



Proposed Enhancements to the X3D Geospatial Component

Title	Proposed Enhancements to the X3D Geospatial Component
Item Type	Conference Paper
Authors	McCann, Michael;Puk., Richard;Hudson, Alan;Melton, Rex;Brutzman, Don
Citation	Web3D 2009
URI	https://hdl.handle.net/10945/71181
Publisher	Association for Computing Machinery (ACM)
Date Issued	2009-06
Download date	2026-04-14 10:54:21
Link to Item	https://hdl.handle.net/10945/71181

Downloaded from NPS Archive: Calhoun

Proposed Enhancements to the X3D Geospatial Component

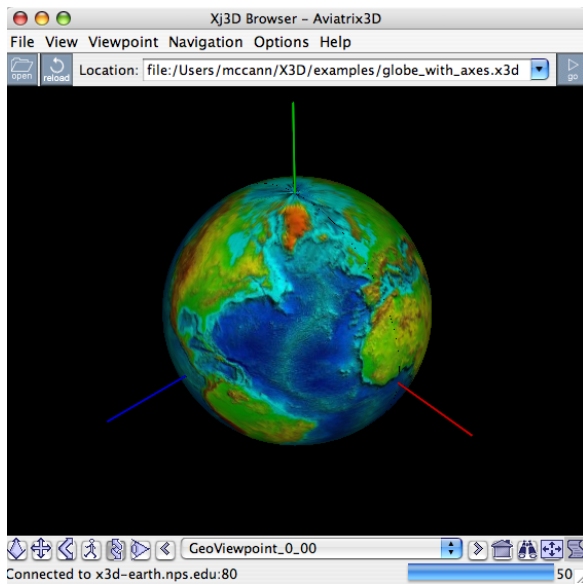
Michael McCann*
MBARI

Richard Puk†
Intelligraphics Inc.

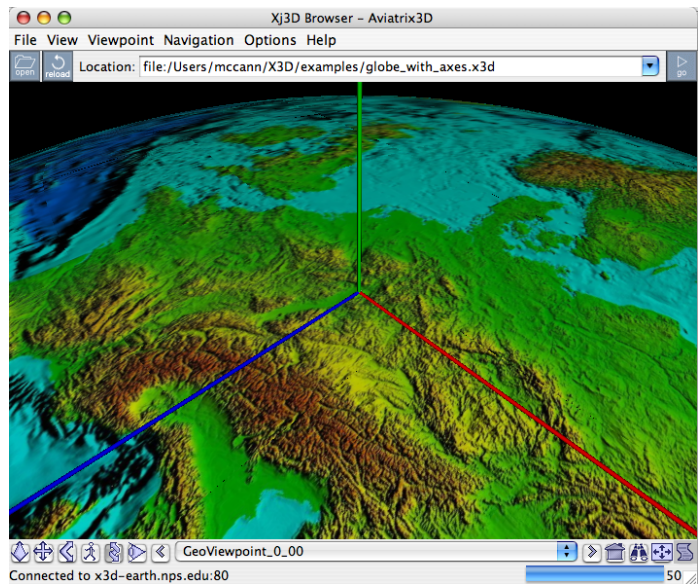
Alan Hudson‡
Yumetech Inc.

Rex Melton§
Yumetech Inc.

Don Brutzman¶
Naval Postgraduate School



(a) View of globe with symbolic axes placed at center of the Earth.



(b) View of northern Europe with symbolic axes placed at Darmstadt, Germany.

Figure 1: Typical use of geospatial data in X3D: global 1-minute horizontal resolution oceanographic and subaerial terrain data included as an inline from <http://x3d-earth.nps.edu> [Yoo 2007]. Figure (a) shows axes for the geocentric cartesian coordinate system, which is appropriate for global views such as this. Figure (b) shows axes for a local coordinate system that is appropriate for high resolution object placement and navigation in the region of northern Europe.

Abstract

The Geospatial Component of the X3D Standard suffers from some deficiencies that prevent its wider adoption. This paper addresses two of these deficiencies. The first problem is that in order to reduce spatial jitter content must be built with regionally defined GeoOrigin nodes. This approach is fine for localized regional geospatial data visualization requirements, but fails for accurately viewing data in a global context or for combining content with different GeoOrigins. The second problem is the limited options for providing terrain data to the browser for rendering. The GeoLOD node is not the optimal solution for allowing the X3D browser to render terrain data in a high performing manner. This paper presents solutions to these problems for consideration by the Web3D community.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism

Keywords: 3-D Geography, X3D, geospatial, X3D-Earth, floating origin, terrain rendering, data visualization

*e-mail: mccann@mbari.org

†e-mail: puk@igraphics.com

‡e-mail: giles@yumetech.com

§e-mail: rex@yumetech.com

¶e-mail: brutzman@nps.edu

1 Introduction

X3D is ideally suited for building visualizations of all manner of real-world objects and information constructs in a geospatial context. The eleven nodes that comprise the Geospatial Component of the X3D Standard [Web3D Consortium 2007] have proven capable for building operational scientific data visualization tools [McCann 2004] but have failed to achieve wider adoption. Fig. 1 shows typical views rendered by the X3D Geospatial component. Attempts to use the existing capability for "mash-ups" of data on a global scale have revealed some problems with the existing specification. This paper will describe two major issues and present some options for improving the situation.

1.1 GeoOrigin Problem

In the original GeoVRML implementation [Reddy et al. 2000] the concept of GeoOrigin was introduced to improve spatial positioning and navigational accuracy for a small region on the globe. This allowed early implementations to work with the leading VRML

Copyright © 2009 by the Association for Computing Machinery, Inc.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.

Web3D 2009, Darmstadt, Germany, June 16 – 17, 2009.

© 2009 ACM 978-1-60558-432-4/09/0006 \$10.00

browsers of the time. The assumption was that the creator of virtual worlds was the best person to specify a local area of interest. The initial specification of the X3D Geospatial Component retained most of this capability only improving it slightly by using the available X3D double precision data types which were not available in VRML.

There are two requirements for achieving positional accuracy. The first requirement is for double precision geospatial (latitude and longitude) coordinates. The second requirement is to have some technique for converting the high-precision geographic coordinates to the lower-precision coordinates that are used in the computer graphics pipeline. The GeoOrigin node is the technique currently in use in the Geospatial Component to accomplish this transition. GeoOrigin is a straightforward but limiting method for transitioning from high-precision geographic coordinates to lower-precision computer graphics pipeline coordinates.

There are several problems with GeoOrigin, namely:

- Accuracy of object placement and camera movement will be high within only about 100 kilometers of the GeoOrigin, beyond this distance spatial jitter becomes apparent (single precision numbers are good to about 1 part in 100,000)
- Content built with one GeoOrigin cannot be combined with content built with another GeoOrigin
- A backdrop X3D model of planet Earth built with a GeoOrigin at the center of the Earth (the default if no GeoOrigin is specified) will not have adequate accuracy for objects or camera motion near the surface of the Earth (Fig. 1 (a))

These issues are described in detail in [Thorne 2007] and [Thorne 2005]. Section 2 of this paper presents a specific solution to the problem with GeoOrigin.

1.2 Terrain Rendering Problem

Any realistic X3D terrain model of the Earth has dozens of levels of detail. To implement level of detail management for terrain in X3D the content author currently uses the GeoLOD node, which has four fields for the four child tiles, which themselves typically contain GeoElevationGrids and another GeoLOD node up to the highest resolution tiles of the data set. In order for the browser to load higher resolution tiles it must have first loaded the parent tile. There is no way around this with the GeoLOD node – it is a specific rendering structure with distance rules that don't allow the skipping of levels. This presents a serious performance issue for the viewer who wishes to navigate from space-to-face. The viewpoint may quickly move to a few hundred meters above city streets, but the browser is busy loading all the intervening tiles and their children before the high-resolution tiles can be presented.

A similar performance hit occurs for a viewpoint moving at a fixed distance above the terrain. The ideal situation would allow adjacent "cousin" tiles to immediately load as they come into view. However, this is not the case; all the parent tiles need to be loaded, and then unloaded, before the child tiles can be loaded. For a viewpoint traveling close to the globe this can be a severe hit on performance as several generations of ancestor tiles would need to be loaded and unloaded before the appropriate tiles are displayed. Fig. 2 shows that for a viewpoint moving left-to-right and looking to the right would first require the parent tile P2 to be loaded, and then unloaded, before child tiles C5 and C7 can be loaded.

Another problem with GeoLOD is that the specification allows any X3D content in the childUrls. Requiring the browser to parse and render all possible content within GeoLOD complicates what the

browser needs to do and hinders its ability to build an efficient data structure for high performance terrain rendering. This paper proposes a solution in the form of a new node to be added to the Geospatial Component.

2 Fixing Floating Point Precision by Removing GeoOrigin

Considering the previously stated problems with GeoOrigin and our desire to simplify the use of geospatial data we propose deprecating the use of the GeoOrigin node. This is also the suggested action proposed by [Thorne 2005]. This change does not eliminate the need to properly translate geographic coordinates to computer graphics coordinates – that duty is transferred to the browser. Instead of the content developer deciding on a fixed GeoOrigin the browser will need to perform the translation.

The browser will determine how the local origin will be located at or near to the geographic coordinates of the user's viewpoint. It may be moved piecewise as the user navigates over large distances or it may be recalculated continuously, or via any other method the browser developer can devise.

2.1 Accuracy hint

The X3D specification needs to stipulate that browsers implementing the geospatial component must meet an accuracy metric. This metric is offered to the browser through a new accuracy field.

We propose adding the following field to NavigationInfo:

```
SFFloat [in,out] accuracy 10 [0,∞)
```

The accuracy field specifies the requested positional accuracy (in meters) for purposes of object placement and navigation in the graphics pipeline. This is particularly important for the Geospatial Component where large-scale global visualizations will have an accuracy requirement of hundreds of meters and regional city-scale visualizations will have an accuracy requirement of 0.1 meters. The accuracy field serves as a hint to the browser that it may need to employ special techniques, such as adjusting the local origin, to achieve the requested accuracy for rendering.

2.2 Implementation

We successfully tested the implementation of a dynamic GeoOrigin in the open source Xj3D Java toolkit. Internal to the browser, all geo-referenced coordinates are converted into GCC (Geocentric Cartesian) for use in rendering and transformation. The original implementation offsets those coordinates by the GeoOrigin position specified in the content. The functional addition to the browser consists of a manager class that makes determinations based on the user's Viewpoint on when and where a new local origin is required, and notifies the loaded Geo nodes that they must update their internal state to account for the new setting.

Local origins are configured at either the ellipsoid center, or at the ellipsoid surface. The default origin manager's criteria for switching the local origin setting are configured with two parameters. An elevation above the ellipsoid is used as a threshold for establishing an initial local origin. A Viewpoint elevation above the threshold causes the local origin to be configured at the ellipsoid center. An elevation below the threshold causes a local origin to be configured at the Viewpoint's geodetic position on surface of the ellipsoid. Once a local origin is established at the ellipsoid surface, an orthodromic distance from that local origin is used as a threshold for reconfiguration. The value of the NavigationInfo accuracy field

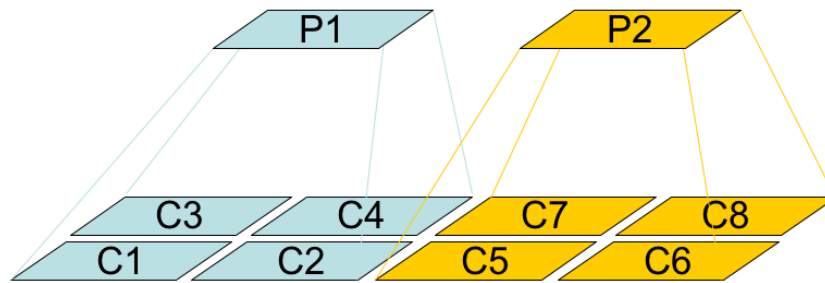


Figure 2: Level Of Detail for two parent tiles and their 4 children. In the existing GeoLOD scheme a viewpoint traveling left-to-right would require the parent tile P2 to be loaded, and then unloaded, before child tiles C5 and C7 can be loaded: an unnecessary operation that impacts performance.

is used to set this threshold distance. When the Viewpoint moves beyond the threshold distance from the local origin a new origin is established, again at the Viewpoint's geodetic position on surface of the ellipsoid.

3 Improving Terrain Rendering with GeoTerrainLOD

3.1 Just the Data, Ma'am

A complicating aspect of the GeoLOD node for terrain rendering is that there are no restrictions on the kind of X3D content that can be contained in the childUrls. GeoLOD allows anything, including Script nodes containing runtime content. Handling all such content complicates what the browser must do. We want the browser to load in terrain data and have it render it in the best way possible. Therefore, we propose the new GeoTerrainLOD node whose content is restricted to just the GeoTerrainLOD and GeoElevationGrid nodes. The signature of the node is shown in Fig. 3.

When provided with just the terrain data the browser is free to choose an appropriate rendering algorithm. There are many to choose from, e.g. [Duchaineau et al. 1997], [Hoppe 1998].

Note that the differences from the GeoLOD node are few but significant, namely:

- the `children` field has been removed
- the `level_changed` field has been removed
- the `geoOrigin` field has been removed
- the `rootUrl` field has been removed
- the `range` field remains, but only as a hint to the browser, which is permitted to use other methods to determine when to swap tiles
- the `childNUrls` content is restricted to GeoTerrainLOD and Shape nodes containing GeoElevationGrid and Appearance with ImageTexture nodes

This new node is somewhat of a compromise on the path to achieving an optimal solution for the streaming of terrain data to an X3D browser. The existing tool chains (MB-System [Caress and Chayes 2006] and Rez [Thorne 2008]) that have been developed for GeoLOD can be easily modified to change their output to GeoTerrainLOD. Within the 3D graphics community, there are at least half a dozen commonly used rendering techniques for large scale terrains. Each technique requires the underlying data to be

fetches in its own unique form. Ideally, the X3D browser would receive the data in a format that is native to each technique – a simple grid of data points as provided in GeoElevationGrid nodes is not always the most suitable way for these rendering engines to work. The GeoTerrainLOD + GeoElevationGrid combination can be used to deliver data to the browser at appropriate levels of detail. Because the childUrl content is restricted to just terrain data the browser is free to feed the data - after translation into appropriate data structures - into its choice of rendering algorithms.

The GeoTerrainLOD node will provide an improvement over the existing GeoLOD node. However, to meet future requirements for large terrain data visualization X3D needs to evolve a system where it can handle streamed data while maintaining compatibility with X3D's traditional design philosophy. The design of such a system should also support allowing the browser to scale the content's detail based on the capabilities of the individual system upon which the browser is installed. While the content developer has the option of asking and instructing the browser on the type(s) of content to display, the browser has the option of filtering it in order to maintain a constant frame rate.

As proposed by [de La Beaujardière et al. 2000] an attractive option for delivering digital elevation and map image data to X3D browsers is to employ interoperable standards such as those of the Open Geospatial Consortium (OGC). For example, the Web Coverage Service can deliver digital elevation data and the Web Map Service (and Web Map Tile Service) can deliver images for texture mapping over the elevation data. The proposed OGC Web 3D Service is being updated to specify X3D (instead of VRML) as the transport content. Elimination of X3D's GeoOrigin node and implementation of GeoTerrainLOD will provide a clear path for getting data from OGC services into X3D browsers.

3.2 Tile Index

For the browser to make efficient use of the quad-tree terrain tile set provided by the GeoTerrainLOD and to avoid the problems mentioned above there needs to be an index relating geospatial position and tile resolution to the Urls. Given such an index the browser can immediately load the appropriate tiles without traversing the parent-child hierarchy. All of the information needed to build an index into the terrain tile set is contained within the attributes of the GeoElevationGrid and the structure of the GeoTerrainLODs. One way for this index to be created is for the browser to traverse a branch of the tree (without loading the GeoElevationGrid and Texture data) to a leaf node to discover the number of levels and the resolution at each level, and then traverse the edges to discover the range of the tiles. This quick examination will reveal the pattern of

```

GeoTerrainLOD : X3DChildNode, X3DBoundedObject {
  SFNode [in,out] metadata NULL [X3DMetadataObject]
  SFVec3d [] center 0 0 0 (-∞,∞)
  MFString [] child1Url [] [URI]
  MFString [] child2Url [] [URI]
  MFString [] child3Url [] [URI]
  MFString [] child4Url [] [URI]
  MFString [] geoSystem ["GD", "WE"] [see 25.2.3]
  SFFloat [] range 10 [0,∞)
  MFNode [] rootNode [] [X3DChildNode]
  SFVec3f [] bboxCenter 0 0 0 (-∞,∞)
  SFVec3f [] bboxSize -1 -1 -1 [0,∞) or -1 -1 -1
}

```

Figure 3: Node signature for the new *GeoTerrainLOD* node which is modeled after the *GeoLOD* node but allows only *GeoElevationGrid* content, simplifying the browser's understanding of Terrain data so that it is allowed to render it with its choice of algorithms.

the index. Once the index is available the browser can load only the tiles that are needed to be loaded depending on the properties of the user's navigation (view frustum, distance above terrain, speed of movement, etc.)

This index generating approach will be successful only if the terrain tile sets are regular and consistent - as is the case for the bathymetric terrain tile sets used in [McCann 2004]. A terrain tile set may contain a region(s) of high resolution data for a city or other area of interest. In this case the higher resolution area(s) may not be discovered by the quick examination described above. For tile sets such as this some other mechanism is required to deliver the meta-data needed to build an accurate index. Design of such an indexing catalog is beyond the scope of this paper and is an upcoming task for the X3D-Earth Working Group.

4 Conclusion

The popularity of virtual globe viewers such Google Earth increases interest in using geographic data which increases the market for additional ways to visualize geospatial phenomena. The international X3D Standard appeals to customers that require a 3D file format and runtime specification that is open, royalty free, and ISO ratified. The enhancements to the standard discussed in this paper aim to simplify it, improving use of geospatial data in X3D. Removing the requirement for *GeoOrigin* and implementing requested resolution capability within the browser makes it easier to combine data from various sources. Adding *GeoTerrainLOD* permits more efficient loading of terrain data and allows browser developers to more easily use terrain rendering algorithms of their choice.

Acknowledgements

The suggestions presented in this paper are the result of many discussions that have taken place over the last few years within the X3D-Earth Working Group of the Web 3D Consortium. In particular, a workshop held at MBARI on 5 and 6 March 2009 lead to the specific solutions presented in this paper. Several individuals contributed to the discussion; we would like to thank Chris Thorne, John Stewart, Anita Havele, Justin Couch, Mike Aratow, and Byoungyun, Yoo. This research and the workshop were supported by the Monterey Bay Aquarium Research Institute through funding from the David and Lucile Packard Foundation.

References

CARESS, D., AND CHAYES, D., 2006. MB-System: Mapping the Seafloor. <http://www.ldeo.columbia.edu/res/pi/MB-System> and <http://www.mbari.org/data/mbsystem>.

DE LA BEAUJARDIÈRE, J., MITCHELL, H., RASKIN, R., AND RAO, A. 2000. The nasa digital earth testbed. In *GIS '00: Proceedings of the 8th ACM international symposium on Advances in geographic information systems*, ACM, New York, NY, USA, 47–53.

DUCHAINEAU, M., WOLINSKY, M., SIGETI, D. E., MILLER, M. C., ALDRICH, C., AND MINEEV-WEINSTEIN, M. B. 1997. Roaming terrain: real-time optimally adapting meshes. In *VIS '97: Proceedings of the 8th conference on Visualization '97*, IEEE Computer Society Press, Los Alamitos, CA, USA, 81–88.

HOPPE, H. 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *VIS '98: Proceedings of the conference on Visualization '98*, IEEE Computer Society Press, Los Alamitos, CA, USA, 35–42.

MCCANN, M. P. 2004. Using *geovrml* for 3D oceanographic data visualizations. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, ACM, New York, NY, USA, 15–21.

REDDY, M., IVERSON, L., AND LECLERC, Y. G. 2000. Under the hood of *geovrml* 1.0. In *VRML '00: Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, ACM, New York, NY, USA, 23–28.

THORNE, C. 2005. Using a floating origin to improve fidelity and performance of large, distributed virtual worlds. In *CW '05: Proceedings of the 2005 International Conference on Cyberworlds*, IEEE Computer Society, Washington, DC, USA, 263–270.

THORNE, C., 2007. Next Steps for X3D Geospatial Specification. <http://www.web3d.org/x3d-earth/meetings/2007November/NextStepsForGeospatial.pdf>.

THORNE, C., 2008. Rez3D Terrain Generation software. <http://www.rez3d.com> and <http://sourceforge.net/projects/planet-earth>.

WEB3D CONSORTIUM, 2007. IISO/IEC 19775-1.2:2008. <http://www.web3d.org/x3d/specifications/>, December.

YOO, B., AND BRUTZMAN, D. 2009. X3D Earth Terrain-Tile Production Chain for Georeferenced Simulation. In *Web3D '09: Proceedings of the 14th international conference on 3D Web technology*, ACM.

YOO, B., 2007. X3D-Earth Globe Under Development. <http://x3d-earth.nps.edu/>.