

Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings

Volume 16 Bonn, Germany

Article 7

2016

Is there life in Virtual Globes?

Konstantinos Evangelidis

Technological Educational Institute of Central Macedonia, Serres, Greece

Theofilos Papadopoulos

Technological Educational Institute of Central Macedonia, Serres, Greece

Follow this and additional works at: <https://scholarworks.umass.edu/foss4g>

 Part of the [Geographic Information Sciences Commons](#)

Recommended Citation

Evangelidis, Konstantinos and Papadopoulos, Theofilos (2016) "Is there life in Virtual Globes?," *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*: Vol. 16 , Article 7.

DOI: <https://doi.org/10.7275/R56M3518>

Available at: <https://scholarworks.umass.edu/foss4g/vol16/iss1/7>

This Paper is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

Is there life in Virtual Globes?

Theofilos Papadopoulos, Konstantinos Evangelidis

Technological Educational Institute of Central Macedonia, Serres, GREECE – priestont@gmail.com,
kevan70@gmail.com

KEY WORDS: Virtual Globes, 3D Geovisualizations, WebGL, Javascript

ABSTRACT:

Virtual globes have been widely used during last decade mainly for simulating observations of the earth from the outer space and navigation experiences over its surface which may be portrayed with various types of views and textures. The present work aims to extend virtual globe capabilities by incorporating three dimensional events on them. Such events may include animation and motion effects on 3D models representing real world living or inanimate spatial objects, modeling of natural resources and phenomena representations and any type of geovisualized activities and demonstrations.

1 Introduction

Virtual globes are browser-based web interfaces mainly used to offer to the user, navigation experience around the planet earth as being an observer at the outer space, with the ability in most cases to modify the observatory position and viewing angle. Various types of views are usually portrayed on the surface of the earth like for example ground terrain view, or road network map view or just aerial or satellite images of the real world. In addition, it is possible to represent three dimensional (3D) objects and convey on them types of textures. However, what is not being typically provided is animation and motion effects on 3D spatial entities for simulation and demonstration purposes.

The general idea behind the presented work is to extend current Virtual Globes capabilities by incorporating in their views, also motion and animation effects on 3D models representing living or inanimate spatial objects. Then to make it possible to create virtual geospatial worlds viewable on specified zoom levels on an area of the globe simulating various types of events such as historical, or related to flora and fauna, natural resources, phenomena etc.

What is needed to achieve the objectives described is a tool for developing layered-based geospatial scenes containing 3D actors participating in motions over a georeferenced area. Following that, a set of arrangements and specifications that will assure a proper appearance of all the above on a virtual globe. In this work, we make an attempt to develop a representative case of a living virtual globe displaying 3D scenes and events on the earth's surface and identify the development issues for such a venture.

2 Virtual Globes: Features and Trends

In the 00s, Riedl (2007) focused on the following Virtual Globe parameters: a) transportability, which refers to the capability of transporting the related geospatial data over the Internet, same way as it happens with any other digital information, b) scalability, as the capability of presenting geospatial information at any desired scale, c) interactivity, since Virtual Globes provide a web interface to receive feedback from end users, d) choice of topics as the capability of displaying unlimited number of global themes and e) currency, as the ability to present real-time or updated data at any given time span. Later on, Tuttle (2008), proposed two additional features: pseudo-3D, due to the naturally understood by people environment of Virtual Globes and client-side, which refers to the capability of end users to totally manage (insert, update, modify) the published data.

Since then, the geospatial technologies were boosted by the enormous progress occurred in information and communication technologies. We would choose as most significant, also affecting Virtual Globes, a) the developments in geospatial web semantics, which influenced data sharing, exchange and interoperability through XML based standards (KML, GML etc.) and OGC web services (WMS, CSW, WFS etc.) and b) the popularity of Javascript which triggered the development of specialized geospatial libraries.

A critical moment in Virtual Globes timeline was when they started to cooperate with Geospatial Web Services. According to Tuttle (2008) the simultaneous popularity of both technologies can not be a coincidence. In 2006 (Dunne and Shutton, 2006) discuss the powerful combination of Web Map Services with Virtual Globes. Today, it is possible to exploit geospatial services in the cloud in order to produce valuable geoinformation (Evangelidis et al, 2012) and Virtual Globes represent the mean for displaying this information.

As Javascript is nowadays the prevailing cross-platform programming language on the web compiled just-in-time on the client side, it is worth focusing on WebGL Javascript API, the standard for hardware accelerated interactive 3D graphics rendering on the web. Emerging visualization frameworks deployed on top of WebGL contribute significantly in the field of 3D geospatial developments, including those related to Virtual Globes, such as OpenWebGlobe⁷, WebGL Globe⁸ and Cesium.js⁹. An incomparable advantage of the above platforms is that they are directly executed by recent web browsers without the need of installing plug-ins or other software.

Cesium.js seems to have become the dominant geospatial visualization framework for 3D mapping on the web after Google announced its Earth API, deprecation (Angevine, 2015). It is built on top of WebGL, HTML5 standard, supports industry standard vector formats (KML, GeoJSON), Collada models when converted to its own glTF format, is open source and cross-platform, running in any browser supporting WebGL. It also supports 3D models animation and user-controlled motion over the terrain and numerous such examples were ported from Google Earth to Cesium (Fig. 1)

⁷ <http://www.openwebglobe.org/>

⁸ <https://www.chromeexperiments.com/globe>

⁹ <https://cesiumjs.org/>



Figure 1: Motion and animation effects with Cesium.js.

However, the Javascript library much respected and recognized by the developers community which also exploits WebGL renderer capabilities, with numerous contributors and free examples for animation and extreme motion effects is that of three.js (<http://threejs.org/>). It supports complex geometries and highly detailed objects, however, to the best of our knowledge, it doesn't maintain the geospatial aspect of the involved scenes and virtual worlds. Three.js could act complementary to Cesium.js and vice-versa and their mixing might be the ideal case.

3 Development

On a previous work which is under review, we had developed on top of three.js a GIS based prototype web interface for creating custom geospatial worlds containing overlaid 3D objects moving on specified or dynamically user-defined motion paths and animated textures. To assist demonstration of the above mentioned work a video has been prepared and is available at <https://goo.gl/mutZrs>.

One of the obvious challenges and further developments was to integrate the above described "living" geospatial worlds with Virtual Globes. In the present work we made an attempt to perform such integration. The expected result is to view "life" over Virtual Globes after a reasonable zoom level.

The whole development process comprises the following discrete stages:

- The development of a tool for creating custom "living" geospatial worlds (already performed as previously demonstrated)
- The development of a function for creating Virtual Globes
- The development of a function for georeferencing custom geospatial worlds with the Virtual Globe

The whole development was exclusively based on three.js library.

3.1 Development of Custom Geospatial Worlds

A GIS digital elevation model of the demo area is created by providing a set of grid points with elevation values and by employing a plane type geometry¹⁰

¹⁰ <http://threejs.org/docs/#Reference/Extras.Geometries/PlaneGeometry>

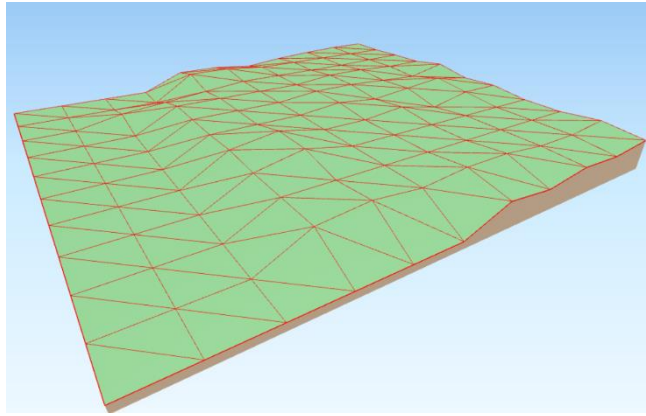


Figure 2: Digital elevation model created with plane type geometry of three.js

The texture of the geospatial world is loaded via a raster image which is adjusted appropriately

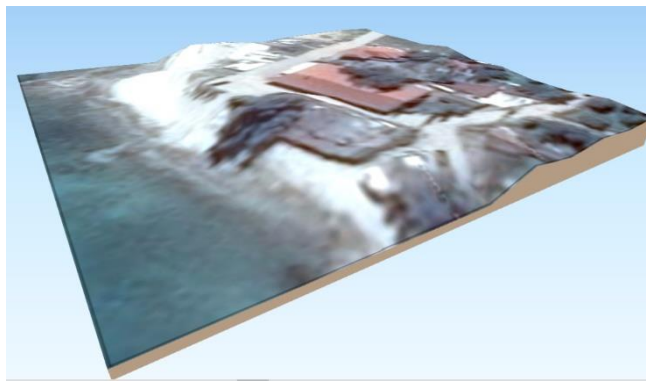


Figure 3: Surface texture adjusted over elevation model

Selected 3D models representing static spatial objects (e.g. house, plants) are scaled to fit in to the geospatial world. The static ones are placed in user-defined points, or along polylines or inside polygons as shown in figure 4.

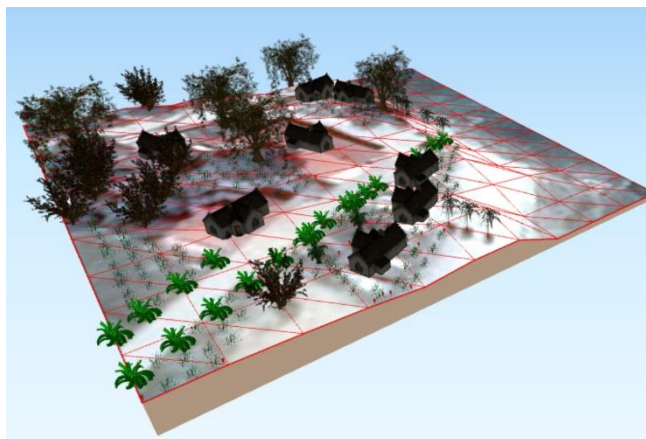


Figure 4: Adding static 3D models placed on the geospatial world

Animated 3D models representing living (e.g. horse) or inanimate spatial objects (e.g. car) are also scaled and placed in user-defined motion paths specified by polylines. Animated textures of the surface (e.g. sea waves) are

also overlaid, as shown in figure 5

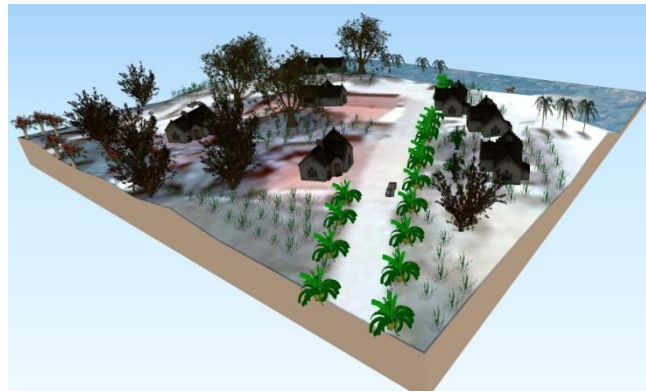


Figure 5: Overlaying motion 3D models on the geospatial world

3.2 Development of a Virtual Globe

To create a Virtual Globe in three.js using the WGS 84 main parameters the "THREE.ShpereGeometry" constructor is used with radius = 6378137 which refers to the major (equatorial) radius a in meters. To apply the polar semi-minor axis b , the flattening $f=1/298.257223563$ is used and applied to the scale method of the sphere geometry "earthGeometry.scale.set(1,1-f,1)" at the y axis (three.js axis are x,y – width,height and the z as the depth axis).

For the presentation purposes the Virtual Globe has been designed with its meridians and parallels every 10o degrees (Fig. 6) .

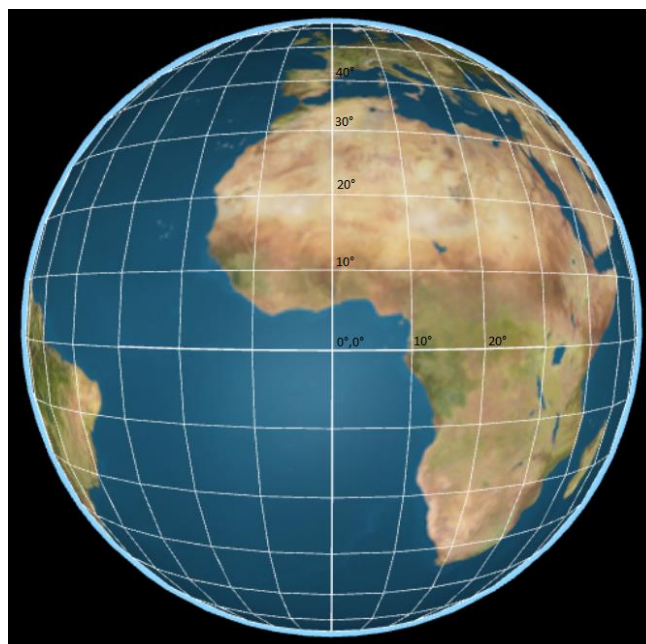


Figure 6: Virtual Globe created with three.js

3.3 Integrating geospatial world with Virtual Globe

In order to georeference the geospatial world with the Virtual Globe the following adjustments and acceptances

should be considered:

- The custom geospatial world has to be designed or adjusted so that its axes are aligned with north-south and east-west orientations ($y+/-$: N/S, $x+/-$: E/W). In other words, the minimum x , minimum y values of the world should refer to the south-west of the custom world and the maximum x , maximum y to the north-east (Fig. 7).

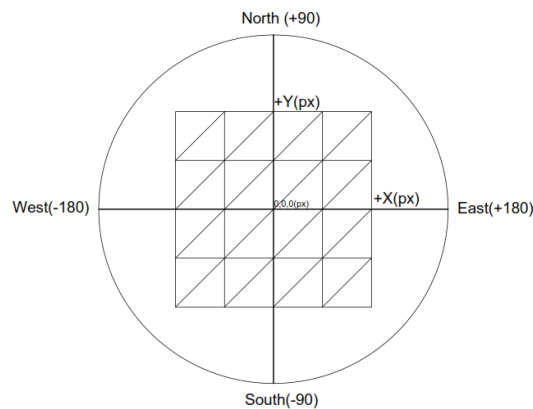


Figure 7: Adjusting custom world's axes orientation

- The center of the geospatial world is positioned at $x:0$, $y:0$ pixel coordinates. Once the geospatial world is georeferenced, then its center will be placed at its real WGS 84 ϕ_{gw} , λ_{gw} coordinate values.
- The center of the Virtual Globe is positioned at $x:0$, $y:0$ pixel coordinates and is placed at $\phi_0:0_0$, $\lambda_0:0_0$ WGS 84 coordinate values.
- Then, the Virtual Globe is rotated counterclockwise around x axis by ϕ_{gw} degrees and clockwise around y axis by λ_{gw} degrees. That way the $x:0$, $y:0$ pixel coordinates of the Virtual Globe are placed at the center of the geospatial world (Fig. 8 and Fig. 9).

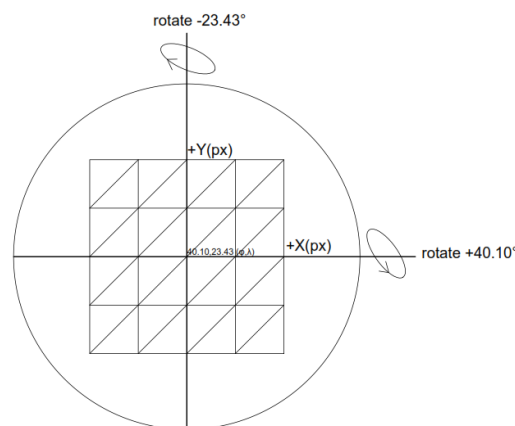


Figure 8: Rotating Virtual Globe to match geospatial world

As a result the 3D scene of the "living" geospatial world created is appropriately adjusted and is already viewable and may be explored through the Virtual Globe (Fig. 9)

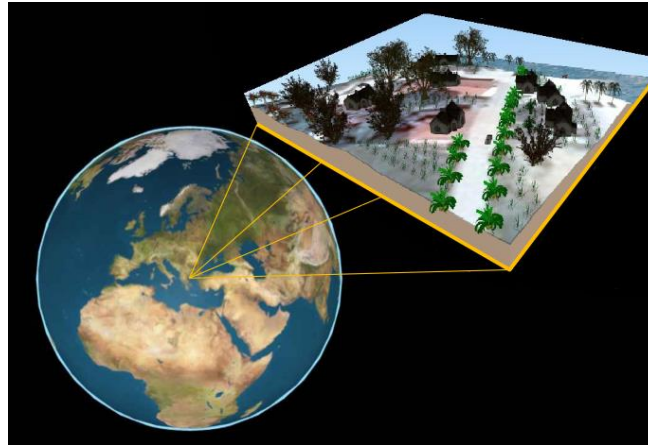


Figure 9: Integrating "living" geospatial worlds with Virtual Globes

An indicative demonstration video presenting all of the above developments is available at <http://goo.gl/M76uw3>.

4 Conclusions & Further Developments

During last decade we experienced a significant progress occurred in the development of WebGL Javascript libraries, APIs and interfaces and their high penetration values on web applications. As regards the geospatial community, similar developments concentrate an ever increasing interest by geoscientists and professionals. Strong representative examples are found in Cesium.js for implementing Virtual Globes and 3D geospatial visualizations and model animations. Others are met in three.js developments for advanced 3D model animation and motion effects. To our intuition, the above Javascript libraries as well as new ones focusing also on geospatial data exchange and interoperability issues, are expected to dramatically affect Virtual Globes evolution and usability.

Challenges involve all Virtual Globes usability cases such as, the educational sector, through sophisticated large scale geo-visualizations containing motion and animation effects on areas sensible on geological phenomena and/or natural disasters or on regions of the atmosphere participating in meteorological processes.

So far, 3D rendering on the web for demanding imagery resolutions required special computer graphics hardware capabilities and high speed and quality internet connections. With the transition of processing on the client side and the adoption of open web standards and advanced interfaces (e.g. physics.js) it could become more efficient to perform complicated geovisualized simulations over the globe for typical engineering modeling: fluid mechanics, hydrological modeling for flood assessment, landslide assessments etc.

A virtual geospatial reality is just around the corner.

5 References

- Riedl, A. (2007). Digital globes. In *Multimedia cartography* (pp. 255-266). Springer Berlin Heidelberg.
- Tuttle, B.T., Anderson, S. and Huff, R. (2008). Virtual globes: an overview of their history, uses, and future challenges. *Geography Compass*, 2(5), pp.1478-1505.
- Dunne, D., & Sutton, G. (2007). 3D Web-mapping: Integrating Marine Data into NASA World Wind. *Hydro International*, 7-9.
- Evangelidis, K., Ntouros, K., & Makridis, S. (2012). Geoprocessing Services over the Web. In *Proceedings of the 32nd EARSeL Symposium, Mykonos, Greece* (pp. 344-349).
- Angevine, G., Migrating from Earth to Cesium, [Online]. Available from: <https://cesiumjs.org/2015/01/27/-Migrating-from-Earth-to-Cesium/>