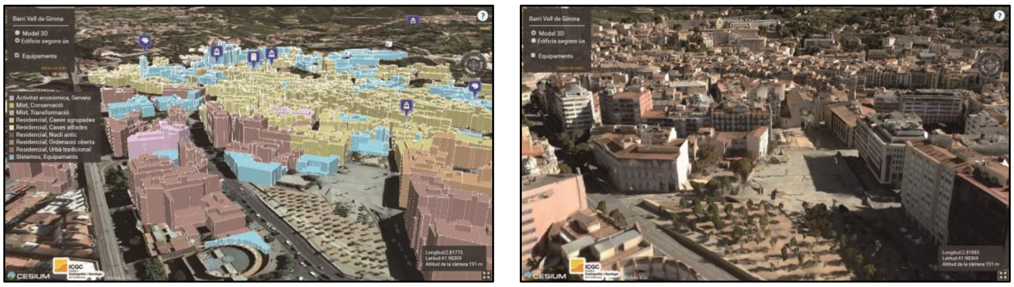
CZML can serve as an effective tool to record and render geometry objects, like points, lines, polygons, and 3D models, because it is convenient to describe complicated geometric shapes. One such example is Taipei 3D City Navigation [51], which hosts a variety of 3D models, including 300,000+ buildings, 200,000+ trees, 100,000+ road lamps, and 20,000+ traffic lights. Those 3D models are grouped into a set of CZML files, according to their geo-locations and building volumes (Figure 3a). A scheme similar to the region/LOD used in KML is employed to render the models efficiently. Similar applications of CZML that require the geometry properties of CZML are Nanaimo Buildings (Figure 3b) [52], and Old Town of Girona (Figure 4) [53].



(**a**)

(**b**)

Figure 3. CZML can be used to record and render complicated geometric shapes. (**a**) A variety of 3D textured models in the virtual 3D Taipei city (image adapted from Taipei 3D City Navigation [51]); and, (**b**) A styled layer of buildings in Nanaimo, BC, Canada (image adapted from Nanaimo Buildings [52]).

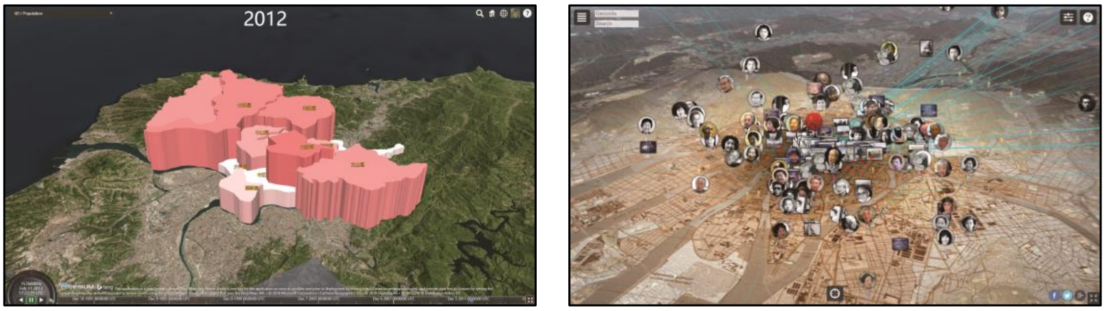


(**a**) (**b**)

Figure 4. A CZML application in the Old Town of Girona, a historical section of Catalonia, Spain. It is a good example of combining data collected from different data sources, in different formats and from different origins, and converting them to a unified CZML format. (**a**) A cube view of the buildings within Girona and their usage (zoning); and, (**b**) Standard 3D models with photographic textures. Images adapted from the Old Town of Girona [53].

2.2.3. Expressing 3D/4D Thematic Information

CZML is also applicable to present thematic information because it has a variety of visual variables to dynamically express 3D/4D data, as well as the capability of incremental streaming for time-dynamic visualization. One such example is 4D Statistics Maps [54], which is an online application aimed to improve users' ability to understand complex information and geovisual analytics by dynamically displaying changing data, such as population change over time (Figure 5a). In the Hiroshima Archive [55], a digital archive for visualizing the specific locations of survivors at the time of the Hiroshima atomic bombing and their movements afterwards, CZML is used to express the victims' spatial distribution and migration, as well as the impact of the disaster on regional development through detailed text description, vivid animations, photographs, and panorama pictures (Figure 5b). Other similar applications of CZML include the Nagasaki Archive [56], Okinawa Archive [57], East Japan Earthquake Archive [58], and Indian Ocean Tsunami Archive [59].



(**a**) (**b**)

Figure 5. CZML can be used to dynamically express 3D/four-dimensional (4D) thematic information, e.g., (**a**) displaying time-sequential population data (image adapted from 4D Statistics Maps [54]); or, (**b**) visualizing the victims' spatial distribution and migration at the time of the Hiroshima atomic bombing (image adapted from Hiroshima Archive [55]).

Moreover, some of the user-created applications of CZML are illustrated in Cesium Demos [60] and the AGI website [61]. These practical applications indicate that CZML is well-suited for representing time-dynamic geospatial objects when they are integrated into a virtual globe platform.

2.3. The Role of CZML in Geoscientific Research

Representation and analysis of geospatial objects on virtual globes are particularly useful as they can help earth scientists to make new inferences about the Earth, to produce new knowledge, and to promote new scientific discoveries [34,62,63]. Nowadays, there are plenty of exploratory studies on how best to model and visualize 2D/3D geospatial information (such as maps, images, and 3D models) on virtual globes [64–66]. Visualizing and analyzing geospatial objects in four dimensions (x, y, z, and time), however, is still a relatively new field [4–15,67–70]. With the increased attention being paid to the fourth dimension of the Earth space, and the significant improvements to acquire spatio-temporal data, more and more people are getting interested in the representation and analysis of time-dynamic geospatial objects on virtual globes [71,72]. 4D visualization enables users to view and analyze time-dynamic objects with any other time-dependent or time-independent spatial dataset in a natural and intuitive way. This would dramatically improve the efficiency of exploring relationships hidden behind the complex and large volume of geospatial data, and lead to an increased comprehensive understanding of the Earth [63].

CZML is especially designed for the purpose of representing time-dynamic geospatial objects on virtual globes. It adopts a distinctive, but efficient, compact method to record, describe, and transmit 4D spatio-temporal information. It is a preferred data format for visualizing and analyzing time-varying data in the network environment, and is supported in numerous virtual globe applications and geospatial software products (cf. Section 2.2). Geoscientific research can benefit from CZML, since its capabilities to accurately represent the geometrical, graphical, and semantic aspects of time-varying geospatial objects, and since progressive model transmission over the Internet is supported by the streamable structure and data-driven visualization provided by CZML (cf. the accompanying first paper of our series). Geoscientists, researchers, and CZML developers have used CZML to showcase their scientific data and results to the public in visually appealing ways, to help explain scientific phenomena, and to explore the relationships among time-varying geospatial objects. With the increased audience and the in-depth identification for the advantages and conveniences of CZML, more and more tools, services, Earth browsers, and data sources that support CZML will appear. If CZML could evolve into a full-fledged international standard for representing time-dynamic geospatial objects, it will promote the visualization and analysis of geospatial objects, encourage interoperability and sharing of geospatial information, and eventually change the way we interact with 4D spatio-temporal data.

3. Comparison with KML

KML is the most influential language in the field of representation, visualization, and analysis of geospatial objects on virtual globes. In this section, CZML is compared to KML, helping us to make a comprehensive understanding about the characteristics of CZML. KML and CZML share a number of similarities with each other, as well as some important differences. Both CZML and KML are open, non-proprietary data formats for representing geospatial objects and describing scenes in their respective clients, and can be generated by a wide variety of applications, and possibly even written by hand. Additionally, both are meant to be sufficiently client-agnostic that other compatible clients can render the scene described therein [22]. However, CZML has a number of important characteristics, some of that distinguish it from KML. As listed in Table 2, CZML is different from KML in five aspects: goals and purposes, grammatical rules, information expressions, data characteristics, as well as supports and applications.

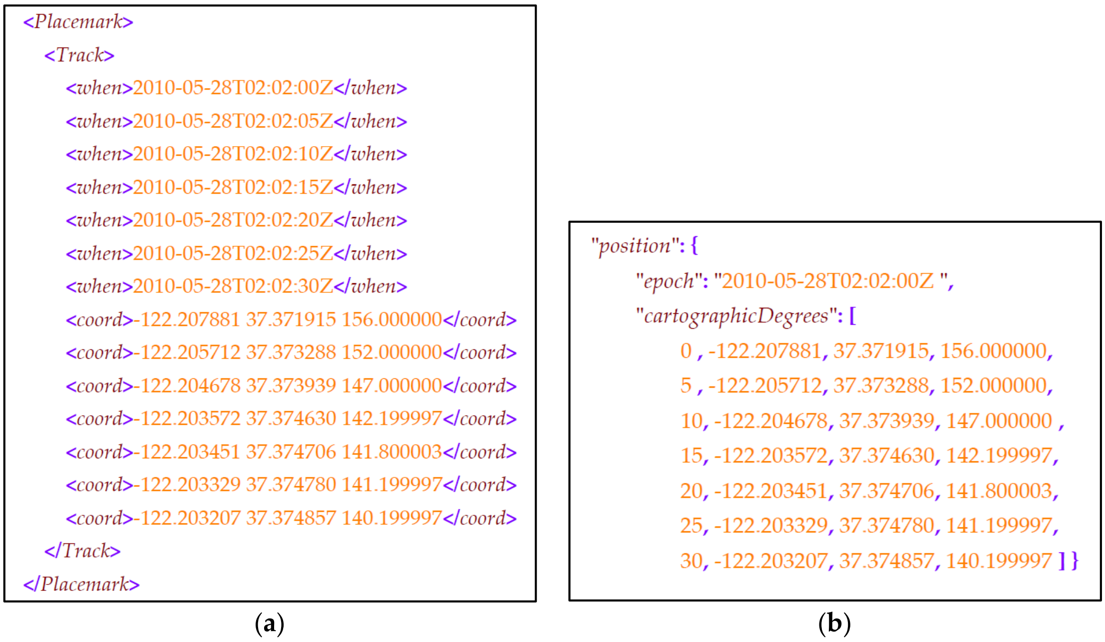
Table 2. Differences between CZML and Keyhole Markup Language (KML).

Criteria Description		KML	CZML
Goals and purposes	Design goals	Encode and transport representations of geographic data for display in 2D/3D earth browsers.	Encoding representations of time-dynamic geospatial objects on 3D virtual globes.
Grammatical rules	Grammatical basis	XML	JSON
	Object-oriented	Yes	No
	Namespace supported	Yes	No
Information expressions	Expression modes	A tag-based structure with nested elements and attributes.	Enumerating a collection of name/value pairs.
	Geometry supported	Six primitive geometry elements and two multiple geometry elements.	Fifteen geometry properties with additional sub-properties.
	Time-varying information	Offering very limited capabilities to represent time-dynamic geographical objects.	Both the geometry, appearance and the semantics of geospatial objects can be changed over time.
Data characteristics	Data volume	Large, verbose, and slow.	Less, concise, and fast.
	Expansibility	Flexible. Offering three ways to add custom data to a KML Feature.	Weaker. Only one way to deal with custom data.
	Streamability supported	No	Yes
Supports and applications	OGC recommendation	Yes	No
	Supporting platforms	Google Earth, Cesium and other comparable earth browsers.	Cesium

3.1. Goals and Purposes

The greatest difference between CZML and KML is their design goals. KML is used to encode and transport representations of geographic data mainly for display in 2D/3D Earth browsers. The purpose of KML includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look [18]. KML does not pay much attention to geospatial objects that are changing over time. Time-dynamic geospatial objects present a significant challenge to KML because they require representing the time-varying geometry, appearance, and semantics of the most relevant objects in the Earth space. Although KML allows for users to associate a time element with any KML Feature for restricting the visibility of the dataset to a given time period or point in time, it lacks sufficient capabilities to compactly represent time series data. In KML, only one time element could be associated with a given KML Feature, except for the KML:Track and kml:MultiTrack elements [16]. In order to represent a time-dynamic geospatial objects, users have to duplicate the object many times to create a set (sometimes dozens) of KML objects, each containing a time element [36]. This leads to very large data redundancy, and greatly reduces the visualization performance in the virtual globe clients. Therefore, KML is complicated and cumbersome when it is used to describe time-dynamic geospatial objects.

CZML is specifically designed for describing time-dynamic geospatial objects and graphical scenes. From the very beginning, CZML's goal has been to represent geospatial objects, along with their properties that vary over time. It wants to be more than just traditional geometrical positions/extents, it wants to describe the semantics and appearance of geospatial objects, as well as how they change with time. CZML has first-class treatment of time and tuned for representing dynamic data accurately and conveniently. By tying the application to the delivery and processing of time-varying data sets, CZML takes advantage of the unique characteristics of CZML properties, such as time intervals and time tags, to finely present a wide range of time-dynamic geospatial objects to the user. CZML's performance is superior to that of KML when it comes to dealing with dynamic data. As shown in Figure 6, when describing time-varying characteristics of geospatial objects, CZML is significantly more compact than KML, with much more functionality and a greater saving on memory and process time.



```
<Placemark>
  <Track>
    <when>2010-05-28T02:02:00Z</when>
    <when>2010-05-28T02:02:05Z</when>
    <when>2010-05-28T02:02:10Z</when>
    <when>2010-05-28T02:02:15Z</when>
    <when>2010-05-28T02:02:20Z</when>
    <when>2010-05-28T02:02:25Z</when>
    <when>2010-05-28T02:02:30Z</when>
    <coord>-122.207881 37.371915 156.000000</coord>
    <coord>-122.205712 37.373288 152.000000</coord>
    <coord>-122.204678 37.373939 147.000000</coord>
    <coord>-122.203572 37.374630 142.199997</coord>
    <coord>-122.203451 37.374706 141.800003</coord>
    <coord>-122.203329 37.374780 141.199997</coord>
    <coord>-122.203207 37.374857 140.199997</coord>
  </Track>
</Placemark>
```

(**a**)

```
"position": {
    "epoch": "2010-05-28T02:02:00Z ",
    "cartographicDegrees": [
        0 , -122.207881, 37.371915, 156.000000,
        5 , -122.205712, 37.373288, 152.000000,
        10, -122.204678, 37.373939, 147.000000 ,
        15, -122.203572, 37.374630, 142.199997,
        20, -122.203451, 37.374706, 141.800003,
        25, -122.203329, 37.374780, 141.199997,
        30, -122.203207, 37.374857, 140.199997 ] }
```

(**b**)

Figure 6. KML (**a**) vs. CZML (**b**) snippets for describing the same path of a moving object over a specified period of time. (**a**) In KML, the time positions of a kml:Track element are captured in an array of kml:when elements and the space coordinates are captured in an equal-length array of kml:coord elements; (**b**) In CZML, a series of time-tagged samples are used to compactly represent the time-stamped positions of the path.

3.2. Grammatical Rules

Another difference between CZML and KML are the grammatical rules. KML uses a standardized syntax conforming to XML, and utilizes certain geometry elements that are derived from GML (Geography Markup Language) [73]. It represents a hierarchical data system where geographical objects can be populated in a tag-based structure with nested elements and attributes. Therefore, the KML document has a relatively complicated structure with large size and high redundancy. In contrast, CZML is a JSON-based schema and file format, and follows general rules for JSON. It uses collections of name/value pairs to enumerate the details of geographical objects. When compared with KML, CZML is significantly more lightweight in file size, clearer, and more compact in structure, and more convenient to read and write.

CZML is better than KML in terms of the syntactic complexity and learning difficulty. It is easy to parse because a CZML document only contains an array, and each element in the array is a CZML packet. CZML-formatted geospatial objects are fully defined by JavaScript objects, so CZML properties can be directly parsed by the browser JavaScript engine without any extra middle-ware layer. KML is more complicated than CZML. A KML document has a hierarchical structure with nested elements. Each KML element must be spelled exactly, as shown in the KML specification, and with the same capitalization [74]. Moreover, KML child elements must appear in the same order as listed within their parent element in the KML specification [74]. Those rules greatly increase the complexity when parsing a KML document.

KML is object-oriented, and all KML elements are derived from kml:Object, which is an abstract base class that provides the *id* attribute. CZML does not possess any prominent object-oriented characteristic (like hierarchy or inheritance). There is no clear abstraction or class tree for the CZML properties and their sub-properties.

KML supports namespaces, which provide a simple method for qualifying element and attribute names used in KML documents by associating them with namespaces that were identified by URI references. This is an effective mechanism for supporting new features that contain information beyond what is available in the standard KML specification. CZML does not support namespaces, but it also can be extended with custom properties using some sort of identifier.

3.3. Information Expressions

CZML and KML differ not only in the objectives and grammars, but also in the way in which they express information. The KML document is structured by nested tags. Comments can appear in the KML code. CZML is derived from JSON. It uses a clear and self-evident document, which consists of a number of name/value pairs, to describe time-varying properties of geospatial objects. Comments are not allowed in the CZML code. Both KML and CZML support styling inside the format and embedding CSS-like methods, which can be used to style geometry elements, page elements, and to control the size and appearance of the description balloon.

In KML, geographical objects are generalized as points, lines, polygons, and other types of geometric elements. OGC KML 2.3 supports six types of primitive geometry elements derived from the abstract kml:Geometry element, including kml:Point, kml:LineString, kml:Track, kml:LinearRing, kml:Polygon, and kml:Model [16,66]. Multiple discrete geometric primitives can be combined together to create more complicated geometries like the kml:MultiGeometry and kml:MultiTrack elements [16,17]. This approach provides great flexibility and lets users construct almost any geometry that they could dream of. However, it also leads to tremendous inconvenience: the vast majority of geometrical shapes cannot be directly represented in KML, even if they are regular and frequently-used. For instance, when representing a rectangle, users of KML are required to pre-calculate the geographic coordinates of all vertices in that rectangle. CZML overcomes this limitation through providing a more advanced graphics abstraction mechanism and more geometry primitives. It is highly capable of supporting 15 geometry properties, and each property contains plenty of additional sub-properties to control the visual appearance and the graphical style of geospatial objects. In CZML, it is easy to describe and draw various shapes and volumes on the virtual globe. Armed with basic knowledge of geometry, users can create a variety of complicated graphics by simply using the predefined CZML geometry properties. Regardless of their geometric definition, all of the CZML geometry properties have a common set of properties that control their appearance. Moreover, the CZML specification provides an effective extension mechanism for defining more complicated geometric shapes. Therefore, it is convenient to express the geometries and appearances of various geospatial objects.

KML has adequate capabilities to represent the trajectory of a moving object over a specified period of time [2,3,16]. However, its capability in describing time-dynamic objects is still restricted. It would be very difficult to use KML to represent time-varying properties of geospatial objects, especially when the properties are interpolatable. CZML has a powerful capability for expressing time-varying geospatial information, especially in the case of large collections of time-dynamic objects in the network environment. In CZML, not only can the spatial positions/extents be dynamically updated, but the semantics and appearance of geospatial objects can also be changed continuously over time. CZML also provides additional functionality than KML, for example, interpolation of unknown property values with time. With this functionality, users can create and display time-dynamic visualization using only a limited number of time-tagged samples. Thus, CZML is more expressive than KML.

3.4. Data Characteristics

KML data are verbose, which is derived from the grammatical characteristics of XML. KML files can grow very large and can be slow to load [40], especially when representing dynamically-updated geospatial data. CZML is concise and informative because it is optimized for representing time-dynamic geospatial objects on the web. For web applications, less data volume means that users will obtain faster file transfers and better data services. More importantly, CZML is a declarative JSON schema, which makes it possible to represent time-varying objects for asynchronous browser/server communication (AJAJ), and to replace XML, which is used by AJAX (cf. the accompanying first paper of our series). Thus, CZML is more suitable for disseminating and exchanging geospatial information in the network environment.

KML possesses the characteristic of good expansibility. It has a flexible extension mechanism, so custom data can be added to a KML Feature as needed for specific use cases [74]. CZML is measurably weaker than KML in expansibility. All CZML properties and their data types are predefined in the CZML specification. Custom data that are not defined in the CZML specification cannot be directly added to CZML packets as new properties. The only way to deal with custom data is to save them as an HTML snippet, and to embed them into the *description* property of the CZML packet. Finally, the custom data are displayed in the descriptive balloons of geospatial objects by default.

KML presents significant hurdles to transmit time-varying data as it lacks the ability of incremental streaming to the client. Although the interval refresh feature of kml:NetworkLink can be used to refresh the time-dynamic data automatically, the complete KML dataset needs to be fetched before clients process and display it. Moreover, it is not allowed to divide a complete KML element into several data units, which limits KML's flexibility in data distribution and transmission on the web. In contrast, as a web-friendly format, CZML is streamable and optimized for custom consumption. It is structured for efficient, incremental streaming to the client. When describing a time-dynamic geospatial object, users often have to deal with a great deal of time-dependent sampling data. In CZML, the properties of a single geospatial object can be broken into a series of CZML packets and streamed using server-sent events. The entire CZML document need not be present on the client before the scene can be displayed. This leads to incremental loading and the display of time-varying data, and eventually improves responsiveness and interactivity for visualizing and analyzing geospatial objects on the client.

3.5. Supports and Applications

KML version 2.2 is an international standard of OGC for the representation and exchange of geographic data since 2008 [17]. An updated version 2.3 has been approved by the OGC membership in October 2015 [16,18]. As an OGC recommendation, KML is widely supported by numerous virtual globe applications, including Google Earth, Cesium, and other comparable online earth browsers. Up to now, at least 93 KML implementations are registered in the OGC implementations databases [75], and the list of virtual globe applications that support KML is growing daily. Nowadays, there is a wealth of KML-formatted geospatial information shared on the World Wide Web. The wide acceptance of the KML standard allows for those KML-formatted datasets to be interoperable between different web mapping systems.

As an emerging specification for representing time-dynamic geospatial objects on virtual globes, CZML has not yet become part of the official data standards supported by OGC. It is an open format but currently only used via the Cesium virtual globe. Compared with KML, CZML still lacks adequate learning materials, data resources, as well as tools and services. However, with the increasing demand to express time-dynamic geospatial objects, the Cesium team, the governor of CZML, intends to formalize the CZML specification with a standard body, and to submit it to OGC proposed as an OGC implementation standard in the near future [22].

3.6. Summary

When compared with KML, the key advantages of CZML are that: it is compact, easy to learn, use, and parse; it is convenient to accurately describe geospatial objects varying with time; it is optimized for client consumption; and, it supports efficient, incremental streaming to the client. The results of the above-mentioned comparison indicate that CZML is complementary to KML as the optimum data format for expressing time-dynamic objects on web-based virtual globe applications.

4. Discussion and Future Developments

As shown in the previous sections, CZML is very successful in both visual representation and data transmission of time-dynamic geospatial objects. It has been broadly accepted by the geospatial industry as a means to create time-dynamic displays. At present, an increasing number of geospatial objects are represented in CZML or corresponding databases. However, the currently available CZML specification still has limitations and, thus, future improvements are required to make CZML more efficient.

Firstly, more advanced data extension mechanisms should be introduced to support custom data. Although the current CZML specification offers the *description* property to preserve custom data, the custom data must be saved as HTML snippets, and, thus, cannot be used by external computer applications. In contrast, KML offers three ways to add custom data to a KML Feature: the kml:Data element allows users to add untyped name/value pairs to the user data associated with a given Feature; the kml:Schema and kml:SchemaData elements allow for users to add typed data to a Feature; arbitrary XML data allows users to preserve user data within a KML file [74]. Users of KML can choose an appropriate approach to add their data, depending on the kind of data, as well as how they plan to use the data in their KML presentations. Obviously, CZML should draw on the experience of the extended mechanism in KML. Planned future developments for CZML should include the ability to freely capture, add, and process custom data that can be used for certain data entity tasks, such as statistical analysis and semantic modelling.

Secondly, to effectively distribute time-dynamic geospatial data, more advanced technologies for streaming and updating CZML should be explored. While CZML can be incrementally streamed to the client, streaming CZML is limited to only transmitting a single document or data source. At present, dynamically loading multiple CZML documents via external links is not allowed. In the future, CZML should be expanded to support the distribution of both internal data and external links, which is similar to the KML update feature [16,17,74]. An external link can reference a CZML document on a local or remote network. This reference causes CZML clients to fetch the linked CZML document, and transmit and display it in the 3D viewer of a virtual globe. By using external links, massive time-dynamic geospatial information can be distributed into multiple CZML documents. Each document can be shared with multiple end users, and each user can refresh the most recent data automatically.

Lastly, more CZML clients should be developed in order to provide more robust supports to CZML. At present, KML performs better than CZML in code debugging and error prompt. In most of the KML clients, users can enable error checking mode to locate misplaced KML elements or other errors. However, Cesium, the only virtual globe client that supports CZML currently, cannot generate error messages to report invalid CZML properties, which creates major difficulties in editing and modifying CZML codes. Future opportunities for extension of CZML functionality should include letting Cesium remedy this performance issue through providing a sophisticated error-checking mode.

5. Conclusions

CZML makes it possible to combine time together with geospatial properties for geoscientific research and better understanding of geospatial objects around us. It is intended to be a base schema that defines time-varying aspects that are common to the most relevant applications of 3D geospatial objects. Nowadays, CZML has been widely embraced by the geospatial industry. For users that need to represent time-dynamic objects in their online virtual globe applications, CZML should be their first choice. Scientific users benefit from CZML, which makes time-varying geospatial objects more accessible to a wide audience within the geoscientific community. It is clear that CZML is transforming our ability to represent and visualize 4D spatio-temporal information on virtual globes.

CZML is an emerging specification and an ongoing development. It is supported by a vibrant and active community of developers. The CZML community expects to use CZML for a number of ongoing and new geoscience research projects that are broadly centered on time-dynamic geospatial objects, and finally intends to propose CZML as an OGC standard. All of the stakeholders in fields related to time-dynamic geospatial objects are encouraged to contribute to CZML, particularly to participate in the CZML community that promotes the improvement of the CZML specification, to develop more virtual globe applications that support CZML, to construct new tools that read, write, visualize, and convert CZML data, or to conduct usability study on the efficiency and user satisfaction of CZML.

Acknowledgments: The research leading to this paper was supported by the National Natural Science Foundation of China (Grant No. 41672327), the Natural Science Foundation of Shanghai Municipality (Grant No. 16ZR1408900), the Project of Shanghai Science and Technology Committee (Grant No. 17DZ1202804 & 17DZ1202805), and the Social Science Foundation of Shanghai (Grant No. 2014BCK002). We would like to thank the members of the Cesium team and community for their persistent engagement in developing CZML. Furthermore, we gratefully thank the editor and four anonymous reviewers for their useful comments that improved the quality of the manuscript.

Author Contributions: Liangfeng Zhu was in charge of the research work and wrote the paper. Zhiwen Li and Zhongliang Wang supervised the research work and the manuscript conception. All authors participated to the improvement of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, L.; Wang, Z.; Li, Z. Representing Time-Dynamic Geospatial Objects on Virtual Globes Using CZML—Part I: Overview and Key Issues. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 97. [CrossRef]
2. Ferreira, K.R.; Vinhas, L.; Monteiro, A.M.V.; Camara, G. Moving Objects and KML Files. In Proceedings of the IEEE International Conference on Data Engineering Workshops, Arlington, VA, USA, 1–5 April 2012; pp. 355–359.
3. Ferreira, K.R.; Vinhas, L.; Monteiro, A.M.V.; Câmara, G. Moving objects and spatial data sources. *Revista Brasileira De Cartografia* **2012**, *64*, 796–806.
4. Gonçalves, T.; Afonso, A.P.; Martins, B. Cartographic visualization of human trajectory data: Overview and analysis. *J. Locat. Based Serv.* **2015**, *9*, 138–166. [CrossRef]

5. Patel, N.N.; Angiuli, E.; Gamba, P.; Gaughan, A.; Lisini, G.; Stevens, F.R.; Tatem, A.J.; Trianni, G. Multitemporal settlement and population mapping from Landsat using Google Earth Engine. *Int. J. Appl. Earth Obs.* **2015**, *35*, 199–208. [CrossRef]
6. Andrienko, N.; Andrienko, G. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*; Springer: Heidelberg, Germany, 2006.
7. Andrienko, G.; Andrienko, N.; Bak, P.; Keim, D.; Wrobel, S. *Visual Analytics of Movement*; Springer: Heidelberg, Germany, 2013.
8. Andrienko, G.; Andrienko, N.; Demsar, U.; Dransch, D.; Dykes, J.; Fabrikant, S.I.; Jern, M.; Kraak, M.-J.; Schumann, H.; Tominski, C. Space, time and visual analytics. *Int. J. Geogr. Inf. Sci.* **2010**, *24*, 1577–1600. [CrossRef]
9. Andrienko, G.; Andrienko, N.; Wrobel, S. Visual analytics tools for analysis of movement data. *ACM SIGKDD Explor. Newsl.* **2007**, *9*, 38–46. [CrossRef]
10. Von Landesberger, T.; Brodkorb, F.; Roskosch, P.; Andrienko, N.; Andrienko, G.; Kerren, A. MobilityGraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 11–20. [CrossRef] [PubMed]
11. Amini, F.; Rufiange, S.; Hossain, Z.; Ventura, Q.; Irani, P.; McGuffin, M.J. The impact of interactivity on comprehending 2D and 3D visualizations of movement data. *IEEE Trans. Vis. Comput. Graph.* **2015**, *21*, 122–135. [CrossRef] [PubMed]
12. Andrienko, G.; Andrienko, N.; Dykes, J.; Fabrikant, S.I.; Wachowicz, M. Geovisualization of dynamics, movement and change: Key issues and developing approaches in visualization research. *Inf. Vis.* **2008**, *7*, 173–180. [CrossRef]
13. Andrienko, N.; Andrienko, G.; Gatalsky, P. Exploratory spatio-temporal visualization: An analytical review. *J. Vis. Lang. Comput.* **2003**, *14*, 503–541. [CrossRef]
14. Buschmann, S.; Trapp, M.; Döllner, J. Animated visualization of spatial-temporal trajectory data for air-traffic analysis. *Vis. Comput.* **2016**, *32*, 371–381. [CrossRef]
15. Buschmann, S.; Trapp, M.; Döllner, J. Real-time visualization of massive movement data in digital landscapes. In Proceedings of the 16th Conference on Digital Landscape Architecture (DLA 2015), Dessau, Germany, 4–6 June 2015; pp. 213–220.
16. OGC KML 2.3. OGC12-007r2. Available online: http://docs.opengeospatial.org/is/12-007r2/12-007r2.html (accessed on 23 January 2018).
17. OGC KML. OGC07-147r2. Available online: http://portal.opengeospatial.org/files/?artifact_id=27810 (accessed on 23 January 2018).
18. KML. Available online: http://www.opengeospatial.org/standards/kml (accessed on 23 January 2018).
19. Ballagh, L.M.; Raup, B.H.; Duerr, R.E.; Khalsa, S.J.S.; Helm, C.; Fowler, D.; Gupte, A. Representing scientific data sets in KML: Methods and challenges. *Comput. Geosci.* **2011**, *37*, 57–64. [CrossRef]
20. Zhu, L.; Pan, X.; Gao, G. Assessing place location knowledge using a virtual globe. *J. Geogr.* **2016**, *115*, 72–80. [CrossRef]
21. Zhu, L.; Kan, W.; Zhang, Y.; Sun, J. Visualizing the Structure of the Earth's Lithosphere on the Google Earth Virtual-Globe Platform. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 26. [CrossRef]
22. CZML Guide. Available online: https://github.com/AnalyticalGraphicsInc/czml-writer/wiki/CZML-Guide (accessed on 23 January 2018).
23. Cozzi, P.; Ring, K. *3D Engine Design for Virtual Globes*; CRC Press: Boca Raton, FL, USA, 2011.
24. STK generates CZML to share data. Available online: https://cesiumjs.org/demos/STKCZML (accessed on 31 January 2018).
25. CZML Brings STK and Cesium Together. Available online: http://www.agi.com/news/blog/february-2016/czml-brings-stk-and-cesium-together?feed=AGIBlogsFeed (accessed on 31 January 2018).
26. JSONUtil. Available online: http://jsonutil.riaforge.org (accessed on 23 January 2018).
27. Simplejson—JSON Encoder and Decoder. Available online: https://simplejson.readthedocs.io (accessed on 23 January 2018).
28. Fastjson: A Fast JSON Parser/Generator for Java. Available online: https://github.com/alibaba/fastjson (accessed on 23 January 2018).
29. GitHub. Available online: https://github.com (accessed on 31 January 2018).
30. Cesium—WebGL Virtual Globe and Map Engine. Available online: https://cesiumjs.org (accessed on 23 January 2018).

31. Cesium Viewer. Available online: https://cesiumjs.org/Cesium/Build/Apps/CesiumViewer/index.html (accessed on 31 January 2018).
32. Cesium Sandcastle. Available online: http://cesiumjs.org/Cesium/Apps/Sandcastle/index.html (accessed on 31 January 2018).
33. Hunter, J.; Brooking, C.; Reading, L.; Vink, S. A Web-based system enabling the integration, analysis, and 3D sub-surface visualization of groundwater monitoring data and geological models. *Int. J. Digit. Earth* **2016**, *9*, 197–214. [CrossRef]
34. Muller, R.D.; Qin, X.D.; Sandwell, D.T.; Dutkiewicz, A.; Williams, S.E.; Flament, N.; Maus, S.; Seton, M. The GPlates Portal: Cloud-Based Interactive 3D Visualization of Global Geophysical and Geological Data in a Web Browser. *PLoS ONE* **2016**, *11*, e0150883. [CrossRef] [PubMed]
35. Bailey, J.E.; Chen, A.J. The role of Virtual Globes in geoscience. *Comput. Geosci.* **2011**, *37*, 1–2. [CrossRef]
36. De Paor, D.G.; Whitmeyer, S.J. Geological and geophysical modeling on virtual globes using KML, COLLADA, and Javascript. *Comput. Geosci.* **2011**, *37*, 100–110. [CrossRef]
37. Announcing Deprecation of the Google Earth API. Available online: http://googlegeodevelopers.blogspot.com/2014/12/announcing-deprecation-of-google-earth.html (accessed on 5 January 2018).
38. De Paor, D.G. Beyond the Google Earth API—Cesium, a Plugin-free, WebGL-based virtual globe. In Proceedings of the Geological Society of America 2015, Baltimore, MD, USA, 1–4 November 2015; p. 187.
39. Migrating from Earth to Cesium. Available online: https://cesium.com/blog/2015/01/27/migrating-from-earth-to-cesium (accessed on 22 January 2018).
40. Migrating from Google Earth to Cesium. Available online: http://52.4.31.236/presentations/FOSS4GNA2016/GE-Cesium.pdf (accessed on 31 January 2018).
41. Cycling the Alps Migrates to Cesium. Available online: https://cesium.com/blog/2015/02/05/cycling-the-alps-migrates-to-cesium (accessed on 23 January 2018).
42. Powder Tacks. Available online: https://cesiumjs.org/demos/powdertracks (accessed on 23 January 2018).
43. Red Bull X-Alps. Available online: https://cesiumjs.org/demos/xalps (accessed on 23 January 2018).
44. Geoglyph—WebGL 3D. Available online: http://earth.geoglyph.info (accessed on 23 January 2018).
45. ATO Visualizer. Available online: https://cesiumjs.org/demos/atovisualizer (accessed on 23 January 2018).
46. 3D Virtual Drone Flights Using the Oculus Rift. Available online: https://cesiumjs.org/demos/DronesOculus (accessed on 23 January 2018).
47. SpaceBook. Available online: http://apps.agi.com/SatelliteViewer (accessed on 23 January 2018).
48. Orbital Predictor. Available online: http://www.orbitalpredictor.com (accessed on 23 January 2018).
49. EarthKAM Explorer. Available online: https://cesiumjs.org/demos/earthkamexplorer/ (accessed on 23 January 2018).
50. Eclipsetracks.org—Interactive Solar Eclipses with CesiumJS. Available online: http://eclipsetracks.org (accessed on 23 January 2018).
51. Taipei 3D City Navigation. Available online: https://3d.taipei (accessed on 23 January 2018).
52. MapGuide Rest. Available online: http://mapguide.ca/nanaimo.htm (accessed on 23 January 2018).
53. Girona-Barri Vell a vol d'ocell. Available online: http://betaserver.icgc.cat/cesium/Girona3D.html (accessed on 23 January 2018).
54. 4D Statistics Maps. Available online: https://3d.taipei/apps/4DCMap.html (accessed on 23 January 2018).
55. Hiroshima Archive. Available online: http://hiroshima.archiving.jp (accessed on 23 January 2018).
56. Nagasaki Archive. Available online: http://e.nagasaki.mapping.jp/p/nagasaki-archive.html (accessed on 23 January 2018).
57. The Peace Leaning Archive of OKINAWA. Available online: http://peacelearning.jp/okinawaNew/index.html (accessed on 23 January 2018).
58. Japan Earthquake | Nagasaki Archive. Available online: http://e.nagasaki.mapping.jp/p/japan-earthquake.html (accessed on 23 January 2018).
59. Indian Ocean Tsunami Archive. Available online: http://aceh.archiving.jp/index_jp.html (accessed on 23 January 2018).
60. Cesium Demos. Available online: https://cesiumjs.org/demos.html (accessed on 23 January 2018).
61. AGI Home. Available online: http://www.agi.com (accessed on 23 January 2018).

62. Yamagishi, Y.; Yanaka, H.; Suzuki, K.; Tsuboi, S.; Isse, T.; Obayashi, M.; Tamura, H.; Nagao, H. Visualization of geoscience data on Google Earth: Development of a data converter system for seismic tomographic models. *Comput. Geosci.* **2010**, *36*, 373–382. [CrossRef]
63. Zhu, L.F.; Sun, J.Z.; Li, C.L.; Zhang, B. SolidEarth: A new Digital Earth system for the modeling and visualization of the whole Earth space. *Front. Earth Sci.* **2014**, *8*, 524–539. [CrossRef]
64. De Paor, D.G.; Pinan-Llamas, A. Application of novel presentation techniques to a structural and metamorphic map of the Pampean Orogenic Belt, NW Argentina. In Proceedings of the Geological Society of America 2006, Philadelphia, PA, USA, 24–26 October 2006; p. 326.
65. Huang, Y.-K. Within Skyline Query Processing in Dynamic Road Networks. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 137. [CrossRef]
66. Zhu, L.F.; Wang, X.F.; Pan, X. Moving KML geometry elements within Google Earth. *Comput. Geosci.* **2014**, *72*, 176–183. [CrossRef]
67. Zhu, L.F.; Pan, X.; Sun, J.Z. Visualization and dissemination of global crustal models on virtual globes. *Comput. Geosci.* **2016**, *90*, 34–40. [CrossRef]
68. Zhu, L.F.; Wang, X.F.; Zhang, B. Modeling and visualizing borehole information on virtual globes using KML. *Comput. Geosci.* **2014**, *62*, 62–70. [CrossRef]
69. Zhu, L.F.; Li, M.J.; Li, C.L.; Shang, J.G.; Chen, G.L.; Zhang, B.; Wang, X.F. Coupled modeling between geological structure fields and property parameter fields in 3D engineering geological space. *Eng. Geol.* **2013**, *167*, 105–116. [CrossRef]
70. Li, Z.; Guan, X.; Li, R.; Wu, H. 4D-SAS: A Distributed Dynamic-Data Driven Simulation and Analysis System for Massive Spatial Agent-Based Modeling. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 42. [CrossRef]
71. Saeedi, S.; Liang, S.; Graham, D.; Lokuta, M.F.; Mostafavi, M.A. Overview of the OGC CDB Standard for 3D Synthetic Environment Modeling and Simulation. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 306. [CrossRef]
72. Zhu, L.; Zhang, C.; Li, M.; Pan, X.; Sun, J. Building 3D solid models of sedimentary stratigraphic systems from borehole data: An automatic method and case studies. *Eng. Geol.* **2012**, *127*, 1–13. [CrossRef]
73. OGC Geography Markup Language (GML)—Extended Schemas and Encoding Rules. OGC10-129r1. Available online: https://portal.opengeospatial.org/files/?artifact_id=46568 (accessed on 23 January 2018).
74. Wernecke, J. *The KML Handbook: Geographic Visualization for the Web*; Addison-Wesley: Boston, MA, USA, 2009.
75. OGC Certified and Implementing Products. Available online: http://www.opengeospatial.org/resource/products (accessed on 31 January 2018).